

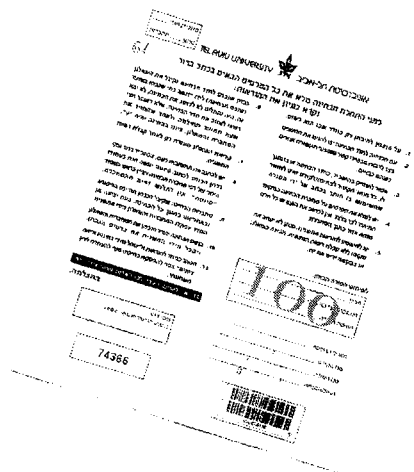
ועד הנדסה



# 0509.1821 תכנות למהנדסים

בהצלחה,

ועד הנדסה





סמסטר א' תשי"ע  
משך הבחינה : שלוש שעות  
אין להשתמש בחומר עזר

## בחינה בקורס תכנות ב- C

**מרצים :** אפרת משיח, אילונה קיפר, מתי שמרת

**מתרגלים :** נוגה אורון-לוי, ידעאל ולדמן, אסף זריצקי,  
יהודית חסון, נעמה מאייר, גיא קרליבך

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספרייה (זכרו לבצע `#include` מתאים).
- בסוף המבחן נמצא דף-עזר.
- נא לכתוב בכתב קריא וברור.
- אם המקום המוקצה מקום לפתרון אינו מספיק תוכלו לכתוב גם בצידו השני של הדף.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.
- מומלץ לקרוא כל שאלה עד סופה, על כל סעיפיה, לפני תחילת הפתרון.

**בהצלחה!**

### לשימוש הבודקים :

תעודת זהות	
מספר מחברת	
שאלה 1	32
שאלה 2	31
שאלה 3	28
ציון	91

**שאלה מס' 1 (33 נקודות)**

בשאלה זו תכתבו פונקציות לבדיקת תקינות תחביר של פקודות מסוימות בשפת C. בכל הסעיפים ניתן להניח שהמצביע המתקבל ע"י הפונקציה הוא אינו NULL.

**סעיף א' (7 נקודות):**

כתבו פונקציה המקבלת מחרוזת שמייצגת שם של משתנה. הפונקציה בודקת האם שם המשתנה הוא חוקי, לפי הכללים הבאים:

- o השם מכיל רק אותיות, מספרים ואת התו '\_'
- o השם לא יתחיל בספרה

הפונקציה מחזירה 1 אם השם חוקי ו-0 אחרת.

```

int checkVariable(const char *var) {
    int i=0;
    if (var[0] >='0' && var[0] <='9')
        return 0;
    /* המלגה מתחיל כסיפורה */
    while (var[i] != '\0')
    {
        if (!(var[i] >='A' && var[i] <='Z'))
            //
            if (!(var[i] >='0' && var[i] <='9'))
                //
                if (var[i] != '_')
                    //
                    return 0;
        i++;
    }
    return 1;
}
    
```

6/7

z ?

**סעיף ב' (12 נקודות):**

בסעיף זה נכתוב פונקציה רקורסיבית הבודקת חוקיות של ביטוי עם סוגריים מהסוג '( )'. הביטוי חוקי פירושו שמתקיימים התנאים הבאים:

- במעבר על הביטוי משמאל לימין, בשום שלב אין יותר סוגריים ימניים משמאליים.
- מספר הסוגריים השמאליים והימניים בביטוי הוא שווה (ראו דוגמא).

הפונקציה מקבלת את המחרוזת exp המכילה ביטוי כלשהו, ופרמטר נוסף numPar המייצג את מספר הסוגריים שלא נסגרו עד המיקום הנוכחי במחרוזת. הפונקציה מחזירה 1 אם הביטוי חוקי ו-0 אם הוא לא חוקי.

**לדוגמא:**

- הביטוי הבא חוקי מבחינת סוגריים

((3+4)-5)

- הביטוי הבא אינו חוקי מבחינת סוגריים כי במקום השישי במחרוזת מספר הסוגריים הימניים (1) קטן ממספר הסוגריים השמאליים (2).

(a+b))+3

עליכם להשלים את הפקודות החסרות בפונקציה.

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכתוב ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

```

int checkParenthesis(const char *exp, int numPar) {
    // base cases
    if ((*exp == '/' && numPar > 0) || numPar < 0)
        return 0;

    if (*exp == '/' && numPar == 0)
        return 1;

    // solve recursively
    if ( *exp == '(' )
        return checkParenthesis(exp+1, numPar+1);
    else if (*exp == ')' )
        return checkParenthesis(exp+1, numPar-1);
    else
        return checkParenthesis(exp+1, numPar );
}
    
```

12  
12

**סעיף ג' (14 נקודות):**

כתבו פונקציה המקבלת את המחרוזת cmd שמכילה פקודה ובודקת את חוקיותה. פקודה חוקית יכולה להיות מהסוגים הבאים:

1. הכרזה על משתנה מסוג int, כלומר מחרוזת שנראית כך:

int משתנה חוקי;

2. פקודת if

if (ביטוי חוקי מבחינת סוגריים)

בכדי שפקודה תהיה חוקית, צריכים להתקיים הכללים הבאים:

- חייב להיות רווח בודד אחרי המילה if וגם אחרי המילה int.
- בסוף מחרוזת עם הכרזה על משתנה חייב להופיע נקודה-פסיק (;).
- הביטוי שאחרי המילה if:

- חייב להתחיל ב- '(' ולהסתיים ב- ')'

- הביטוי בתוך הסוגריים חייב להיות חוקי מבחינת ביטוי הסוגריים, כמוגדר בסעיף ב' (ראו דוגמאות בהמשך).

אם נמצא שהפקודה חוקית, יוחזר 1. אחרת יוחזר 0.

**שימו לב**

- ניתן להניח שאורך שם משתנה וכן אורך הביטוי שבסוגריים לא יעלה על 50 תווים
- אין לשנות את מחרוזת הקלט המקורית.

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכתובת ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

• פקודות חוקיות

```
int id;
if ( () () )
```

• פקודות לא חוקיות

```
int id
int 2id;
if ()()
```

```
if a==b
#include <string.h> #define MAX 50
```

```

int checkCommand(const char *cmd)
{
  int i=0;
  char var[MAX+1-5]; // המשתנה
  if (strcmp("int", cmd, 4) == 1)
  {
    // אם התחריט נכונה נשגה אקו יל מוח
  }
  // checkVariable(cmd)
  while (cmd[i] != '\0' && cmd[i] != ';')
  {
    var[i] = cmd[i+4]; // var - המשתנה
    i++;
  }
  var[i] = '\0'; // הפסק
  if (checkVariable(var) == 0)
  {
    return 0; // לא חוקי
  }
  // checkParenthesis(cmd) == 0
  if (cmd[strlen(cmd)-1] == ';')
  {
    return 1; // חוקי
  }
  if (strcmp("if", cmd, 3) == 1) //
  {
    if (cmd[3] != '(') //
    {
      return 0;
    }
    // checkParenthesis(cmd) == 0
    return 0;
  }
}

```

הקובץ המקובל + '0'

המסק

המסק

המסק

מכיל קריסטל

אם הגא

14/14

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לעלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכתית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

```
if (checkParenthesis(cmd+4) == 0)
```

צתקבוק אחרי הסגר הכאשן והקובץ מסוקרין חוקי

```
return 0;
```

```
return 1;
```

\*

← המשך

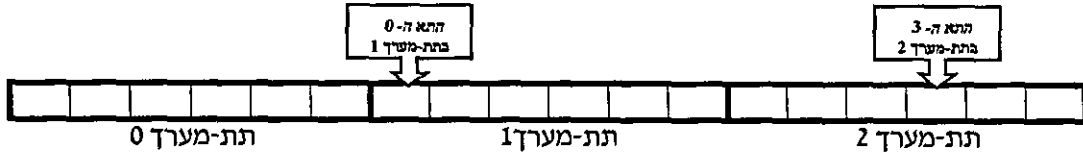
|| ~~return~~ <sup>return</sup> 0;   
 // ~~return~~ <sup>return</sup> 0; if / int - a function   
 }

**שאלה 2 (34 נקודות)**

בשאלה זו תתבקשו לכתוב פונקציות ותוכנת מחשב לניהול משחק הלוטו.

**סעיף א' (5 נקודות):**

"מערך משורשר" הוא מערך המכיל ערכים של מספר תתי מערכים (באותו אורך) באופן רציף בזה אחר זה. לדוגמא, באיור הבא מתואר "מערך משורשר" המכיל 3 תתי מערכים שכל אחד בגודל 6.



כתבו פונקציה המקבלת מערך `arr` של מספרים שלמים שהוא "מערך משורשר", את הגודל של כל תת-מעריך בו ( $n$ ) ו-2 מספרים שלמים  $i$  ו- $j$ . הפונקציה תחזיר את הערך של התא ה- $j$  בתת-המעריך ה- $i$ . ניתן להניח שהקלט לפונקציה הוא תקין, כלומר ש-`arr` הוא אכן "מערך משורשר" בגודל מתאים וש- $0 \leq i, j, n$ .

```
int getVal(int arr[], int n, int i, int j)
{
    return arr[i*n+j];
}
```

**סעיף ב' (14 נקודות):**

בהגרלת הלוטו מעלים בגורל 6 מספרים שונים בטווח 1-49 (כולל). טופס המכיל את ששת המספרים שעלו בגורל זוכה בפרס.

- ממשו פונקציה הבודקת מי הם הזוכים בפרס מבין כל הטפסים שמולאו. הקלט לפונקציה הוא:
- `winNums` - מערך בגודל 6 המכיל את המספרים הזוכים.
- `tickets` - "מערך משורשר" (כפי שהוגדר לעיל) המכיל תתי מערכים באורך 6 המייצגים טפסים של מספר משתתפים.
- `numTickets` - מספר הטפסים במערך `tickets`.
- `res` - מערך נוסף בגודל מספר הטפסים המשתתפים שיעודכן ע"י הפונקציה.

בסיום פעולת הפונקציה התא ה- $i$  במערך `res` יכיל את הערך 1 אם הטופס ה- $i$  זכה בפרס ו-0 אם לא זכה. הפונקציה תחזיר כמה טפסים זכו בפרס הראשון.

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מננית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

- כדי לקבל את מלוא הנקודות בסעיף זה יש לעבור רק פעם אחת על המערך winNums פעם אחת על המערך tickets.
- ניתן להניח שהקלט לפונקציה הוא חוקי (כל הערכים בתחום המתאים, כל המערכים בגודל מספיק וכדומה).

# define SIZE-TIC 6

```

int checkWin(int winNums[], int tickets[], int numTickets,
int res[])
{
    int i, j, curr, counter = numTickets;
    for (i=0; i < numTickets; i++)
        res[i] = 1;

    // נניח ננינו סקור כולק
    for (i=0; i < SIZE-TIC; i++)
        for (j=0; j < numTickets; j += 6)
        {
            if (res[j] == 1)
            {
                // לאקב סב כה אר חס' האוכים
                curr = getVal(arr, SIZE-TIC, j, i);
                if (winNums[i] != curr)
                {
                    // לא תאק ותצבן תצבן ומונה
                    res[j] = 0;
                    counter--;
                }
            }
        }

    return counter;
}

```

למה curr לא

B/14



**סעיף ג' (15 נקודות):**

הפרס הראשון בלוטו הוא 10,000,000 ש"ח.  
אם כמה משתתפים ניחשו נכון את ששת המספרים הזוכים הם יתחלקו בפרס באופן שווה. שימו לב שייטכן מצב בו אף משתתף לא ניחש נכונה את כל ששת המספרים.

כתבו תוכנית הקולטת מהמשתמש את הנתונים הבאים לפי הסדר המתואר:

1. ששת המספרים הזוכים
2. מספר המשתתפים
3. טפסי הניחוש בזה אחר זה

פלט התוכנית הוא טבלה שבכל שורה בה מופיע מספר סידורי של משתתף וסכום הזכייה שלו.

אפשר להניח (ואין צורך לבדוק זאת) שהקלט הוא חוקי.

**דוגמא:**

אם נקלטו המספרים הזוכים הבאים:

6 9 19 25 32 33

מספר המשתתפים שנקלט הוא:

10

ואחריו הטפסים הבאים:

10 12 14 15 28 37  
2 7 9 10 12 20  
1 3 4 32 33 35  
6 9 19 25 32 33  
3 6 22 32 33 36  
6 9 19 25 32 33  
1 11 12 27 31 35  
10 15 19 25 34 39  
6 9 19 25 32 33  
16 19 22 24 26 28

המשתתפים הרביעי השישי והתשיעי ניחשו נכון את המספרים, ולכן פלט התוכנית יהיה:

0	0.00
1	0.00
2	0.00
3	3333333.33
4	0.00
5	3333333.33
6	0.00
7	0.00
8	3333333.33
9	0.00

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

```

#include <stdio.h>
#define SIZE 6

int main()
{
    int win[SIZE], tic-num, i, *res, *tickets, winners;
    printf("enter the winning nums/n");
for
    for (i=0; i<SIZE; i++)
        scanf("%d", &win[i]);
    printf("enter num of tickets/n");
    scanf("%d", &tic-num);
    tickets = (int*)malloc (tic-num*SIZE*sizeof(int));
    res = (int*)malloc (tic-num*sizeof(int));
    printf("enter tickets/n");
    for (i=0; i<tic-num*SIZE; i++)
        scanf("%d", &tickets[i]);
    winners = checkWin(win, tickets, tic-num, res);
    for (i=0; i<tic-num; i++)
    {
        printf("%d ", i);
        printf("%g/n", res[i]*100000000/winners);
    }
    free(res);
    free(tickets);
    return 0;
}

```

לברוק פירם

0.2 1/16 12

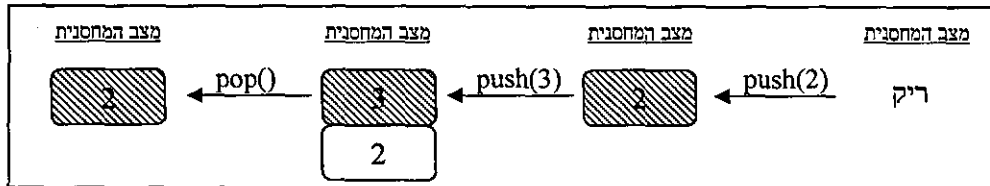
13  
15

**שאלה 3 (33 נקודות)**

מחסנית היא מבנה נתונים שבו הערך שנכנס ראשון יוצא אחרון. נגדיר שתי פעולות על המחסנית:

- push – דחיפת ערך לראש המחסנית (לדוגמא push(2) פירושו דחיפת הערך 2 לראש המחסנית)
- pop – הוצאת הערך העליון במחסנית (לדוגמא pop() פירושו החזרת הערך האחרון שהוכנס למחסנית)

לדוגמא: בציוור מתוארת סדרת הפעולות push(2), push(3) ו-pop(). על מחסנית ריקה, כאשר הערך בעל הרקע המקווקו הוא הערך שנמצא בראש המחסנית. הפעולה pop מחזירה את הערך 3 – הערך שהיה בראש המחסנית טרם הפעולה.



נממש מחסנית של מספרים שלמים בעזרת רשימה מקושרת. פעולת push דוחפת ערך חדש לראש הרשימה ואילו פעולת pop מוציאה את האיבר בראש הרשימה ומחזירה את ערכו. חל איסור על קריאה לפונקציה pop אם המחסנית ריקה.

נתון קוד חלקי המממש מחסנית של מספרים שלמים בעזרת רשימה מקושרת.

- המבנה Stack מכיל שדה המציין את גודל המחסנית, שדה זה חייב בכל זמן להיות תואם למספר האיברים שבמחסנית. בנוסף, המבנה מכיל את השדה head שהוא ראש הרשימה המקושרת של האיברים במחסנית. שימו לב שכל איבר ברשימה הוא מהטיפוס Element אותו תדרשו להגדיר בהמשך.

```
typedef struct stack_t
{
    int size;
    element *head;
} Stack;
```

סמל element  
ומתחיל Element  
(נ"ח א"ח)  
Element ->

- הפונקציה createStack יוצרת מחסנית חדשה ומאתחלת את ערכי השדות כנדרש. אם הקצאת הזיכרון נכשלה הפונקציה תחזיר NULL אחרת תחזיר מצביע למחסנית החדשה.

```
Stack * createStack()
{
    Stack *s = (Stack*) malloc(sizeof(Stack));
    if (s != NULL)
    {
        s->size = 0;
        s->head = NULL;
    }
    return s;
}
```

מס' מחברת : 37 מס' ת.ז. : 30115416

- נתונה הפונקציה createElement היוצרת איבר ברשימה המקושרת המשמשת למימוש המחסנית ומאתחלת את שדותיו.

```
Element * createElement(int data)
{
    Element *e = (Element*) malloc(sizeof(Element));
    if (e != NULL)
    {
        e->data = data;
        e->next = NULL;
    }
    return e;
}
```

- הפונקציה push דוחפת ערך לראש הרשימה על ידי הוספתו בתחילת הרשימה המקושרת.

```
void push(Stack *s, int data)
{
    Element *e = createElement(data);
    if (e == NULL)
    {
        printf("Fatal error: unable to allocate memory!\n");
        exit(1);
    }
    e->next = s->head;
    s->head = e;
    s->size++;
}
```

- הפונקציה isEmpty(Stack \*s) מקבלת מצביע למחסנית ומחזירה true אם המחסנית ריקה או false אחרת. מחסנית היא ריקה אם אין בה אנרים כלומר גודלה הוא 0.

```
int isEmpty(Stack *s)
{
    return s->size == 0;
}
```

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

מס' מחברת : 37 מס' ת.ו. : 30115416

קראו בעיון את הקוד הנתון וענו על כל הסעיפים הבאים :

סעיף א' (5 נקודות):

הגדירו את המבנה Element. ודאו כי הגדרת המבנה תואמת את השימוש שנעשה בו בפונקציות הנתונות.

```
typedef struct el-t {
    int data;
    struct el-t *next;
} Element;
```

סעיף ב' (8 נקודות):

ממשו את הפונקציה pop. הפונקציה מוציאה את האיבר שנמצא בראש המחסנית ומחזירה את ערכו למשתמש. שימו לב שעליכם לעדכן את כל השדות הרלוונטיים במבנה המחסנית וכן לדאוג לשחרור זיכרון אם נדרש.

הנחת מימוש : המחסנית אינה ריקה (אין צורך לבדוק זאת)

```
int pop(Stack *s)
{
    int data;
    Element *temp;
    temp = s->head;
    s = s->head->next;
    s->size--;
    data = temp->data;
    free(temp);
    return data;
}
```

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

סעיף ג' (8 נקודות):

ממשו את הפונקציה stackSum המקבלת כקלט מחסנית ומחזירה את סכום הערכים במחסנית תוך שימוש בפעולות שהוגדרו על המחסנית כדוגמת push, pop ו-isEmpty (חל איסור על גישה לרשימת האיברים באופן ישיר). לאחר הקריאה לפונקציה המחסנית תהיה ריקה.

```
int stackSum(Stack *s)
{
    if (isEmpty(s) == 1)
        return 0;
    return pop(s) + stackSum(s);
}
```

7  
8

סעיף ד' (12 נקודות):

בסעיף זה הינכם נדרשים לכתוב תכנית הקולטת מהמשתמש מספר לא ידוע של ערכים אי-שליליים ועושה שימוש במחסנית כדי לשמור אותם ולהדפיס את סכומם. התכנית תקרא ערכים מהמשתמש עד לקבלת ערך שלילי ותשתמש במחסנית כדי לאחסן ערכים אלו. לאחר סיום קריאת הקלט התכנית תחשב את סכום הערכים שנקראו, תדפיס אותו למשתמש ותסיים.

הערות:

- תוכלו להניח שקלט המשתמש תקין
- במידה והשתמשתם בהקצאת זיכרון דינמית ודאו שבוצע שיחרור של הזיכרון
- בניקוד סעיף זה יושם דגש על אי שכפול קוד ושימוש בפונקציות שהוגדרו בסעיפים קודמים

```
# include <stdlib.h>
```

מס' מחברת : 37 מס' ת.ז. : 3011546

```
# include <stdio.h>
```

```
int main()
{
    int temp;
    Element *e;
    Stack *s; // ... ? create Stack
    printf("enter nums until negative\n");
    scanf("%d", &temp);
    s → size = 0;
    s → head = NULL;
    while (temp >= 0)
    {
        push(s, temp);
        // סכום זכרון זיכרון
        scanf("%d", &temp);
    }
    printf("the sum of stack : %d\n",
           stacksum(s));
    // מקיאה פונקציה משתנה זכרון זיכרון
    return 0;
}
```

שאלה 1

## דף-עזר למבחן בתכנות

**שימו לב:** מטרת דף-העזר היא רק לחסוך לימוד בעל-פה של ההכרזות על הפונקציות המופיעות בו. יתכן שאין צורך להשתמש בפונקציות הללו כדי לפתור את המבחן, ויתכן שצריך להשתמש בפונקציות שנלמדו אך אינן מוזכרות בדף-העזר.

### פונקציות נבחרות מהספרייה `string.h`:

- `int strcmp(const char *s1, const char *s2)` (1) השוואת מחרוזות  
(2) השוואת n תווים ראשונים של מחרוזות  
`int strncmp(const char *s1, const char *s2, int n)`  
(3) חיפוש מופע ראשון של תו במחרוזת  
`char* strchr(const char *s, char c)`  
(4) חיפוש מופע אחרון של תו במחרוזת  
`char* strrchr(const char *s, char c)`  
(5) חיפוש מופע ראשון של מחרוזת במחרוזת אחרת  
`char* strstr(const char *s, const char *to_find)`  
(6) העתקת מחרוזת  
`char* strcpy(char *dest, const char *src)`  
(7) העתקת n תווים ראשונים של מחרוזת  
`char* strncpy(char *dest, const char *src, int n)`  
(8) שירשור מחרוזות  
`char* strcat(char *dest, const char *src)`  
(9) שירשור n תווים ראשונים ממחרוזת  
`char* strncat(char *dest, const char *src, int n)`  
(10) מציאת אורך מחרוזת  
`int strlen(const char *s)`

### פונקציות נבחרות מהספרייה `math.h`:

- `double sqrt(double x)` (1) שורש ריבועי  
`double pow(double base, double exponent)` (2) חזקה  
`int abs(int n)` (3) ערך מוחלט (שלמים)  
`double fabs(double x)` (4) ערך מוחלט (ממשיים)  
`double ceil(double x)` (5) עיגול למעלה  
`double floor(double x)` (6) עיגול למטה

### פונקציות נבחרות מהספרייה `ctype.h`:

- `int isalnum(int c)` (1) האם תו הוא אלפא-נומרי (a-z, A-Z, 0-9)  
`int isalpha(int c)` (2) האם תו הוא אות (a-z, A-Z)  
`int isdigit(int c)` (3) האם תו הוא סיפרה (0-9)  
`int islower(int c)` (4) האם תו הוא אות קטנה (a-z)  
`int isupper(int c)` (5) האם תו הוא אות גדולה (A-Z)  
`int isspace(int c)` (6) האם תו הוא whitespace (רווח, '\n' או '\t')  
`int tolower(int c)` (7) הפיכת אות גדולה לאות קטנה  
`int toupper(int c)` (8) הפיכת אות קטנה לאות גדולה

### פונקציות נבחרות מהספרייה `stdlib.h`:

- `void* malloc(unsigned int size)` (1) הקצאת זיכרון דינאמית  
(2) הקצאת זיכרון דינאמית ואיפוס הזיכרון שהוקצה  
`void* calloc(unsigned int nelements, unsigned int elementSize)`  
(3) הקצאה מחודשת של זיכרון  
`void* realloc(void *pointer, int size)`  
(4) שחרור זיכרון שהוקצה דינאמית  
`void free(void *pointer)`

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.





## בחינה בקורס תכנות ב- C

מרצים: אפרת משיח, אילונה קיפר, מתי שמרת

מתרגלים: נוגה אורון-לוי, ידעאל ולדמן, אסף זריצקי,  
יהודית חסון, נעמה מאייר, גיא קרליבך

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספרייה (זכרו לבצע #include מתאים).
- בסוף המבחן נמצא דף-עזר.
- נא לכתוב בכתב קריא וברור.
- אם המקום המוקצה מקום לפתרון אינו מספיק תוכלו לכתוב גם בצידו השני של הדף.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.
- מומלץ לקרוא כל שאלה עד סופה, על כל סעיפיה, לפני תחילת הפתרון.

**בהצלחה!**

### לשימוש הבודקים:

תעודת זהות	
מספר מחברת	
שאלה 1	31
שאלה 2	32
שאלה 3	30
ציון	93

מס' מחברת: ס 14 מס' ת.ז.: 304217704

31  
33

### שאלה מס' 1 (33 נקודות)

בשאלה זו נממש מחשבון פשוט לביטויים מתמטיים מהצורה

num1 operation num2

#### סעיף א' (6 נקודות):

כתבו פונקציה המקבלת מחרוזת שמייצגת מספר שלם אי-שלילי ומספר נוסף location המייצג מיקום במספר (ניתן להניח שהקלט תקין). הפונקציה תחזיר את ערך הסיפרה במיקום המבוקש. כלומר, עבור location==0 תוחזר סיפרת האחדות, עבור location==1 תוחזר סיפרת העשרות, וכן הלאה. לדוגמא, קריאה לפונקציה עם המחרוזת "123" והמיקום 0 תחזיר את הערך 3.

```

int findDigit(const char *num, int location) #include <string.h>
{
    int i, length, result;
    length = strlen(num); // include Point
    num += length - 1 - location;
    result = *num - '0';
    return result;
}

```

6/6

3

#### סעיף ב' (9 נקודות):

כתבו פונקציה המקבלת מחרוזת המייצגת מספר אי-שלילי שלם (אין צורך לבדוק את תקינות הקלט). הפונקציה הופכת את המחרוזת למספר. הפונקציה מחזירה את המספר שיוצג במחרוזת. מובטח שהמספר הינו בטווח של משתנה שלם. לדוגמא, עבור המחרוזת "123" תחזיר הפונקציה את המספר 123.

שימו לב -

בסעיף זה אין להשתמש בפונקציה atoi או atof

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

```
int strToNum(const char *num) #include <string.h>
{
    int length = strlen(num);
    int i, result = 0;
    i = length;
    while (i > 0)
    {
        result += findDigit(num, i - 1);
        result *= 10;
        i--;
    }
    return result / 10;
}
```

9/9

**סעיף ג' (12 נקודות):**

כתבו פונקציה המחשבת ביטוי חשבוני מהצורה:

num1 operation num2

הפונקציה מקבל מחרוזת המייצגת את הביטוי החשבוני ומחזירה את תוצאת פעולת החשבון.

מובטח כי מחרוזת הקלט תקינה, ובנויה בצורה הבאה:

- מספר שלם אי-שלילי (num1)
- תו רווח יחיד
- פעולת חשבון (operation) שיכולה להיות: +, -, \* (לא /)
- תו רווח יחיד
- מספר שלם אי שלילי (num2)

מובטח שהמספרים אינם ארוכים מ-10 ספרות.

דוגמא: עבור הקלט "1 + 2" תחזיר הפונקציה 3.

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא משופס הבחינה.

10  
12

```

int calcExpression(const char *exp)
{
    char num1[11];
    char num2[11];
    char op;
    int i, j, first, second;

    for (i=0; exp[i] != '\0'; i++)
        num1[i] = exp[i];

    op = exp[++i];
    i += 2;

    for (j=0; exp[i] != '\0'; i++)
    {
        num2[j] = exp[i];
        j++;
    }

    first = strToNum(num1);
    second = strToNum(num2);

    if (op == '+')
        return (first + second);
    if (op == '-')
        return (first - second);
    if (op == '*')
        return (first * second);
    else
        printf("error");
    return 0;
}
    
```

#include <stdio.h>  
#include <string.h>

110' סוף  
120 סוף  
num2, num1 ←

סעיף ד' (6) בקודות):

כתבו תכנית הקולטת מהמשתמש מחרוזת מהצורה:

num1 operation num2

לפי אותם חוקים שהוגדרו בסעיפים הקודמים (אין צורך לבדוק חוקיות קלט).  
התכנית מדפיסה את תוצאת פעולת החשבון. כלומר, אם נקלט:

34 - 12

יודפס בסיום התכנית:

34 - 12 = 22

שימו לב: בסעיף זה אסור להשתמש בפונקציה - gets().

6/6

```
int main()
{
    char s[24];
    char c != 0 ;
    int result, i = 0;
    c = getchar();
    while (c != '\n')
    {
        s[i] = c;
        c = getchar();
        i++;
    }
    s[i] = '\0';
    result = CalcExpression(s);
    puts(s);
    printf(" = %d", result);
    return 0;
}
```

#include <stdio.h>  
#include <string.h>

**שאלה 2 (35 נקודות)**

שאלה זו עליכם לנתח רשת חברתית של סטודנטים בקורס. לכל סטודנט יש "חבר-הכי-טוב" שהוא סטודנט אחר בקורס.

**סעיף א' (5 נקודות):**

הגדירו טיפוס חדש בשם Student שהוא מבנה המכיל את השדות הבאים:

1. מחרוזת המכילה את שם הסטודנט (עד 40 תווים): name
2. ציון הסטודנט בקורס (מספר שלם): grade
3. מספר שלם המיצג את מספרו הסידורי של סטודנט אחר שהוא החבר הכי טוב של הסטודנט: best\_friend

```

typedef struct
typedef struct t_student {
    char name[40];
    int grade;
    int best_friend;
} student;
    
```

"מעגל חברים" הוא קבוצת חברים שאם נתחיל מסטודנט כלשהו בה ונעבור ממנו לחבר-הכי-טוב שלו ושוב ממנו לחבר-הכי-טוב הבא וכן הלאה, נגיע בשלב כלשהו חזרה לסטודנט שממנו התחלנו.

לדוגמא, בתרשים הבא מופיע מערך המתאר קורס ובו 6 סטודנטים, המספר הסידורי של סטודנט הוא האינדקס של התא בו הוא מאוכסן במערך:

name: Avraham grade: 100 best_friend: 1	name: Yitzchaq grade: 90 best_friend: 3	name: Jacob grade: 80 best_friend: 0	name: Sara grade: 100 best_friend: 0	name: Rivka grade: 90 best_friend: 5	name: Rachel grade: 80 best_friend: 4
---	---	--	--	--	---

Index:            0                    1                    2                    3                    4                    5

אם נתחיל לדוגמא מהסטודנט Yitzchaq ונעבור ממנו לחבר-הכי-טוב שלו (אינדקס 3), נגיע ל-Sara. ממנה נמשיך לחבר-הכי-טוב שלה (אינדקס 0) ונגיע ל-Avraham וממנו נעבור לחבר-הכי-טוב שלו (אינדקס 1) ונגיע בחזרה ל-Yitzchaq. כלומר, הסטודנט Yitzchaq נמצא במעגל החברים באורך 3: [Avraham, Yitzchaq, Sara] בקורס המתואר בתרשים יש מעגל חברים נוסף (באורך 2): [Rivka, Rachel] שימו לב שהסטודנט Jacob לא שייך לאף מעגל חברים, כי אם נתחיל את המסלול ממנו לעולם לא נגיע אליו בחזרה.

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

סעיף ב' (10 נקודות):

כתבו פונקציה inCircle המקבלת מערך של סטודנטים בקורס (students), את גודל המערך (size) ואת ההמספר הסידורי של סטודנט מסויים במערך (studentIndex). אם הסטודנט שייך למעגל חברים, הפונקציה תחזיר 1, אחרת הפונקציה תחזיר 0.

מערך הסטודנטים מכיל את הסטודנטים על פי סדר מספרם הסידורי, כלומר course[i] מכיל את הסטודנט שמספרו הסידורי הוא i.

שימו לב

- לא יתכן שגודל מעגל חברים יהיה גדול מ-size.

```
int inCircle(Student students[], int size, int studentIndex)
{
    int next, curr, i = 0;
    curr = studentIndex;
    while (i < size)
    {
        next = (students[curr]).best_friend;
        if (next == studentIndex)
            return 1;
        else
            curr = next;
    }
    return 0;
}
```

✓  
10  
10

סעיף ג' (10 בקורות):

כתבו פונקציה printCircle המקבלת מערך של סטודנטים בקורס (students), מערך נוסף wasPrinted באותו הגודל, את גודל המערכים (size) ואת המספר הסידורי של סטודנט מסוים במערך (studentIndex).  
אם הסטודנט שייך למעגל חברים, הפונקציה תדפיס בשורה אחת את שמות כל החברים במעגל, ותעדכן את המערך wasPrinted בכך שהיא תשים 1 בתא ה-i אם שמו של הסטודנט ה-i הודפס. אם הסטודנט לא שייך למעגל חברים, הפונקציה לא תדפיס כלום ולא תשנה את המערך wasPrinted.

```
void printCircle(Student students[], int wasPrinted[],  
                 int size, int studentIndex) {  
    int curr = studentIndex;  
    if (0 == inCircle(students, size, studentIndex));  
        return i;  
    do {  
        printf("i's ", (students[curr]).name);  
        wasPrinted[curr] = 1;  
        curr = (students[curr]).best_friend;  
    } while (curr != studentIndex);  
}
```

#include <stdio.h>

✓  
10  
10



סעיף ד' (10 בקורות):

כתבו פונקציה printAllCircles המקבלת מערך של סטודנטים בקורס (students) ואת גודל המערך (size). הפונקציה מדפיסה את כל מעגלי החברים בקורס, כשמעגל חברים יודפס בדיוק פעם אחת. יש להדפיס כל מעגל חברים בשורה נפרדת.

הדרכה: השתמשו בפונקציה מסעיף ג' כדי להדפיס את המעגלים ולסמן אילו סטודנטים כבר הודפסו.

```
void printAllCircles(Student course[], int size) {  
    int i=0;  
    int wasPrinted[size] = {0};  
    while (i < size)  
    {  
        if (wasPrinted[i] == 0)  
            printCircle(students, wasPrinted, size, i);  
        i++;  
    }  
}
```

בצע קינאו -3

7  
10

שאלה 3 (32 נקודות)

30 פ

בשאלה זו נבנה מערכת פשוטה המאפשרת חישובים גיאומטריים. בבסיס המערכת נגדיר ייצוג של נקודה במרחב בעזרת קוארדינטות ה-x וה-y שלה. נתון המבנה Point המגדיר נקודה כזו.

```
typedef struct point_t
{
    double x, y;
} Point;
```

סעיף א' (5 נקודות):

בסעיף זה עליכם להגדיר משולש במישור תוך שימוש במבנה הנקודה שהוגדר כבר עבורכם. משולש במישור מיוצג בעזרת שלושת קודקודיו, כל קודקוד הוא נקודה במישור. הגדירו מבנה בשם Triangle המייצג משולש במישור. על המבנה להחזיק את ערכי שלושת הקודקודים המגדירים את המשולש וכן שדה נוסף שיאפשר יצירת רשימה מקושרת של משולשים.

```
typedef struct t-tria {
    Point A;
    Point B;
    Point C;
    struct t-tria *next;
} Triangle;
```



סעיף ב' (8 נקודות):

ממשו את הפונקציה

```
double triangleArea(Triangle *t)
```

הפונקציה מחשבת שטח של משולש. הפונקציה מקבלת מצביע למשולש כקלט ומחזירה את שטח המשולש. ניתן להניח כי הקלט תקין (t אינו NULL)

ניתן לחשב את שיטחו של משולש שקודקדיו הם A,B,C בעזרת הנוסחה:

$$area = \left| \frac{A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)}{2} \right|$$

כאשר  $A_x$  היא קואורדינטת ה-x של קודקוד A ו- $A_y$  היא קואורדינטת ה-y של קודקוד A. באופן דומה עבור קודקודים B ו-C.

```
double triangleArea(Triangle *t)
{
    double result;
    double a, b, c;
    a = (t->A->x) * (t->B->y - t->C->y);
    b = (t->B->x) * (t->C->y - t->A->y);
    c = (t->C->x) * (t->A->y - t->B->y);
    result = (a + b + c) / 2;
    if (result < 0)
        result = -1;
    return result;
}
```

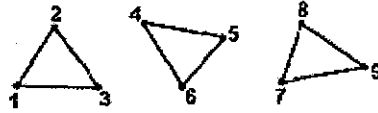
*Handwritten notes:*  $A \times B$  with an arrow pointing to the first line of code. A checkmark is at the bottom right of the code block.

כל הזכויות שמורות ©  
מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

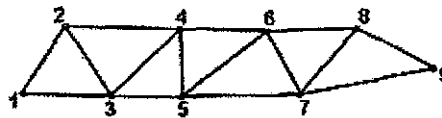
סעיף ג' (8 בקודות):

Mesh הוא מבנה גיאומטרי המכסה תחום במישור בעזרת משולשים. לא תתכן חפיפה כלשהי בין שטחי המשולשים ב mesh אולם ייתכן שיתוף של קודקודים וצלעות.

דוגמאות:



איור 1: mesh ללא שיתוף צלעות וקודקודים



איור 2: mesh עם שיתוף צלעות וקודקודים

נשמור mesh כרשימה מקושרת של משולשים.

ממשו את הפונקציה addTriangle המוסיפה משולש ל mesh. הפונקציה מקבלת שני פרמטרים: (1) head - מצביע לראש רשימת המשולשים, (2) t - מצביע למשולש החדש אותו נרצה להוסיף (ניתן להניח שהמצביע t אינו NULL). הפונקציה תחזיר מצביע לראש הרשימה לאחר ההוספה. שימו לב, אין משמעות לסדר המשולשים ב- mesh.

```
Triangle *addTriangle(Triangle *head, Triangle *t)
{
    t -> next = head;
    return t;
}
```

כל הזכויות שמורות ©  
מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

סעיף ד' (11 נקודות):

ממשו פונקציה רקורסיבית meshArea המחשבת את שטחו של mesh. הפונקציה מקבלת מצביע לרשימת משולשים ומחזירה את שטח ה-mesh.

```
double meshArea(Triangle *head)
{
    double result;
    if (head == null)
        return 0; // רשימה ריקה
    // (if (head->next == null)
    return triangleArea(head);
    result = triangleArea(head);
    result += meshArea(head->next);
    return result;
}
```

✓

## דף-עזר למבחן בתכנות

שימו לב: מטרת דף-העזר היא רק לחסוך לימוד בעל-פה של ההכרזות על הפונקציות המופיעות בו. יתכן שאין צורך להשתמש בפונקציות הללו כדי לפתור את המבחן, ויתכן שצריך להשתמש בפונקציות שנלמדו אך אינן מוזכרות בדף-העזר.

### פונקציות נבחרות מהספרייה string.h:

- int strcmp(const char \*s1, const char \*s2) (1) השוואת מחרוזות  
(2) השוואת n תווים ראשונים של מחרוזות  
int strncmp(const char \*s1, const char \*s2, int n)  
(3) חיפוש מופע ראשון של תו במחרוזת  
char\* strchr(const char \*s, char c)  
char\* strrchr(const char \*s, char c) (4) חיפוש מופע אחרון של תו במחרוזת  
(5) חיפוש מופע ראשון של מחרוזת במחרוזת אחרת  
char\* strstr(const char \*s, const char \*to\_find)  
char\* strcpy(char \*dest, const char \*src) (6) העתקת מחרוזת  
(7) העתקת n תווים ראשונים של מחרוזת  
char\* strncpy(char \*dest, const char \*src, int n)  
char\* strcat(char \*dest, const char \*src) (8) שירשור מחרוזות  
(9) שירשור n תווים ראשונים ממחרוזת  
char\* strncat(char \*dest, const char \*src, int n)  
int strlen(const char \*s) (10) מציאת אורך מחרוזת

### פונקציות נבחרות מהספרייה math.h:

- double sqrt(double x) (1) שורש ריבועי  
double pow(double base, double exponent) (2) חזקה  
int abs(int n) (3) ערך מוחלט (שלמים)  
double fabs(double x) (4) ערך מוחלט (ממשיים)  
double ceil(double x) (5) עיגול למעלה  
double floor(double x) (6) עיגול למטה

### פונקציות נבחרות מהספרייה ctype.h:

- int isalnum(int c) (1) האם תו הוא אלפא-נומרי (a-z, A-Z, 0-9)  
int isalpha(int c) (2) האם תו הוא אות (a-z, A-Z)  
int isdigit(int c) (3) האם תו הוא סיפרה (0-9)  
int islower(int c) (4) האם תו הוא אות קטנה (a-z)  
int isupper(int c) (5) האם תו הוא אות גדולה (A-Z)  
int isspace(int c) (6) האם תו הוא whitespace (רווח, '\n' או '\t')  
int tolower(int c) (7) הפיכת אות גדולה לאות קטנה  
int toupper(int c) (8) הפיכת אות קטנה לאות גדולה

### פונקציות נבחרות מהספרייה stdlib.h:

- void\* malloc(unsigned int size) (1) הקצאת זיכרון דינאמית  
(2) הקצאת זיכרון דינאמית ואיפוס הזיכרון שהוקצה  
void\* calloc(unsigned int nelements, unsigned int elementSize)  
void\* realloc(void \*pointer, int size) (3) הקצאה מחודשת של זיכרון  
void free(void \*pointer) (4) שחרור זיכרון שהוקצה דינאמית

כל הזכויות שמורות ©

מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

סמסטר א' תשס"ט  
מועד : א' 16/02/2009  
משך הבחינה : שלוש שעות  
אין להשתמש בחומר עזר

## בחינה בקורס תכנות ב- C

מרצים : אלון לרנר, עופר פסטרנק

מתרגלים : מתי שמרת, נעמה מאיר, אסף היימן, אפרת משיח,  
בן זנדבנק, גיא קרליבך

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספרייה (זכרו לבצע `#include` מתאים).
- בסוף המבחן נמצא דף-עזר.
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.
- מומלץ לקרוא כל שאלה עד סופה, על כל סעיפיה, לפני תחילת הפתרון.

בהצלחה!

### לשימוש הבודקים:

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	ציון

## שאלה 1 (33 נקודות)

"צופן החלפה" היא שיטת הצפנה שבה כל אות בשפה מוחלפת באות אחרת, וסדר ההחלפה מתואר על ידי "מפתח החלפה".

על מנת ש"מפתח החלפה" יהיה חוקי עליו:

- להכיל 26 תווים שונים זה מזה.
- התווים חייבים להיות אותיות קטנות באנגלית (lower-case).

דוגמאות למפתחות החלפה חוקיים:

```
bcdjqsaklmnopeirfghtuvwxyz  
abcdefghijklmnopqrstuvwxy
```

בעת פענוח מסר, התו המוצפן יוחלף על ידי התו שמיקומו באלף-בית שווה למיקום של התו המוצפן במפתח. כך למשל, אם התו המוצפן x מופיע במקום השלישי במפתח הוא יוחלף בתו c.

### דוגמה:

בעזרת המפתח:

```
bcdjqsaklmnopeirfghtuvwxyz
```

המסר המוצפן:

```
"tklh lh b hqdggt"
```

יתורגם ל:

```
"this is a secret"
```

### שימו לב:

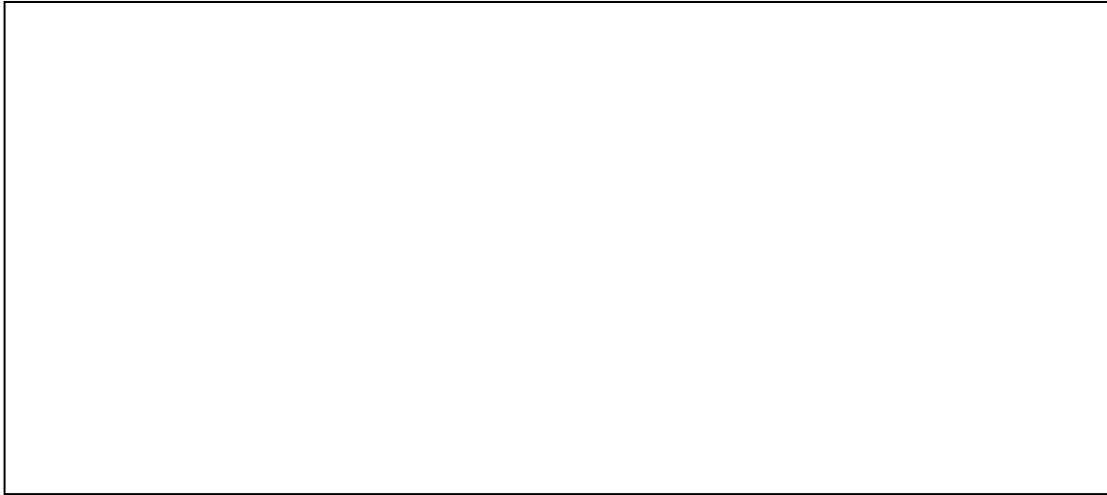
תווים שאינם אותיות קטנות (אותיות גדולות, רווחים, ספרות סימני פיסוק וכו') אינם מוצפנים.

### סעיף א' (6 נקודות):

עליכם לממש פונקציה אשר מחפשת תו מסוים (tav) במחרוזת (str). על הפונקציה להחזיר את המיקום (אינדקס התא) בו מופיע התו לראשונה במחרוזת, אחרת תחזיר הפונקציה -1. **בסעיף זה אין להיעזר בפונקציות ספרייה.**

```
int charInStr(char *str,char tav)
{
    int i=0;
    /* Loop until the end of the string */
    while (str[i] != '\0')
    {
        /* Found the tav */
        if (str[i] == tav)
            return i;
        i++;
    }
    /* Didn't find the tav */
    return -1;
}
```





### סעיף ב' (9 נקודות):

עליכם לממש פונקציה המקבלת מחרוזת (keyStr) המייצגת "מפתח החלפה".  
על הפונקציה להחזיר 1 אם המחרוזת מייצגת "מפתח החלפה" חוקי, אחרת תחזיר 0.

#### תזכורת:

מפתח החלפה חוקי מכיל 26 תווים שונים זה מזה, כאשר כולם אותיות קטנות באנגלית.

```
int isValidKey( char *keyStr )
{
    int i,tmp;

    /* Wrong length of key */
    if (strlen(keyStr) != 26)
        return 0;

    for(i=0;i<26;i++)
    {
        /* Search the string for the i'th letter */
        tmp = charInStr(keyStr,'a'+i);

        /* letter is missing */
        if (tmp<0)
            return 0;
    }

    return 1;
}
```

## סעיף ג' (9 נקודות):

עליכם לממש פונקציה המקבלת "מפתח החלפה" (keyStr) ומחרוזת **מוצפנת** (msg).  
על הפונקציה לתרגם את המשפט המוצפן חזרה לאנגלית ולשמור אותו **במקום** המחרוזת המוצפנת.

### שימו לב:

המחרוזת עשויה להכיל תווים שהם אינם אותיות קטנות (lower-case) באנגלית. יש לתרגם אך ורק את התווים שהם אותיות קטנות באנגלית, שאר התווים יישארו כפי שהם.  
**ניתן להניח שמפתח ההחלפה חוקי.**

### דוגמה:

על ידי המפתח:

```
bcdjqsaklmnopeirfghtuvwxyz
```

המסר המוצפן:

```
"Aaget 25 wbltlea le oidbtlie 45b!!"
```

יתורגם ל:

```
"Agent 25 waiting in location 45a!!"
```

```
void decrypt( char *keyStr, char *str)
{
    int ind;

    /* Loop until the end of the string */
    while(*str!='\0')
    {
        /* Search the key for the character */
        ind = charInStr(keyStr, *str);

        /* Found the character -> Decrypt it */
        if (ind!=-1)
            *str= 'a'+ind;

        str++;
    }
}
```

**סעיף ד' (9 נקודות):**

- כתבו תוכנית אשר קולטת מהמשתמש "מפתח החלפה" ולאחר מכן מחרוזות מוצפנת. על התכנית:
1. לבדוק שהמפתח חוקי. אם הוא אינו חוקי יש להדפיס הודעה מתאימה ולסיים את התוכנית.
  2. לפענח את המשפט המוצפן ולהדפיס את המשפט המפוענח למסך.

**ניתן להניח ש:**

- המחרוזת המוצפנת תופיע בשורה נפרדת מ"מפתח ההחלפה".
- מספר התווים במחרוזת המוצפנת לא עולה על 100.
- המשתמש הקליד את המחרוזת בשורה אחת (כלומר, התו '\n' מסיים את המחרוזת).

```
#include <stdio.h>
#define MAX_SIZE 101
int main()
{
    char key[27], sentence[MAX_SIZE];
    int i=0;
    /* Read the key */
    while (i<26)
    {
        key[i] = getchar();
        if (key[i] == '\n')
            break;
        i++;
    }
    /* The key is too long */
    if (key[i] != '\n' )
    {
        printf("not a valid key\n");
        return 0;
    }
    /* Convert it to a string */
    key[i] = '\0';
    /* Is the key valid? */
    if (!isValidKey(key))
    {
        printf("not a valid key\n");
        return 0;
    }
    i=0;
    /* Read the sentence */
    while (i<MAX_SIZE)
    {
        sentence[i] = getchar();
        if (sentence[i] == '\n')
            break;
        i++;
    }
    /* Convert it to a string */
    sentence[i] = '\0';
    /* Decrypt and print */
    decrypt(key,sentence);
    printf("%s\n",sentence);
    return 0;
}
```

## שאלה מס' 2 (33 נקודות)

פולינום הוא ביטוי מהצורה:  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$   
שבו  $x$  הוא משתנה ו- $a_i$  הם **מקדמי הפולינום** (coefficients).  
ה- $n$  הגבוה ביותר שעבורו  $a_n \neq 0$  נקרא **המעלה של הפולינום** (degree).

### לדוגמה:

- הביטוי  $3x^3 - 5x + 4$  הוא פולינום ממעלה שלוש שמקדמיו הם {3.0, 0.0, -5.0, 4.0}.
- הביטוי  $x^4 + 0.5x^2 + 4.2x$  הוא פולינום ממעלה ארבע שמקדמיו הם {1.0, 0.0, 0.5, 4.2, 0.0}.

## סעיף א' (7 נקודות)

הגדירו מבנה (struct) בשם Polynom. המבנה יכול את מעלת הפולינום (degree) וכן את מקדמי הפולינום (coefficients), שהם מספרים ממשיים.

ניתן להניח שמספר המקדמים של הפולינום קטן ממספר קבוע **MAX\_DEGREE**.

```
#define MAX_DEGREE 10

typedef struct tmp{
    int degree;
    double coeff[MAX_DEGREE];
} Polynom;
```

## סעיף ב' (13 נקודות):

חיבור פולינומים מתבצע ע"י חיבור המקדמים המתאימים בשני הפולינומים.

### לדוגמה:

חיבור הפולינום:  $3x^3 - 5x + 4$  עם הפולינום:  $x^4 + 0.5x^2 + 4.2x$   
יצור את הפולינום:  $x^4 + 3x^3 + 0.5x^2 - 0.8x + 4$

ממשו פונקציה המקבלת מצביעים לשלושה פולינומים (p1, p2, result). על הפונקציה לחבר בין p1 ל-p2 ולשמור את תוצאת החיבור בפולינום result.

```

void add( const Polynom *p1,
          const Polynom *p2,
          Polynom *result )
{
    int i,min_deg;
    Polynom tmp;
    /* Who has the bigger degree? */
    if (p2->degree > p1->degree)
    {
        /* Copy the polynomial with the bigger degree */
        *result=*p2;
        /* Update the min degree variable */
        min_deg = p1->degree;
        /* Copy the polynomial with the smaller degree */
        tmp = *p1;
    }
    else
    {
        *result=*p1;
        min_deg = p2->degree;
        tmp = *p2;
    }

    /* Update the coefficients */
    for (i=0; i<=min_deg; i++)
        result->coeff[i] += tmp.coeff[i];

    /* Loop backwards across the coefficients */
    for (i=result->degree; i>=0; --i)
        /* The biggest non-zero coefficient -> update */
        if (result->coeff[i] != 0)
        {
            result->degree = i;
            break;
        }

    /* All the coefficients equal 0 */
    if (i<0)
        result->degree = 0;
}

```

### סעיף ג' (13 נקודות):

כתבו תוכנית שקולטת שני פולינומים מהמשתמש, מחברת ביניהם ומדפיסה את מקדמי פולינום התוצאה אל המסך. עבור כל פולינום המשתמש מקליד את דרגת הפולינום ולאחר מכן את מקדמיו כך שהמקדם הראשון הוא  $a_n$  והאחרון הוא  $a_0$  (משמאל לימין). ניתן להניח כי דרגת הפולינום קטנה מהקבוע MAX\_DEGREE.

#### לדוגמה:

עבור הפולינומים:  $x^4 + 0.5x^2 + 4.2x - 3x^3 - 5x + 4$  ו-  $3x^3 - 5x + 4$ :  
 הקלט יהיה:

```

3
3.0 0.0 -5.0 4.0
4
1.0 0.0 0.5 4.2 0.0
1.0 3.0 0.5 -0.8 4.0

```

ואילו הפלט הוא:

```
#include <stdio.h>

int main()
{
    int i;
    /* Define the polynoms */
    Polynom a,b,c;
    /* Read the first polynom */
    scanf("%d",&a.degree);
    for (i=a.degree;i>=0;i--)
        scanf("%lf",&a.coeff[i]);
    /* Read the second polynom */
    scanf("%d",&b.degree);
    for (i=b.degree;i>=0;i--)
        scanf("%lf",&b.coeff[i]);
    /* Add the polynoms */
    add(&a,&b,&c);
    /* Print the polynoms */
    for (i=c.degree;i>=0;i--)
        printf("%g ",c.coeff[i]);
    return 0;
}
```

## שאלה 3 (34 נקודות)

### סעיף א' (5 נקודות):

הגדירו מבנה בשם `ListNode` שישמש לבניית רשימה מקושרת, כאשר בכל תא של הרשימה נשמר ערך ממשי.

```
typedef struct node_t
{
    double data;
    struct node_t *next;
}
ListNode;
```

### סעיף ב' (7 נקודות):

עליכם לממש פונקציה המוסיפה איבר חדש בעל הערך `value` בסוף רשימה מקושרת של איברי `ListNode`. ראש הרשימה נתון על ידי המצביע `head`. הפונקציה מחזירה מצביע לראש הרשימה המעודכנת.

```
ListNode *addToList( ListNode *head, double value )
{
    ListNode* tail = head;

    /* Allocate a new ListNode */
    ListNode *newNode = (ListNode*)malloc(sizeof(ListNode));
    /* Allocation failed -> Exit the program */
    if (newNode==NULL)
        exit(1);

    /* Update the node's values */
    newNode->data = value;
    newNode->next = NULL;

    /* The list is empty */
    if (head == NULL)
        return newNode;

    /* Loop until the end of the list */
    while (tail->next != NULL)
        tail = tail->next;

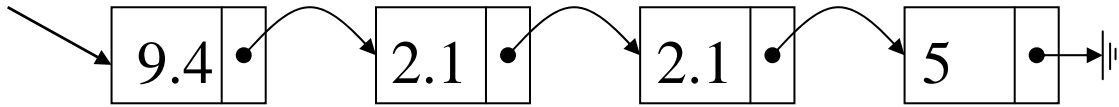
    /* Add the new node */
    tail->next = newNode;

    /* Return the head of list */
    return head;
}
```

**סעיף ג' (11 נקודות):**

עליכם לממש פונקציה המקבלת מצביע לראש רשימה מקושרת (head) ומספר (value). הפונקציה מחזירה את הערך הקטן ביותר שמופיע ברשימה שגדול מ-value. אם לא קיים ערך מתאים אז הפונקציה תחזיר את ערכו של value.

**דוגמה:**



עבור הרשימה הזו:

- בקריאה לפונקציה עם הערך 4 יוחזר הערך 5.
- בקריאה לפונקציה עם הערך 10.3 יוחזר הערך 10.3.

```
double getNumber( ListNode *head, double value )
{
    /* Define minval */
    double minval = value;
    /* Define a flag -> 1 means we found a candidate */
    int flag=0;

    /* Loop across the list */
    while(head!=NULL)
    {
        /* data is bigger AND we haven't found a candidate yet or
        data is a better candidate */
        if (head->data > value && (!flag || head->data < minval) )
        {
            /* Update minval and the flag */
            minval= head->data;
            flag = 1;
        }
        /* Move up the list */
        head=head->next;
    }

    /* Didn't find a candidate -> should return value */
    if (flag==0)
        minval = value;

    return minval;
}
```



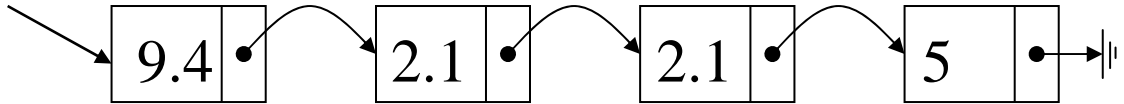
**סעיף ד' (11 נקודות):**

עליכם לממש פונקציה המקבלת מצביע (head) לראש רשימה מקושרת המכילה מספרים חיוביים (לא בהכרח שונים). הפונקציה ומחזירה רשימה חדשה בעלת התכונות הבאות:

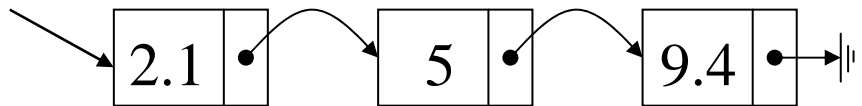
- היא מכילה את כל המספרים שהיו ברשימה הישנה.
- המספרים ממוינים מהקטן לגדול.
- כל מספר מופיע פעם אחת בלבד ברשימה.

**דוגמה:**

מהרשימה הבאה:



תיווצר הרשימה:



```

ListNode *sortList(ListNode *head)
{
    /* This will point to the new list */
    ListNode *new_list = NULL;
    /* Initialize minval to 0 */
    double minval = 0, curval;

    /* Loop across the list */
    while (head!=NULL)
    {
        /* Get smallest number bigger than minval */
        curval = getNumber(head,minval);
        /* Found a duplicate -> skip it */
        if (curval==minval)
            continue;

        /* Update the minimal value */
        minval = curval;
        /* Add the number to the new list */
        new_list = addToList(head, minval);
    }

    /* Return the pointer to the new list */
    return new_list;
}
    
```

מס' מחברת : 77 מס' ת.ז. : 300296894

סמסטר א' תשס"ח  
מועד : א' 3/2/2008  
משך הבחינה : שלוש שעות  
אין להשתמש בחומר עזר



## בחינה בקורס תכנות ב-C

מרצים : אלון לרנר, אמתי ערמון, עופר פסטרנק

מתרגלים : יפתח גמזו, בן זנדבנק, אסף זריצקי,  
אילונה קיפר, גיא קרליבך, מתי שמרת

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספרייה (זכרו לבצע #include מתאים).
- בסוף המבחן נמצא דף-עזר.
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.
- מומלץ לקרוא כל שאלה עד סופה, על כל סעיפיה, לפני תחילת הפתרון.

בהצלחה!

לשימוש הבודקים:

			תעודת זהות
			מספר מחברת
12	15	7	שאלה 1 34
11	11	8	שאלה 2 30
11	11	11	שאלה 3 33
			ציון 97

34

שאלה מס' 1 (34 נקודות)

סעיף א' (7 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void arr_copy(int a[], int b[], int n)
```

הפונקציה מקבלת שני מערכים המכילים מספרים שלמים, ומספר שלם אי-שלילי  $n$  (הניחו שזה אכן מה שהיא מקבלת). על הפונקציה להעתיק את  $n$  הערכים הראשונים במערך  $a$  ל- $n$  המקומות הראשונים במערך  $b$ . הניחו ששני המערכים הם בגודל  $n$  לפחות.

```
void arr_copy(int a[], int b[], int n)
{
    int i;
    for (i=0; i<n; i++)
        b[i]=a[i];
}
```

✓  
7

סעיף ב' (15 נקודות):

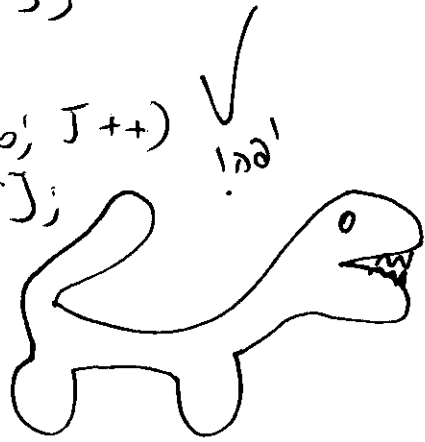
כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void merge( int a[], int size_a, int b[], int size_b,
            int *dest );
```

הפונקציה מקבלת שני מערכים של מספרים שלמים שממויינים בסדר לא-יורד (כלומר מקטן לגדול). כמו-כן, היא מקבלת את גדלי המערכים, ומערך נוסף dest. על הפונקציה להעתיק בצורה ממויינת את הערכים משני המערכים, a ו-b, לתוך המערך dest. כלומר, בסיום הפונקציה המערך dest צריך להכיל את כל  $size\_a + size\_b$  הערכים הללו, והוא צריך להיות ממויין בסדר לא-יורד (מקטן לגדול). הניתו כי יש במערך dest מספיק מקום עבור כל הערכים המועתקים.

```
void merge(int a[], int size_a, int b[], int size_b, int *dest)
{
    int I=0, J=0;
    while (I != size_a && J != size_b)
    {
        if (a[I] < b[J])
        {
            dest[I+J] = a[I];
            I++;
        }
        else
        {
            dest[I+J] = b[J];
            J++;
        }
    }
    if (I != size_a)
        for (I=I; I < size_a; I++)
            dest[I+J] = a[I];
    if (J != size_b)
        for (J=J; J < size_b; J++)
            dest[I+J] = b[J];
}
```

15/15



סעיף ג' (12 נקודות):

לפניכם הפונקציה הרקורסיבית `merge_sort`, שממיינת את המערך `a`, שגודלו `size`. הפונקציה משתמשת בפונקציות מסעיפים א' ו-ב'. עליכם להשלים את הקטעים החסרים בפונקציה, בתוך המסגרות (כל מסגרת היא שורת קוד).

הערה: הניחו שהקצאת הזיכרון הצליחה.

12

```
#include <stdlib.h>
void merge_sort(int a[], int size)
{
    int *left, *right, *temp;

    if ( size <= 1 )
    {
        return;
    }

    left = a;
    right = a + size/2;

    merge_sort( left, size/2 );
    merge_sort( right, size/2+(size%2!=0) );

    temp = (int *) malloc( size * sizeof(int) );
    merge( left, size/2, right, size/2+(size%2!=0), temp );

    arr_copy( temp, a, size );

    free(temp);
}
```

שאלה מס' 2 (33 נקודות)

סעיף א' (11 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
void add_digits( char digit1, char digit2, char digit3,  
                char *tens, char *ones);
```

הפונקציה מקבלת שלושה משתנים המכילים תווי ספרות, ושני מצביעים למשתנים מסוג char (הניחו שזה אכן מה שהיא מקבלת). היא מחשבת את סכום הספרות המיוצגות ע"י שלושת תווי הספרות שהיא קיבלה (ראו דוגמא בהמשך), ושמה את התו המייצג את ספרת האחדות של הסכום במשתנה ש-ones מצביע עליו, ואת התו המייצג את ספרת העשרות של הסכום במשתנה ש-tens מצביע עליו.

לדוגמא: אם digit1='8', digit2='6', digit3='0' אז המשתנה ש-ones מצביע עליו יכיל את התו '4'.  
והמשתנה ש-tens מצביע עליו יכיל את התו '1'.

הערה: ניקוד מלא ינתן למי שלא יכלול בפתרון שלו ערכי ASCII ספציפיים של תווים, אלא רק יסתמך על סדר התווים בטבלת ה-ASCII.

```
void add_digits(char digit1, char digit2, char digit3,  
                char *tens, char *ones)  
{  
    int sum;  
  
    sum = (digit1 - '0') + (digit2 - '0') + (digit3 - '0');  
    *tens = '0' + sum / 10;  
    *ones = '0' + sum % 10; ✓  
}
```

(ג2')

11/11

סעיף ב' (11 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void add_nums(char num1[], char num2[], char result[]);
```

הפונקציה מקבלת שתי מחרוזות ומערך. המחרוזות num1 ו-num2 מייצגות מספרים שלמים חיוביים בעלי אותו מספר ספרות (ראו דוגמא למטה). על הפונקציה לשים במערך result את המחרוזות שמייצגת את תוצאת החיבור בין המספר המיוצג ע"י num1 והמספר המיוצג ע"י num2.


לדוגמא, אם num1 = "4832" ו-num2 = "5428" אז בסיום הפונקציה: result = "10260"

הערות:

- המספרים שמיוצגים על-ידי המחרוזות עשויים להיות גדולים יותר מהמספר המקסימלי שניתן לייצג על-ידי int או double (לא ניתן להניח שאפשר לשמור אותם ב-int או double).
- הניחו שמספר הספרות בשני המספרים הוא זהה, ובשני המספרים יש לפחות ספרה אחת.
- הניחו שהמחרוזות מכילות אך ורק תווים המייצגים ספרות.
- הניחו שבמערך result יש מספיק מקום על מנת לשמור את התוצאה.
- מותר שהספרה הראשונה תהיה אפס (למשל "05678").

```
#include <string.h>
void add_nums(char num1[], char num2[], char result[])
{
    char tens=0, ones=0;
    int i, len;
    len = strlen(num1);
    for (i=len-1; i>=0; i--)
    {
        add_digits(num1[i], num2[i], tens, &tens, &ones);
        result[i+1] = ones;
    }
    result[0] = tens;
    result[len+1] = '\0';
}
```

✓  
אפס  
13/14



סעיף ג' (11 בקודות):

כיתבו תוכנית הקולטת שתי מתרזות המייצגות מספרים שלמים, שיש להם אותו מספר ספרות (הניחו שמס' הספרות הוא לכל היותר 1000). על התוכנית לחשב את סכום המספרים האלה, ולהדפיס את התוצאה. כמו-כן, החערות המופיעות בסוף סעיף ב' הן תקפות גם לסעיף זה. ניתן להניח שהקלט תקין ושכל מתרזות מוכנסת בשורה נפרדת.

```
#include <stdio.h>
#include <string.h>
#define SIZE 1001
void input(char str[], size_t size); /* מה נפוהור זמא על שאר הסוקנ'
int main()                          /* מס' הסוקסטימ' הקודמ' */
{
    int i=0;
    char num1[SIZE], num2[SIZE], result[SIZE];

    input(num1, SIZE);
    input(num2, SIZE);
    add_nums(num1, num2, result);
    for(i=0; result[i]!='\0'; i++)
        printf("%d", result[i]);
    printf("\n");
    return 0;
}

void input(char str[], size_t size)
{
    int i=0;
    char tav;
    tav=getchar();
    while(tav!='\n' && i < size)
    {
        str[i]=tav;
        i++;
        tav=getchar();
        tav=getchar();
    }
}
```

8/11



מס' מחברת : 77 מס' ת.ז. : 300296294

33

**שאלה מס' 3 (33 נקודות)**

שאלה זו מתייחסת למבנה Item המוגדר באופן הבא:

```
typedef struct Item_t
{
    int data;
    struct Item_t *next;
} Item;
```

**סעיף א' (11 נקודות):**

כיתבו פונקציה בעלת המפרט הבא:

```
int in_list (Item *head, int num);
```

הפונקציה מקבלת מצביע לראש רשימה מקושרת של מבני Item ומספר שלם. עליה להחזיר 1 אם המספר הזה נמצא ברשימה, ואחרת עליה להחזיר 0.

```
#include <stdlib.h> /* מאתר ומשתמש ב-NULL */
int in_list (item *head, int num)
{
    item *find = head;
    while (find != NULL)
    {
        if (find->data == num)
            return 1;
        find = find->next;
    }
    return 0;
}
```

11

**סעיף ב' (11 נקודות):**

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Item *list_intersect (Item *head1, Item *head2);
```

הפונקציה מקבלת מצביעים לתחילת שתי רשימות-מקושרות של מבני Item. עליה ליצור רשימה מקושרת חדשה, שתכלול רק את המספרים שנמצאים גם ברשימה ש- head1 מצביע עליה וגם ברשימה ש- head2 מצביע עליה. בסיום עליה להחזיר מצביע לראש הרשימה החדשה.

**הערות:**

- אין חשיבות לסדר שבו המספרים יופיעו ברשימה החדשה.
- ניתן להניח שברשימות המקוריות כל מספר מופיע רק פעם אחת.
- על הרשימות המקוריות להישאר ללא שינוי.



```
#include <stdlib.h>
item *list_intersect (item *head1, item *head2)
{
    item *cmp1 = head1, *new_item, *new_head = NULL;
    int found = 0;
    If (head1 == NULL || head2 == NULL)
        return NULL;
    while (cmp1 != NULL)
    {
        found = in_list (head2, cmp1->data);
        if (found)
        {
            new_item = (item *) malloc (sizeof (item));
            if (new_item == NULL)
            {
                printf ("out of memory");
                exit (1);
            }
            *new_item = *cmp1;
            new_item->next = new_head;
            new_head
            new_head = new_item;
        }
        cmp1 = cmp1->next;
    }
    return new_head;
}
```

(stdlib.h)

סעיף ג' (11 בקורות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
Item *join_lists(Item *head1, Item *head2);
```

הפונקציה מקבלת מצביעים לתחילת שתי רשימות מקושרות של מבני Item. עליה לצרף את הרשימה ש-head2 מצביע עליה לסוף הרשימה ש-head1 מצביע עליה (כלומר האיבר האחרון ברשימה ש-head1 מצביע עליה צריך להצביע לאיבר ש-head2 מצביע עליו). היא מחזירה מצביע לראש הרשימה המאוחדת.

```
#include <stdlib.h> /*מאחר ומשתמש ב NULL*/
Item *join_lists (Item *head1, Item *head2)
{
    Item *last = head1;
    If (head1 == NULL)
        return head2;
    If (head2 == NULL)
        return head1;

    while (last->next != NULL)
        last = last->next;
    last->next = head2;
    return head1;
}
```

11

## דף-עזר למבחן בתכנות – סמסטר א' תשס"ח

שימו לב: מטרת דף-העזר היא רק לחסוך לימוד בעל-פה של ההכרזות על הפונקציות המופיעות בו. יתכן שאין צורך להשתמש בפונקציות הללו כדי לפתור את המבחן, ויתכן שצריך להשתמש בפונקציות שנלמדו אך אינן מוזכרות בדף-העזר.

### פונקציות נבחרות מהספרייה string.h: (1) השוואת מחרוזות

- int strcmp(const char \*s1, const char \*s2); (2) השוואת n תווים ראשונים של מחרוזות
- int strncmp(const char \*s1, const char \*s2, int n); (3) חיפוש הופעה ראשונה של תו במחרוזת
- char\* strchr(const char \*s, char c); (4) חיפוש הופעה אחרונה של תו במחרוזת
- char\* strrchr(const char \*s, char c); (5) חיפוש הופעה ראשונה של מחרוזת במחרוזת אחרת
- char\* strstr(const char \*s, const char \*to\_find); (6) העתקת מחרוזת
- char\* strcpy(char \*dest, const char \*src); (7) העתקת n תווים ראשונים של מחרוזת
- char\* strncpy(char \*dest, const char \*src, int n); (8) שירשור מחרוזות
- char\* strcat(char \*dest, const char \*src); (9) שירשור n תווים ראשונים ממחרוזת
- char\* strncat(char \*dest, const char \*src, int n); (10) מציאת אורך מחרוזת
- int strlen(const char \*s);

### פונקציות נבחרות מהספרייה math.h:

- double sqrt(double x); (1) שורש ריבועי:
- double pow(double base, double exponent); (2) חזקה:
- int abs(int n); (3) ערך מוחלט:
- double fabs(double x); (4) ערך מוחלט בממשיים:
- double ceil(double x); (5) עיגול למעלה:
- double floor(double x); (6) עיגול למטה:

### פונקציות נבחרות מהספרייה stdlib.h: (1) הקצאת זיכרון דינאמית:

- void\* malloc(unsigned int size);
- (2) הקצאת זיכרון דינאמית ואיפוס הזיכרון שהוקצה:
- void\* calloc(unsigned int n\_elements, unsigned int element\_size);
- (3) הקצאה מחודשת של זיכרון:
- void\* realloc(void \*pointer, unsigned int size);
- (4) שחרור זכרון שהוקצה דינאמית:
- void free(void \*pointer)

סמסטר ב' תשס"ז  
 מועד: א' 10/7/2007  
 משך הבחינה: שלוש וחצי שעות  
אין להשתמש בחומר עזר

## פתרונות

### שאלה מס' 1 (33 נקודות)

#### סעיף א' (8 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:  
`int char_in_string(const char str[], char tav);`

הפונקציה מקבלת מחרוזת ותו, (הניחו שזה אכן מה שמועבר). במידה והתו מופיע במחרוזת תחזיר הפונקציה את מיקומו (האינדקס שלו). אחרת תחזיר -1.

דוגמה: עבור המחרוזת **vacation** והתו **'c'** תחזיר הפונקציה **2**. עבור התו **'k'** תחזיר -1.

```
int char_in_string(const char str[], char tav)
{
    int ind=0;
    while (str[ind]!='\0')
    {
        if (str[ind]==tav)
            return ind;
        ind++;
    }
    return -1;
}
```

#### סעיף ב' (15 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

`int scrabble(char str[], char letters[], int size);`

הפונקציה מקבלת מחרוזת בשם `str`, מערך של `char-ים` המכיל תווים בשם `letters`, ושם המיצג את אורך המערך (הניחו שהקלט תקין ומכיל רק אותיות ב-lower case). הפונקציה צריכה לבדוק האם בעזרת האותיות שבמערך `letters`, ניתן לבנות את המילה שבמחרוזת `str`.

שימו לב שבמידה ואות מסוימת חוזרת מספר פעמים במחרוזת, היא צריכה להופיע גם מספר פעמים במערך (בדיוק כמו בשבץ נא). הפונקציה מחזירה 1 במידה וניתן לבנות את המילה, ו-0 אחרת.

לדוגמה, עבור המחרוזת "barbara" והאותיות `d,f,a,b,r,b,r,k,a,a` תחזיר הפונקציה 1. ועבור האותיות `d,f,a,b,r,b,r,k,a` תחזיר הפונקציה 0. (כי צריך שלושה תווי `a`).

```
int scrabble(char str[], char letters[], int size)
{
    int i;
    char *tmp = (char *)malloc((size+1)*sizeof(char));
    if (tmp==NULL)
    {
        printf("Allocation Failure");
        exit(1);
    }
    // It is not desired to change the original letters array, so
    // we duplicate it.
    for (i=0;i<size;i++)
        tmp[i] = letters[i];
    // The array is converted to a string
    tmp[size] = '\0';

    // In case there are not enough letters to build the word
    if (strlen(str)>size)
        return 0;

    while (*str != '\0')
    {
        i= char_in_string(tmp,*str);
        if (i== -1)
            return 0;

        // Changing tmp[i] to a non letter char was also acceptable
        tmp[i] = tmp[size-1];
        tmp[size-1] = '\0';
        size--;
        str++;
    }
    return 1;
}
```

**סעיף ג' (10 נקודות):**

כתבו **תוכנית** שקולטת עשרה תווים ועשר מילים (מופרדות ברווחים), ומדפיסה את המילה הארוכה ביותר מבין המילים שניתן לבנות בעזרת התווים. במידה ואף אחת מהמילים אינה ניתנת לבניה, תדפיס התכנית את המילה failed.

ניתן להניח שהקלט תקין, ושהמילים קצרות מאורך קבוע כלשהוא.

ניתן להניח שהמילה failed לא תתקבל כקלט.

אם יש יותר ממילה אחת באורך המקסימלי על התכנית להדפיס אחת מהן.

ניתן להניח שהמשתמש יודע מה עליו לספק לתכנית (כלומר, אין צורך בהדפסות שמבקשות קלט או מסבירות למשתמש מה עליו להקליד)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX_SIZE 15

int main()
{
    char str[MAX_SIZE+1]; // Leave room for the '\0'
    char tav[10];
    char tmp[11] = "failed";

    int i, sz=0;
    for (i=0;i<10;i++)
        tav[i] = getchar();

    for (i=0;i<10;i++)
    {
        scanf("%MAX_SIZEs",str);
        if (scrabble(str,tav,10))
        {
            // Remember the longest word
            if (strlen(str)>sz)
            {
                sz = strlen(str);
                strcpy(tmp,str);
            }
        }
    }
    printf("%s\n",tmp);
    return 0;
}
```

**שאלה מס' 2 (33 נקודות)**

ניתן לחשב קירוב לנגזרת עבור כל ערך בסדרת מספרים על ידי חישוב ההפרש בין שני הערכים הסמוכים לו. כלומר, הנגזרת,  $d$ , במקום ה- $i$  בסדרת המספרים  $X$  באורך  $n$  מוגדרת בעזרת הנוסחה הבאה:

$$d(i) = 0 \quad \text{עבור } i=0 \text{ או } i=n-1$$

$$d(i) = (X(i+1) - X(i-1)) / 2 \quad \text{עבור } 0 < i < n-1$$

**סעיף א' (10 נקודות):**

כתבו פונקציה בעלת המפרט הבא:

```
double *derivative(const double vec[],int size);
```

הניחו ש- $vec$  הינו מערך של ממשיים המייצג סדרת מספרים ו- $size$  האורך שלו. הפונקציה תחשב את הנגזרת עבור כל נקודה בסדרה לפי הנוסחה. הפונקציה תחזיר מערך באורך  $size$  ובו, בכל אינדקס, ערך הנגזרת עבור המקום המתאים במערך  $vec$ .

דוגמה:

עבור  $vec$ :

2 2.3 2.4 2.1 2 5 5.2 5.3 5 5.3 4.5

יוחזר המערך

0 0.2 -0.1 -0.2 1.45 1.6 0.15 -0.1 0 -0.25 0

למשל, החישוב עבור הערך השלישי במערך,

$$d(2) = (X(3)-X(1)) / 2 = (2.1-2.3)/2 = -0.1$$

```
double *derivative(const double vec[],int size)
{
    double *div;
    int i;
    div = (double *) calloc(size,sizeof(double));
    if (div==NULL)
    {
        printf("Allocation Failure");
        exit(1);
    }
    for (i=1;i<size-1;i++)
        div[i] = (vec[i+1] - vec[i-1])/2;
    return div;
}
```



**סעיף ב' (8 נקודות):**

כתבו פונקציה בעלת המפרט הבא :

```
void change_to_mean(double vec[], int size)
```

הפונקציה מקבלת מערך של ממשיים, ואת גודלו (הניחו קלט לפונקציה תקין). הפונקציה תחליף את ערכי המערך במוצע ערכי המערך.

```
double mean(const double vec[], int size)
{
    int i;
    double sum=0;
    for(i=0;i<size;i++)
        sum = sum + vec[i];
    return sum/size;
}

void change_to_mean(double vec[], int size)
{
    int i;
    double mn = mean(vec,size);
    for (i=0;i<size;i++)
        vec[i] = mn;
    return;
}
```

**סעיף ג' (15 נקודות):**

כתבו פונקציה בעלת המפרט הבא :

```
void smooth(double vec[], int size, double thresh)
```

הפונקציה מקבלת מערך המייצג סדרה של ממשיים, את אורכו, ומספר ממשי כלשהו, thresh. הפונקציה תחלק את סדרת המספרים למקטעים בעלי ערכים דומים, לפי הנגזרת בכל נקודה, בצורה הבאה :

המקטע הראשון מתחיל בערך הראשון בסדרה ומסתיים בנקודה הראשונה שבה יש נגזרת שערכה המוחלט גבוה מ-thresh.

המקטע הבא יתחיל מהנקודה הבאה בסדרה, ויגמר בנקודה הראשונה בהמשך הסדרה, שלה נגזרת שערכה המוחלט גבוה מ-thresh. לפי כלל זה יחולקו גם שאר המקטעים, כאשר המקטע האחרון נגמר בערך האחרון בסדרה.

בכל מקטע כזה **תחליף** הפונקציה את ערכי המערך **בממוצע הערכים שבמקטע**.

לדוגמה:

עבור  $\text{thresh} = 0.3$  ו-vec מהדוגמה הקודמת, יתקבלו המקטעים הבאים:

2	2.3	2.4	2.1	2	5	5.2	5.3	5	5.3	4.5
---	-----	-----	-----	---	---	-----	-----	---	-----	-----

ויוחזר המערך:

2.16 2.16 2.16 2.16 2.16 5.05 5.05 5.05 5.05 5.05 5.05

המקטעים מחולקים כך מכיוון שהנקודה הראשונה במערך שלה נגזרת גבוהה מ-0.3 היא הנקודה החמישית במערך. לפי ההגדרה הנקודה הבאה (השישית) מתחילה את המקטע השני, ומכיוון שאין לאחריה נקודה נוספת עם נגזרת גבוהה מהסף, נגמר המקטע השני בערך האחרון בסדרה.

```
void smooth(double vec[], int size, double thresh)
{
    double *div;
    int st=0,ed=0;
    div = derivative(vec,size);

    while(st<size)
    {
        // find the end of the interval
        while (ed<size-1 && (div[ed] < fabs(thresh)))
            ed++;

        // Change the interval to its mean
        change_to_mean(vec+st,ed-st+1);

        // The next interval starts at the next array cell.
        st = ed+1;

        // Takes care of the end of the new interval (Interval
        length is at least 2 cells)
        ed = st+1;
    }
    free(div);
    return;
}
```

**שאלה מס' 3 (34 נקודות)**

בשאלה זו נשתמש במבנה הבא שמייצג חשבון בנק של לקוח:

```
typedef struct account_t{
    char name[40];
    double amount;
    int date;
    struct account_t *next;
} account;
```

השדה name מייצג את שם הלקוח (ניתן להניח שהשם נכתב תמיד ב-lower case ותמיד שם המשפחה קודם לשם הפרטי), השדה amount מייצג את היתרה בחשבון, והשדה date מייצג את תאריך העדכון האחרון (בעזרת מספר שמונה ספרתי מהצורה yyymmdd, כך למשל התאריך של היום יכתב כ-20070710).

יש להניח שאיבריה של רשימה מקושרת נתונה הוקצו בצורה דינאמית.

**סעיף א' (6 נקודות):**

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int is_newer(account *newer, account *older);
```

הפונקציה מקבלת שני מצביעים למבנים מסוג account (הניחו שאכן זה מה שהיא תקבל). הפונקציה מחזירה 1 במידה ושני החשבונות הם של אותו האדם ותאריך העדכון של new חדש יותר מאשר של old. אחרת תחזיר הפונקציה 0.

```
int is_newer(account *newer, account *older)
{
    if (!strcmp(newer->name, older->name) &&
        (newer->date >= older->date))
        return 1;
    return 0;
}
```

**סעיף ב' (16 נקודות):**

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
account *delete_account(account *head, char name_d[],
                        int date_d, int *flag);
```

הפונקציה מקבלת מצביע לרשימה מקושרת של חשבונות, שם ותאריך, ומוחקת את החשבון הראשון ברשימה המשוך לשם זה ושהתאריך שלו קודם לתאריך המבוקש (כלומר קטן יותר). הפונקציה מחזירה מצביע לראש הרשימה החדש. במידה ולא נמצא איבר מתאים למחיקה, flag יצביע על הערך 0. אחרת flag יצביע על הערך 1.

```
account *delete_account(account *head, char name_d[],int date_d,
int *flag)
{
    account *cur = head, *prev = head;
    account to_check;
    to_check.date = date_d;
    strcpy(to_check.name,name_d);
    // Check if the list is empty,
    if (head==NULL)
    {
        *flag = 0;
        return head;
    }
    // Check if the first item in the list is to be deleted.
    if (is_newer(cur,&to_check))
    {
        cur = head->next;
        free(head);
        *flag = 1;
        return cur;
    }
    // Look for the item to be deleted.
    while(cur->next!=NULL)
    {
        prev = cur;
        cur = cur->next;
        if (is_newer(cur,&to_check))
        {
            prev->next = cur->next;
            free(cur);
            *flag = 1;
            return head;
        }
    }
    // The item was not found.
    *flag = 0;
    return head;
}
```

**סעיף ג' (12 נקודות):**

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
account *squeeze(account *head);
```

הפונקציה מקבלת מצביע לרשימה של מבנים ו"מצמצמת" אותה כך שבעבור כל לקוח יש לכל היותר חשבון אחד. הצמצום מתבצע כך ש-account שהתאריך שלו הוא החדש ביותר (הגבוה ביותר) נשמר, וכל השאר נמחקים.

```
account *squeeze(account *head)
{
    account *ptr = head;
    int flag = 1;
    while (ptr!=NULL)
    {
        while(flag)
            head = delete_account(head, ptr->name, ptr->
date, &flag);
        ptr = ptr->next;
    }
    return head;
}
```

סמסטר אי+בי תשס"ו  
מועד מיוחד, 28/9/2006  
משך הבחינה: שלוש וחצי שעות  
אין להשתמש בחומר עזר

## בחינה בקורס תכנות ב- C

**מרצים:** עדי אבידור, נורית הספל, אמיתי ערמון

**מתרגלים:** אמיר אפשטיין, בן זנדבנק, שחר הרוסי, צור יצחקיאן,  
אולגה סורקין, עופר פסטרנק, ליעם רודיטי, נירה שפיר

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש בטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספריה (זכרו לבצע `#include` מתאים).
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.

**בהצלחה!**

**לשימוש הבודקים:**

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
	ציון

## שאלה מס' 1 (20 נקודות)

כתבו **תוכנית** שקולטת מהמשתמש  $n$  מספרים **ממשיים** :  $a_1, a_2, \dots, a_n$ . התוכנית תדפיס  $n$  מספרים ממשיים  $b_1, b_2, \dots, b_n$ , כך ש-  $b_i$  הוא **המכפלה** של כל המספרים שנקלטו, **פרט למספר  $a_i$** . אורך הקלט,  $n$ , נקלט מהמשתמש לפני שנקלטים המספרים הממשיים, ועל התוכנית להקצות עבורם מערך בגודל זה (לא ניתן להניח ש-  $n$  קטן מקבוע כלשהו).

**הערה:** מלוא הניקוד יינתן לפתרון שמבצע מעבר על מערך הקלט פעם אחת בלבד (כמובן בנוסף לפעם שבה קולטים את המספרים לתוך המערך הזה).

## שאלה מס' 2 (25 נקודות)

### סעיף א' (12 נקודות)

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void order_str(char s[], int ind1, int ind2);
```

הפונקציה מקבלת **מחרוזת** (ניתן להניח שהיא חוקית) ו**שני אינדקסים** (ניתן להניח שהם אי-שליליים ושהם קטנים מאורך המחרוזת, וכן ש- $ind2 > ind1$ ). על הפונקציה להזיז את התו שנמצא במקום  $ind2$  אל המקום  $ind1$ , ולהזיז את כל התווים שבין המקום  $ind1$  למקום  $ind2-1$  **מקום אחד קדימה**. במילים אחרות, על הפונקציה "לדחוף" את התו שנמצא ב- $ind2$  לתחילת הרצף המתחיל ב- $ind1$ , תוך שמירה על הסדר של התווים במקומות  $ind1, \dots, ind2-1$ . שאר המחרוזת נותרת ללא שינוי (ואורך המחרוזת לא משתנה).

**דוגמא:** אם  $s$  היא המחרוזת "I am Here!!!", אז לאחר הפעלת  $order\_str(s,2,7)$ , המחרוזת  $s$  תהיה "I ram Hee!!!".

### סעיף ב' (13 נקודות)

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void change_str(char s[]);
```

הפונקציה מקבלת מחרוזת המורכבת מאותיות גדולות וקטנות, ומסדרת אותה מחדש כך שכל האותיות הגדולות באות לפני כל האותיות הקטנות. הסדר הפנימי של האותיות הגדולות והקטנות **לא ישתנה**. ניתן להניח שהמחרוזת המועברת חוקית, ומכילה אותיות בלבד.

**דוגמאות:**

KuKuRiKu --> KKRKuuiu

MILLIONisALOT --> MILLIONALOTis

**הערה:** שימו לב שבשני הסעיפים יש לשנות את המחרוזת המועברת לפונקציה (ולא להחזיר מחרוזת אחרת במקומה).



### שאלה מס' 3 (30 נקודות)

בשאלה זו נשתמש בהגדרה הבאה :

```
typedef struct _element
{
    double key;
    struct _element *next;
} element;
```

#### סעיף א' (10 נקודות) :

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
element *find_val(element *list, double value);
```

הפונקציה תחזיר מצביע לאיבר **הראשון** ברשימה המקושרת list, שערך שדה ה-key שלו שווה ל-value. במידה ואין ברשימה איבר עם הערך הזה, הפונקציה תחזיר NULL.

#### סעיף ב' (10 נקודות) :

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
element *find_val_last(element *list, double value);
```

הפונקציה תחזיר מצביע לאיבר **האחרון** ברשימה המקושרת list, שערך שדה ה-key שלו שווה ל-value. במידה ואין ברשימה איבר עם הערך הזה, הפונקציה תחזיר NULL.

#### סעיף ג' (10 נקודות) :

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
element *insert(element *list, element *add, double after);
```

הפונקציה מכניסה לרשימה list את הרשימה add, אחרי האיבר **האחרון** ברשימה list שערך שדה ה-key שלו שווה ל-after. אם אין ברשימה איבר עם הערך after, אז הרשימה add תתווסף בתחילת הרשימה list. הפונקציה מחזירה את ראש הרשימה המעודכנת.

**לדוגמא:** אם הרשימה list מכילה (משמאל לימין) את האברים 1 2 3 4 3 5, הרשימה add מכילה את האיברים 6 7 8, והערך של after הוא 3, אז הרשימה שתוחזר תהיה :

1 2 3 4 3 6 7 8 5

**הערה:** הקפידו לטפל במקרי קצה, בפרט במקרים של רשימה ריקה.

**שאלה מס' 4 (25 נקודות)**

**סעיף א' (9 נקודות):**

מה מבצעת הפונקציה הבאה? נמקו את תשובתכם.

```
#include <string.h>
void secret(int n, char *str)
{
    int len;

    if (n < 0) {
        str[0] = '-';
        secret(-n, str+1);
        return;
    }
    if (n < 10) {
        str[0] = '0'+n;
        str[1] = '\\0';
        return;
    }
    secret(n/10, str);
    len = strlen(str);
    str[len] = '0'+n%10;
    str[len+1] = '\\0';
}
```

---

---

---

---

---

---

---

---

**סעיף ב' (8 נקודות):**

מה מבצעת הפונקציה הבאה? נמקו את תשובתכם.

```
void secret(int n)
{
    int i;

    for (i = 2; i < n; i++)
        if (n%i == 0)
            break;
    if (i == n)
        printf("%d\\n", n);
    else {
        printf("%d*", i);
        secret(n/i);
    }
}
```

---

---

---

---

---

---

---

**סעיף ג' (8 נקודות):**  
נתונה הפונקציה הבאה:

```
void secret(int *nums, int k, int *n)
{
    int i=0, j;
    while(i < *n)
    {
        if (nums[i]==k)
        {
            for(j = i; j < *n-1; j++)
                nums[j]=nums[j+1];
            *n=*n-1;
        }
        else
            i++;
    }
}
```

מה מבצעת הפונקציה הזו? נמקו את תשובתכם.

---

---

---

---

---

---

---

# בהצלחה!

## הצעת פתרון למבחן בתכנות למהנדסים – מועד ג', סמסטר א' תשס"ו

### שאלה מס' 1

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int i, n, zeros=0;
    double *a, mult=1;

    printf("How many numbers?\n");
    scanf("%d", &n);
    while(n<2)
    {
        printf("Should be at least 2 - enter again\n");
        scanf("%d\n", &n);
    }

    a = (double *) malloc (n * sizeof(double));
    if (a == NULL)
    {
        printf("Allocation failed\n");
        exit(1);
    }
    printf("Enter the numbers\n");
    for(i=0; i < n; i++)
    {
        scanf("%lf", &a[i]);
        if (a[i]!= 0)
            mult *= a[i];
        else
            zeros++;
    }

    printf("The results are:\n");
    for(i=0; i < n; i++)
    {
        if (zeros>1)
        {
            printf("%d ", 0);
            continue;
        }
        if(zeros==1)
        {
            if(a[i]!=0)
                printf("%d ", 0);
            else
                printf("%g ", mult);
        }
        else
            printf("%g ", mult/a[i]);
    }

    free(a);
    return 0;
}
```

[לא הופחתו נקודות לגבי בדיקת תקינות הקלט]

שאלה מס' 2  
סעיף א':

```
void order_str(char s[], int ind1, int ind2)
{
    int i;
    char temp;

    temp=s[ind2];
    for(i=ind2; i > ind1; i--)
        s[i]=s[i-1];
    s[ind1]=temp;
}
```

סעיף ב':

```
#include<string.h>
void change_str(char s[])
{
    int i, len, place=0;

    len=strlen(s);
    for(i=len-1; i > 0; i--)
        if ((s[i]>'A') && (s[i]<'Z'))
            {
                order_str(s,place,i);
                place++;
            }
}
```

שאלה מס' 3  
סעיף א':

```
#include<stdlib.h>
element *find_val(element *list, double value)
{
    while(list != NULL)
    {
        if (list->key == value)
            return list;
        list=list->next;
    }
    return NULL;
}
```

סעיף ב':

```
#include<stdlib.h>
element *find_val_last(element *list, double value)
{
    element *last=NULL;

    while(list != NULL)
    {
        if (list->key == value)
            last=list;
        list=list->next;
    }
    return last;
}
```

```
#include<stdlib.h>

element *insert(element *list, element *add, double after)
{
    element *place, *last_add;

    if (add==NULL)
        return list;

    last_add=add;
    while (last_add->next!=NULL)
        last_add=last_add->next;

    place=find_val_last(list, after);
    if (place==NULL)
    {
        last_add->next = list;
        return add;
    }

    last_add->next = place->next;
    place->next = add;
    return list;
}
```

#### שאלה מס' 4:

##### סעיף א':

הפונקציה מקבלת מספר שלם ומצביע למחרוזת (שמניחים שכבר הוקצתה באורך מספיק). היא שמה במחרוזת את ספרותיו של המספר הזה (כתווים), תוך שמירת סימן מינוס בתחילת המספר אם היה כזה. למשל עבור המספר 123 יהיה במחרוזת "123" ועבור -123 יהיה במחרוזת "-123". בבסיס הרקורסיה ספרת האחדות מתורגמת לתו המתאים לה, ובכל שלב ברקורסיה הספרה שהייתה מימין לספרה הקודמת במספר מתוספת מימינה במחרוזת.

##### סעיף ב':

הפונקציה מקבלת מספר שלם (שהיא מניחה שהוא גדול מ-1), ומדפיסה את הפירוק שלו לגורמים ראשוניים (עם סימן כפל ביניהם). בכל שלב ברקורסיה מחפשים גורם קטן ככל האפשר של המספר (ולכן בהכרח ראשוני), מדפיסים אותו, וממשיכים עם המספר שנשאר אחרי החלוקה בו. עוצרים כשמה שנשאר הוא מספר ראשוני (מתחלק רק בעצמו).

##### סעיף ג':

הפונקציה מקבלת מערך של מספרים שלמים, ערך למחיקה  $k$ , ומצביע למשתנה המכיל את גודל המערך. הפונקציה מוחקת מהמערך את כל המופעים של  $k$ , "מצופפת" את המערך על חשבון התאים שנמחקו, ומעדכנת בהתאם את המשתנה המכיל את אורך המערך באמצעות המצביע. הלולאה החיצונית עוברת על כל תאי המערך, ובכל פעם שהערך  $k$  מתגלה הלולאה הפנימית מצופפת את תאי המערך על חשבון וזורקת את אורך המערך מעודכן.

מס' מחברת : S מס' ת.ז. : 066162629



סמסטר א' תשס"ז  
מועד : א' 18/2/2006  
משך הבחינה : שלוש וחצי שעות  
אין להשתמש בחומר עזר

## בחינה בקורס תכנות ב- C

מרצים : אמתי ערמון, עופר פסטרנק, ליאור שפירא

מתרגלים : אברהם בן-ארויה, רן גל, בן זנדבנק, אלון לרנר,  
אילונה קיפר, מתי שמרת, נירה שפירר, אלון שקלר

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיזטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספרייה (זכרו לבצע #include מתאים).
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.

בהצלחה!

לשימוש הבודקים:

תעודת זהות	מספר מחברת
שאלה 1	24
שאלה 2	25
שאלה 3	25
שאלה 4	25
ציון	99

**שאלה מס' 1 (25 נקודות)**

**סעיף א' (5 נקודות):**

כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int how_many(int array[], int size, int value);
```

הפונקציה מקבלת מערך של מספרים שלמים, את גודל המערך, size, ומספר שלם נוסף, value (הניחו שזה אכן מה שמועבר אליה). היא מחזירה את כמות ההופעות של המספר value במערך, כלומר כמה תאים במערך שווים ל-value.  
לדוגמא, עבור מערך שערכיו הם 1 2 1 2 1 2 1 21 12 1 והמספר 1 יחזור 3.

```
#include <stdio.h>
int how_many(int array[], int size, int value)
{
    int i=0, counter=0;
    for(i=0; i<size; i++)
    {
        if(array[i]==value)
            counter++;
    }
    return counter;
}
```

5/5

**סעיף ב' (10 נקודות):**

כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int most_popular(int array[], int size);
```

הפונקציה מקבלת מערך של מספרים שלמים ואת גודל המערך (הניחו שזה אכן מה שמועבר אליה). עליה להחזיר את המספר הנפוץ ביותר במערך, כלומר המספר שמופיע בו הכי הרבה פעמים (אם יש כמה מספרים עם כמות הופעות זהה, אז יחזור אחד מהם).  
לדוגמא, עבור מערך שערכיו הם: 12 23 34 45 1 2 3 4 5 4 3 2 1 4 יחזור 4 (כי הוא מופיע במערך שלוש פעמים, וזה יותר מכל מספר אחר)

```
int most_popular(int array[], int size)
{
    int i=0, most_popular=0, index=0;
    for(i=0; i<size; i++)
    {
        if(how_many(array, size, array[i]) > most_popular)
        {
            most_popular = how_many(array, size, array[i]);
            index = i;
        }
    }
    return array[index];
}
```

וש לזכור  
קצת קשה  
אולי לא  
9/10



**סעיף ג' (10 נקודות):**

כיתבו תכנית שקולטת מספר שלם חיובי  $n$ , שמייצג את אורך הקלט, ואז קולטת  $n$  מספרים שלמים כלשהם. לאחר מכן, התוכנית קולטת שני מספרים שלמים נוספים,  $start$  ו- $end$ , שערכיהם הם בין 1 ל- $n$ , כך ש- $end$  גדול-שווה ל- $start$ . התוכנית תדפיס את המספר הנפוץ ביותר בקטע של רשימת  $n$  המספרים הנייל, שמתחיל במספר ה- $start$  שנקלט ומסתיים במספר ה- $end$  שנקלט (ראו דוגמא למטה). המספרים שאינם בקטע המבוקש לא משפיעים על הפלט. אם כמה מספרים נפוצים באותה מידה בקטע המבוקש אז יודפס רק אחד מהם. לדוגמא, עבור הקלט (משמאל לימין):

10  
1 2 3 2 3 2 1 1 1 1  
3 5

הפלט יהיה 3 (כי הוא הנפוץ ביותר בקטע שמתחיל במספר השלישי ומסתיים במספר החמישי)

**הערות:** - לא ניתן להניח שכמות המספרים שיקלטו יותר קטנה ממספר קבוע כלשהו.  
- ניתן להניח שהקלט תקין, וגם שהמשתמש יודע איזה קלט להכניס (אין צורך להדפיס הודעות שיסבירו לו מה הקלט הדרוש).

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i=0, n=0, score=0, end=0, temp=0;
    int* nums=NULL;
    printf("please enter an integer\n");
    scanf("%i", &n);
    nums=(int*) malloc(sizeof(int)*n);
    if (nums == NULL)
    {
        printf("out of memory\n");
        exit(1);
    }
    printf("please enter %d integers\n", n);
    scanf("%i", &temp);
    for(i=0; i<size; i++)
    {
        nums[i]=temp;
        scanf("%i", &temp);
    }
    nums[i]=temp;
    printf("enter 2 integers\n");
    scanf("%i", &start);
    scanf("%i", &end);
    start--;
    start
    nums+=start;
    temp = most_popular(nums, end-start);
}
```

מס' ת.ז.?  
num?

10/10

: 6 8x0 2017

```
printf("id %n", temp);
```

```
free(temp);
```

```
free(nums - start); /* 1378 'id' nums-f r'51h */
```

```
return 0;
```

```
}
```

שאלה מס' 2 (25 נקודות)

סעיף א' (12 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void kth_word (char words[], int k, char kth[]);
```

הניחו ש-words היא מחרוזת המכילה מילים באנגלית, כשבין כל שתי מילים סמוכות יש תו-רווח אחד, k הוא מספר שלם חיובי, ו-kth היא מחרוזת נוספת. בסיום הפונקציה, המחרוזת ב-kth צריכה להיות המילה ה-k-ית מבין המילים שהועברו ב-words (למשל אם k=1 אז תועתק ל-kth המילה הראשונה). אם יש ב-words פחות מ-k מילים, אז הפונקציה תשים ב-kth מחרוזת ריקה (כלומר תו '\0' בלבד). הניחו שב-kth יש מספיק מקום לצורך ההעתקה.

```
#include <stdio.h>
#include <stdlib.h>
char* next_word(char* words)
{
    char* next_word = words;
    while(*next_word != '\0')
    {
        if(*next_word == '/')
            return NULL;
        next_word++;
    }
    next_word++;
    return next_word;
}

void kth_word(char words[], int k, char kth[])
{
    int i;
    char* ptr = words;
    for(i=1; i <= k; i++)
    {
        ptr = next_word(ptr);
        if(ptr == NULL)
        {
            *kth = '\0';
            return;
        }
    }
}
```

\* פונקציה שכתבתי  
אם k > מספר המילים  
אז תחזיר NULL

12  
12

\* אם k > מספר המילים  
אז תחזיר NULL

אם k > מספר המילים  
אז תחזיר NULL

(2) pen

```
while(*ptr != '\0' || *ptr != '\n')
```

```
{
```

```
    *kch = *ptr;
```

```
    ✓ ptr++;
```

```
    kch++;
```

```
}
```

```
*kch = '\0';
```

```
return;
```

```
}
```

**סעיף ב' (13 נקודות):**

כיתבו תוכנית הקולטת משפט באנגלית, באורך 300 תווים לכל היותר (בסיומו מוקלד Enter). לאחר מכן התוכנית קולטת מילה באנגלית, באורך 30 אותיות לכל היותר (שבסיומה מוקלד Enter). על התוכנית להדפיס את מספר הפעמים שהמילה הזו מופיעה כמילה במשפט שנקלט, בהנחה שהמשפט מכיל רק מילים באנגלית ובין כל שתי מילים סמוכות יש תו-רווח.  
**לדוגמא:**

- אם המשפט הוא "I am what I am" והמילה היא "am" אז הפונקציה תחזיר 2
- אם המשפט הוא "I am what I am" והמילה היא "at" אז הפונקציה תחזיר 0

**הערות:**

- ניתן להניח שאורך כל מילה במשפט הוא 30 אותיות לכל היותר.
- ניתן להניח שהקלט תקין, וגם שהמשתמש יודע איזה קלט להכניס (אין צורך להדפיס הודעות שיסבירו לו מה הקלט הדרוש).

```
#include <string.h>
int main()
{
    char str1[301];
    char str2[31];
    char kth[31];
    int i=0, counter=0;
    char temp;
    printf("please enter first string\n");
    temp = getch();
    for(i=0; i<301; i++)
    {
        if(temp == '\n') break;
        str1[i] = temp;
        temp = getch();
    }
    str1[i] = '\0';
    printf("enter second string\n");
    temp = getch();
    for(i=0; i<31; i++)
    {
        if(temp == '\n') break;
        str2[i] = temp;
        temp = getch();
    }
    str2[i] = '\0';
    kth = word(str1, 1, kth);
    for(i=2; *kth != '\0'; i++)
    {
        if(strcmp(str2, kth) == 0)
            counter++;
        kth = word(str1, i, kth);
    }
}
```

\* (קוד) מילים 300

\* (קוד) מילות שני

13  
73

מס' מחברת : 5

(22) 2013

```
printf("%d\n", counter);  
return 0;  
}
```

שאלה מס' 3 (25 נקודות)

סעיף א' (5 נקודות):

הגדירו מבנה (struct) בשם Animal, שיכיל את המידע הבא: סוג-החיה (עד 30 תווים), כמות החיות המשוערת מסוג זה בארץ (מס' שלם), והאם החיה בסכנת-הכחדה (1 או 0). כמו כן, על המבנה לאפשר יצירת רשימה מקושרת של חיות.

```
typedef struct animal_d
{
    char type[31];
    double amount;
    int danger;
    struct animal_d* next;
} Animal;
```

5/5

סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Animal *in_danger(Animal *head);
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של חיות. עליה להחזיר מצביע לתחילת רשימה מקושרת חדשה, שתכלול רק את החיות בסכנת-הכחדה מבין החיות ברשימה המקורית (לא משנה באיזה סדר). על הרשימה המקורית להישאר ללא שינוי.

```
#include <stdio.h>
#include <stdlib.h>

Animal* create_copy(Animal* to_copy)
{
    int i=0;
    Animal* new = NULL;
    new = (Animal*) malloc(sizeof(Animal));
    if(new == NULL)
    {
        printf("out of memory");
        exit(1);
    }
    for(i=0; to_copy->type[i]!='\0'; i++)
        new->type[i] = to_copy->type[i];
    new->type[i] = '\0';
    new->amount = to_copy->amount;
    new->danger = to_copy->danger;
    new->next = NULL;
}
```

10/10

(אם לא יצא Animal \* אז יצא NULL)

} return new; Page 40

אם לא יצא Animal \* אז יצא NULL

Animal\* In-danger(Animal\* head)

{

Animal\* new-head = NULL;

Animal\* new-list;

int counter = 0;

~~head->danger = 1;~~

/\* the pointer to the next node is NULL \*/

while(head != NULL)

{

if (head->danger == 1)

{

if (counter == 0)

{

new-head = create-copy(head);

new-list = new-head;

counter++;

}

else { new-list->next = create-copy(head);

new-list = new-list->next;

}

}

head = head->next;

}

return new-head;

}



**סעיף ג' (10 נקודות):**

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Animal *sorted_add(Animal *head, Animal *to_add);
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של חיות, שממויינת לפי הכמות בארץ (בסדר עולה), ומצביע למבנה של חיה שיש להוסיף לרשימה. עליה להוסיף את המבנה הזה לרשימה תוך שמירה על סדר המיון. על הפונקציה להחזיר מצביע לתחילת הרשימה המעודכנת.

```
Animal* sorted_add(Animal* head, Animal* to_add)
{
    Animal* prev = NULL;
    Animal* curr = head;
    while((to_add->amount > curr->amount) && (curr != NULL))
    {
        prev = curr;
        curr = curr->next;
    }
    if (prev == NULL)
    {
        to_add->next = curr;
        return to_add;
    }
    if (curr == NULL)
    {
        prev->next = to_add;
        to_add->next = NULL;
        return head;
    }
    prev->next = to_add;
    to_add->next = curr;
    return head;
}
```

if (to\_add->amount > curr->amount) && (curr != NULL)

if (curr == NULL)
 {
 prev->next = to\_add;
 to\_add->next = NULL;
 return head;
 }

return head;

10/10

מס' מחברת: S מס' ת.ז.: 066162629

### שאלה מס' 4 (25 נקודות)

בסעיפים א' ו-ב' אנו משתמשים בתגדרה הבאה:

```
typedef struct data_t {
    int info;
    struct data_t *next;
} Data;
```

#### סעיף א' (7 נקודות):

נתונה הפונקציה הבאה, שיש בה בעיה מסויימת:

```
#include <stdio.h>
#include <stdlib.h>
void secret(Data *p)
{
    if (p!=NULL)
        secret(p->next);
    printf("%d ", (*p).info);
}
```

7/7

- מה הבעיה שיש בפונקציה הזאת? (הקיפו אחת מבין האפשרויות הבאות)
  - הביטוי (\*p).info איננו ביטוי חוקי בשפת C
  - לא ניתן לקרוא לפונקציה secret עם הערך p->next
  - הפונקציה צפויה לגשת לערך שנמצא בכתובת NULL
  - חסרה בסוף הפונקציה פקודת return להחזרת ערך

- מה לדעתכם הפונקציה הזאת אמורה לבצע? (כלומר בהנחה שהבעיה הנייל תתוקן)

הפונקציה Secret היא כונק' קורסיבית שלאווה לעברים אז ערכים ה- info של יציאת data לקוחים מהיסוד לעיכוליה

#### סעיף ב' (6 נקודות):

נתונה הפונקציה הבאה:

```
#include <stdlib.h>
int secret(Data *head)
{
    if ((head==NULL) || (head->next==NULL))
        return 0;
    if (head->info == head->next->info)
        return 1+secret(head->next);
    else
        return secret(head->next);
}
```

6/6

מה משמעות הערך שהפונקציה הזו מחזירה?

הפונקציה Secret מקבלת גברים לראש ויש לה לקשרת וסוברת

אם כמות העלגים קבן בטאים סמוכה עס'יה גופל' אונו ערק פחלו.

הערך 0 - Secret מחזירה הוא כמה בטאים יש אונו פחלו

אם העס'יה יקדי או גולף יש אדם גולפו 0.

סעיף ג' (6 נקודות):  
נתונה הפונקציה הבאה:

```
#include <stdio.h>
void secret()
{
    int a[] = {1,2,3,4,5};
    int b, x = 0;
    for (b=5 ; b>0; b--)
    {
        x += *a;
        a++;
    }
    printf("%d\n",x);
}
```

6/6

- מה הפונקציה הזאת נועדה לחשב לדעתכם? אם סכום האיברים ק אולי
- מה הבעיה בפונקציה הזאת? (הקיפו אחת מבין האפשרויות הבאות)
  - א. לא ניתן לבצע את הפקודה `x += *a;`
  - ב. לא ניתן לבצע את הפקודה `a++;`
  - ג. יש טעות בקידום משתנה הלולאה (המשתנה b)
  - ד. לא ניתן להגדיר כך את המערך a

סעיף ד' (6 נקודות):

נתונה התכנית הבאה, שיש בה בעיה מסויימת:

```
#include <stdio.h>
int main()
{
    int a[10];
    int n, i;

    scanf("%d",&n);
    for (i=0; i<n; i++)
        scanf("%d",&a[i]);

    for (i=n-1; i>=0; i--)
        printf("%d ",a[i]);

    return 0;
}
```

6/6

- מה הבעיה בתוכנית הזאת? (הקיפו אחת מבין האפשרויות הבאות)
  - א. יש & מיותר בשורה `scanf("%d",&a[i]);`
  - ב. עלולים לגשת אל `a[-1]`
  - ג. יש & מיותר בשורה `scanf("%d",&n);`
  - ד. עלולים לגשת אל `a[10]`

• מה לדעתכם התכנית הזאת הייתה מיועדת לעשות? (כלומר אם לא הייתה בה הבעיה הנ"ל)

התוכנית אמורה לדפוס את המספרים הנכנסים בסדר הפוך

אולי אולי לא, אבל זה תלוי בהגדרת המערך a

אם המערך a הוא מסוג char אז זה ידפיס את האותיות בסדר הפוך

מס' מחברת : 67 מס' ת.ז. : 60413978

67

סמסטר ב' תשס"ו  
מועד : ב' 27/6/2006  
משך הבחינה : שלוש וחצי שעות  
אין להשתמש בחומר עזר

## מבחן בתכנות

מרצה : עדי אבידור

מתרגלים : בן זנדבנק, ליעם רודיטי

### הנחיות כלליות

- נא לכתוב בכתב קריא וברור.
- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה.
- במחברת הבחינה יש להשתמש כטיוטה בלבד.
- יש למלא מספר תעודת הזהות ומספר מחברת בכל עמוד של השאלון.
- בכל שאלה הינכם רשאים להשתמש בתוצאות הסעיפים הקודמים גם אם לא פתרם אותם.

311	א"ה
	1
24	2
	3
(98)	סה"כ

# בהצלחה!

מס' מחברת : 67 מס' ת.ז. : 060413978

שאלה מס' 1 (45 נקודות)

א. (10 נקודות)  
כתוב פונקציה `int is_prime(int n)` המקבלת מספר טבעי  $n$  ומחזירה 1 אם המספר הוא ראשוני ואפס אחרת (מספר טבעי יקרא ראשוני אם הוא מתחלק בעצמו ובאחד בלבד. כמו כן אחד אינו מוגדר כראשוני).  
אם  $n$  קטן מ 1 הפונקציה תחזיר -1.

פתרון לסעיף א.

```
int is_prime(int n)
{
    int i;
    if (n < 1) return -1;
    if (n == 1) return 0;
    for (i = 2; i < n; i++)
        if (n % i == 0) return 0;
    return 1;
}
```




ב. (10 נקודות)

כתוב פונקציה `int multiplicity(int n, int p)` המקבלת שני מספרים טבעיים  $n$  ו  $p$  ומחזירה את הריבוי של  $p$  ב  $n$ . כלומר, הפונקציה מחזירה את החזקה  $i$  הכי גבוהה של  $p$  כך שהמספר  $p^i$  מחלק את  $n$ .

לדוגמא: `multiplicity(162, 3)` תחזיר 4, כי  $3^4$  מחלק את 162 אבל  $3^5$  לא מחלק את 162.

אם  $p$  לא מחלק את  $n$  הפונקציה תחזיר 0. אם  $n$  קטן מ 1 או ש  $p$  קטן מ 2 הפונקציה תחזיר -1.

```
פתרון לסעיף ב.
int multiplicity(int n, int p)
{
    int i = 0;
    if (n < 1 || p < 2) return -1;
    while (n % p == 0)
    {
        n = n / p;
        i++;
    }
    return i;
}
```



ג. (15 נקודות)

כתוב פונקציה (int n) \*prime\_factors(int n) המקבלת מספר טבעי n ומחזירה מצביע לאבר ראשון במערך המכיל במקום הראשון את מספר הגורמים הראשוניים של n וביתרת המקומות את הגורמים הראשוניים.

לדוגמא: עבור n=525 הפונקציה תחזיר מצביע למערך שארבעת המקומות הראשונים שלו יהיו 3 3 5 7, כי ל 525 יש שלושה גורמים ראשוניים והם 3 ו 5 ו 7. עבור 11 למשל התא הראשון במערך יכיל 1 והשני את 11.

$$\frac{n}{2} + 1 = 11$$

ניתן להניח בפתרון כי מספר הגורמים הראשוניים של n הוא לכל היותר  $\frac{n}{2}$ .

אם n קטן מ 2 הפונקציה תחזיר NULL.

פתרון לסעיף ג.

```
int *prime_factors(int n)
{
    int i=2, *arr, count=0;

    if (n < 2) return NULL;

    arr = (int *) malloc(sizeof(int) * (n/2 + 1));
    if (!arr)
    {
        printf("memory allocation error!\n");
        exit(1);
    }

    while (n > 1)
    {
        if (is_prime(i) == 1)
        {
            while (n % i == 0) n = n / i;
            count++;
            arr[count] = i;
        }
        i++;
    }

    arr[0] = count;

    return arr;
}
```

יש כנול אצול  
 > סכנו אר להקל  
 #include <stdlib.h>

הפונקציה מקבלת מספר טבעי n ומחזירה מצביע לאבר ראשון במערך המכיל במקום הראשון את מספר הגורמים הראשוניים של n וביתרת המקומות את הגורמים הראשוניים.

ד. (10 נקודות)

כתוב תוכנית הקולטת מספר טבעי  $n$  ומדפיסה את הפירוק שלו לגורמים ראשוניים, כולל ריבוי.

לדוגמא: עבור המספר 19992 התוכנית תדפיס  $19992 = 2^4 \cdot 3^3 \cdot 7 \cdot 11$  (כאשר הסימן  $^$  מסמן חזקה). עבור הקלט 11 התוכנית תדפיס  $11 = 11$ .

במידה ו  $n$  קטן מ 2 התוכנית תדפיס הודעת שגיאה ותצא.

```

# include <stdio.h>
# include <stdlib.h>

// אני מטה את כל הפרוטוקול של הסוגים
// ...

void main()
{
    int n;
    int *arr;
    printf("Enter an integer bigger than 1");
    scanf("%d", &n);
    if (n < 2)
    {
        printf("Invalid input!\n");
        return;
    }
    arr = prime_factors(n);
    printf("nd = ", n);
    for (i=1; i<=arr[0]; i++)
    {
        printf("%d^%d ", arr[i], multiplicity(n, arr[i]));
        if (i != arr[0]) printf(" * ");
    }
    printf("\n");

    free(arr);
}
    
```

פתרון לסעיף ד.

↓  
כל  
אל

המיון הוא 1  
אלה עובדים

- 1/2



שאלה מס' 2 (25 נקודות)

א. (10 נקודות)  
כתוב פונקציה `char most_common(char *str)` המקבלת מחרוזת `str`  
ומחזירה את התו השכיח ביותר במחרוזת.

לדוגמא: עבור המחרוזת "abcabaabcccycc" הפונקציה תחזיר את התו 'c'.

אם יש שנים (או יותר) תווים "הכי שכיחים" אזי הפונקציה תחזיר את התו בעל ערך  
Ascii הקטן ביותר מביניהם. אם המחרוזת ריקה או שערכה NULL הפונקציה תחזיר  
'\0'.

```
פתרון לסעיף א.  
char most_common(char *str)  
{  
    int i=0, index, max=0;  
    char arr[256], ans;  
    if (!str || *str == '\0') return '\0';  
    for (i=0; i<256; i++) arr[i]=0;  
    ans = '\0';  
    while (str[i] != '\0')  
    {  
        index = str[i];  
        arr[index]++;  
        if (arr[index] > max) max = arr[index];  
        if (max == arr[index] && str[i] < ans)  
            ans = str[i];  
        i++;  
    }  
    return ans;  
}
```

התשובה: אנו צריכים למצוא את התו השכיח ביותר במחרוזת. נשתמש במערך `arr` שגודלו 256, ונמלא אותו ב-0. נחזיר את התו בעל הערך הקטן ביותר שיש בו מספר של 1 או יותר.

ב. (15 נקודות)  
כתוב תוכנית הקוראת את הקלט מילה אחר מילה, ומדפיסה כל מילה ללא התו הכי שכיח בה. (אם במילה יש שניים (או יותר) תווים "הכי שכיחים" אזי התוכנית תדפיס את המילה ללא התו בעל ערך Ascii הקטן ביותר מביניהם.)

לדוגמא: עבור הקלט `abcd badaa dadzda` התוכנית תדפיס `bcd bd aza`

```
#include <stdio.h>
#define MAX_LETTERS 50 //
void main()
{
    char word[MAX_LETTERS], c, mc;
    int i=0;

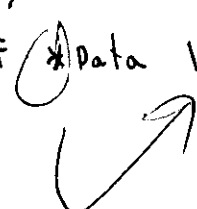
    while ((c=getchar())!='\n')
    {
        if (c==' ')
        {
            word[i]='\0';
            mc = most-common(word);
            for (i=0; word[i]!='\0'; i++)
                if (word[i]!=mc)
                    putchar(word[i]);
            putchar(' ');
            i=0;
        }
        else
        {
            word[i]=c;
            i++;
        }
    }
    printf("\n");
}
```

פתרון לסעיף ב.  
ארגון 'מס' מס' א  
לא 'ג' 'ו'  
את המילה  
האחרונה  
| - |

שאלה מס' 3 (30 נקודות)

א. (5 נקודות)  
הגדר טיפוס נתונים לייצוג אבר ברשימה מקושרת חד-כיוונית המכילה מספרים טבעיים.  
כמו כן, הגדר טיפוס נתונים בשם list לייצוג הרשימה.

```
פתרון לסעיף א.  
typedef struct _Data {  
    int num;  
    struct _Data *next;  
} Data;  
typedef (*Data) list;
```



- 1

ב. (10 נקודות)

כתוב פונקציה `list split(list lst)` המפצלת רשימה מקושרת `lst` לשתי רשימות. לאחר קריאה לפונקציה הרשימה שתוחזר תכיל את האברים במיקומים הזוגיים (כלומר את האבר השני הרביעי וכו') והרשימה `lst` תצביע לרשימה המכילה את האברים במקומות האי-זוגיים ברשימה המקורית (כלומר, האבר הראשון השלישי וכו').

פתרון לסעיף ב.

```
list split(list lst)
{
    list list even, tmp tmp;

    if (lst == NULL) return NULL;
    if (lst->next == NULL) return NULL;
    even = lst->next;
    tmp = even;
    while (tmp->next)
    {
        lst->next = tmp->next;
        lst = lst->next;
        tmp->next = lst->next;
        if (tmp->next) tmp = tmp->next;
    }
    lst->next = NULL;
    return even;
}
```

עיקר כוונת השאלה  
לשכנע את המורה  
#include <stdlib.h>



ג. (10 נקודות)

כתוב פונקציה `list merge(list lst1, list lst2)` המקבלת שתי רשימות מקושרות ממוינות (בסדר עולה) ותחזיר רשימה אחת המכילה את כל אברי `lst1` וכל אברי `lst2` בסדר ממוין. הפונקציה רשאית לשנות את אברי הרשימות `lst1` ו `lst2`. לא בהכרח בעלות אותו האורך, בפרט עליך לטפל במקרה שאחת מהן או שתיהן ריקות).

```

פתרון לסעיף ג.
list merge (list lst1, list lst2)
{
    list L, tmp;

    if (!lst1) return lst2;
    if (!lst2) return lst1;

    if (lst1->num > lst2->num) { L = lst2; lst2 = lst2->next; }
    else { L = lst1; lst1 = lst1->next; }

    tmp = L;
    while (lst1 && lst2)
    {
        if (lst1->num > lst2->num)
        {
            tmp->next = lst2;
            lst2 = lst2->next;
            tmp = tmp->next;
        }
        else
        {
            tmp->next = lst1;
            lst1 = lst1->next;
            tmp = tmp->next;
        }
    }

    while (lst1) { tmp->next = lst1; lst1 = lst1->next; tmp = tmp->next; }
    while (lst2) { tmp->next = lst2; lst2 = lst2->next; tmp = tmp->next; }
    tmp->next = null;

    return L;
}

```



מסי מחברת : 116 מס' ת.ז. : 066192030

סמסטר א' תשס"ו  
מועד: א' 26/2/2006  
משך הבחינה: שלוש וחצי שעות  
אין להשתמש בחומר עזר



## בחינה בקורס תכנות ב-C

מרצים: עדי אבידור, נורית הספל, אמיתי ערמון

מתרגלים: אמיר אפשטיין, שחר הרוסי, צור יצחקיאן, אולגה סורקין,  
עופר פסטרנק, ליעם רודיטי, נירה שפיר

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטייטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספריה (זכרו לבצע #include מתאים).
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.

בהצלחה!

לשימוש הבדקים:

תעודת זהות	מספר מחברת
<del>066192030</del> 066192030	<del>116</del> 116
	שאלה 1 23.5
	שאלה 2 25
	שאלה 3 23.5
	שאלה 4 24
	ציון 96

שאלה מס' 1 (25 נקודות)

סעיף א' (5 נקודות):

כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int is_in_array(int array[], int size, int value);
```

הפונקציה מקבלת מערך של מספרים שלמים, את גודל המערך (size), ומספר שלם נוסף (value). אם המספר value נמצא במערך או על הפונקציה להחזיר 1, ואחרת עליה להחזיר 0.

סעיף ב' (10 נקודות):

כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int different_nums(int nums[], int len);
```

הפונקציה מקבלת מערך של מספרים שלמים ואת גודל המערך. עליה להחזיר כמה מספרים שונים נמצאים במערך (ראו דוגמא להלן). הפונקציה לא תשנה את המערך שמועבר אליה.

לדוגמא, עבור מערך שערכיו הם: 1 1 7 9 12 7 56 1 587 7 יוחזר 6 (כי המספרים השונים הם 1, 7, 9, 12, 56, 587).

סעיף ג' (10 נקודות):

כיתבו תכנית שקולטת מספר שלם חיובי n, שמייצג את אורך הקלט, ולאחר מכן קולטת n מספרים שלמים. התוכנית תדפיס כמה מספרים שונים יש בין n המספרים האלה (לא כולל המספר הראשון שנקלט, n).

לדוגמא, אם הקלט היה (משמאל לימין): 1 1 7 9 12 7 56 1 587 10 אז הפלט יהיה 6 (כי המספרים השונים הם 1, 7, 9, 12, 56, 587).

הערות: ניתן להניח שחקלט תקין. לא ניתן להניח שהמספר n יותר קטן ממספר קבוע כלשהו.

```
*include <stdio.h>
int is_in_array(int array[], int size, int value)
{
    int i;
    for (i=0; i<size; i++)
    {
        if (array[i]==value)
            return 1;
    }
    return 0;
}
```

(K)  
✓  
S



```

#include <stdio.h>
#include <stdlib.h>

int different_nums(int nums[], int len)
{
    int *a, i, j, count=0, flag=1;
    a = (int*) malloc (len*sizeof(int));
    for (i=0; i<len; i++)
    {
        for (j=0; j<len; j++)
        {
            if (a[j]==nums[i])
                flag=0;
                break;
        }
        if (flag==1)
        {
            a[count]=nums[i];
            count++;
        }
        flag=1;
    }
    free(a);
    return count;
}
    
```

כך נקרא  
\*הקובץ  
to  
if (a=NULL)  
print("allocation failed");  
exit(1);

(D)

~~100~~  
הנסיגה  
הנסיגה  
הנסיגה  
הנסיגה  
-1.5  
7  
13.5

(2)

```

* include <stdio.h>
* include <stdlib.h>

int main()
{
    int n, i, count=0, *arr;
    printf("write the amount of nums n");
    scanf("%d", &n);
    arr = (int*) malloc (n*sizeof(int));
    if (arr == NULL)
    {
        printf("allocation fail\n");
        return (-1);
    }
    printf("write the nums n");
    for (i=0; i<n; i++)
        scanf("%d", &arr[i]);
    count = different_nums(arr, n);
    printf("the amount of different nums is: %d\n", count);
    free(arr);
    return 0;
}

```

perce.  
 2 3 2 1 2 2  
 (2) find

(2)  
 ✓  
 10

שאלה מס' 2 (25 נקודות)

סעיף א' (15 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int skip_strstr (char *str1, char *str2, int skip);
```

הפונקציה תבדוק אם המחרוזת str2 מופיעה בדילוגים של skip במחרוזת str1 (ראו דוגמאות להלן). אם כן, הפונקציה תחזיר את האינדקס של המקום הראשון ב-str1 שבו החלה הופעה כזו של str2, ואחרת היא תחזיר -1.

דוגמאות:

- 1) המחרוזת abc מופיעה בדילוגי 2 במחרוזת cdbadbfea החל ממקום 3.
- 2) המחרוזת abcd מופיעה בדילוגי 1 במחרוזת abcd החל ממקום 0.
- 3) המחרוזת abcd לא מופיעה בדילוגי 2 במחרוזת abcd.

הערה: אם אחת המחרוזות (או שתיהן) באורך 0, אז יש להחזיר -1.

סעיף ב' (10 נקודות):

כתבו תוכנית שקולטת מהמשתמש שתי מחרוזות str1, str2 (שתיהן באורך 80 לכל היותר), ומספר שלם חיובי max\_skip, שמייצג דילוג מקסימלי. התוכנית תדפיס את כל הדילוגים בין 1 ל-max\_skip שבהם str2 מופיעה ב-str1, ועבור כל דילוג כזה היא תדפיס את אינדקס המקום הראשון שבו הייתה הופעה כזו.

הערה: אתם יכולים להניח כי הקלט שהמשתמש מכניס הוא תקין, ושהמחרוזות מופרדות ע"י רידת-שורה ולא כוללות בתוכן רווחים כלשהם.

```
include <stdio.h>
include <string.h>
int skip_strstr(char *str1, char *str2, int skip)
{
    int s1, s2, i, j, x=0;
    s1 = strlen(str1);
    s2 = strlen(str2);
    if (s2 > s1)
        return -1;
    for (i=0; i<s1; i++)
    {
        if (str[i] == str2[0])
        {
            for (j=1; j<s2; j++)
            {
                if (str[i+j] != str2[j])
                    break;
                if (j == s2-1)
                    return i;
            }
            i += skip;
        }
    }
    return -1;
}
```

(E)

Handwritten notes in a circle.

Handwritten notes and arrows at the bottom left.

④ פירוט (אנשי)

```
it ((skip * x + i) % 5) || (str2[j] != str1[i + skip * x])
{
    x = 0;
    break;
}
x++;
}
if (j == s2)
    return i;
}
return -1;
}
```



```

#include <stdio.h>
#include <string.h>
#define n 81

int main()
{
    int max_skip, keep, i;
    char str1[n], str2[n];
    printf("write the first string\n");
    scanf("%s", str1);
    printf("write the second string\n");
    scanf("%s", str2);
    printf("write your max skip\n");
    scanf("%d", &max_skip);

    for (i=1; i<=max_skip; i++)
    {
        keep = skip_strstr(str1, str2, i);
        if (keep != -1)
        {
            printf("index: %d for skip: %d\n", keep, i);
        }
    }

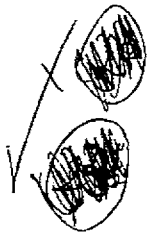
    return 0;
}

```

2

10

הערות  
מיון  
5/1/2017



שאלה מס' 3 (25 נקודות)

סעיף א' (5 נקודות):

הגדירו מבנה (structure) בשם Car, שייצג מכונית. המבנה יכלול את מספר הרישוי, את סוג הרכב (ניתן להניח שזה דורש לכל היותר 30 אותיות), ואת שנת היצור. כמו-כן, המבנה יאפשר יצירת רשימה-מקושרת של מבני Car.

סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Car *add_car(Car *cars_list, Car *to_add);
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מבני Car, ומצביע למבנה Car נוסף (to\_add). הפונקציה יוצרת העתק של המבנה ש-to\_add מצביע עליו, ומוסיפה אותו לתחילת הרשימה ש-cars\_list מצביע עליה. הפונקציה מחזירה מצביע לראש הרשימה החדש (שאר הרשימה לא משתנה).

סעיף ג' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Car *ready_cars(Car *cars, Car *working);
```

הפונקציה מקבלת שני מצביעים לתחילת רשימות מקושרות של מבני Car. היא מחזירה מצביע לראש רשימה מקושרת חדשה, שכוללת רק את המכוניות שנמצאות גם ברשימה ש-cars מצביע עליה וגם ברשימה ש-working מצביע עליה (מכונית שנמצאת רק באחת מהרשימות לא תיכלל ברשימה החדשה). אין לשנות את הרשימות שהמצביעים אליהן מועברים לפונקציה.

הערה: ניתן להניח שהשוואת מס' הרישוי של מכוניות מספיקה כדי לקבוע שהן זהות.

5

```
typedef struct car
{
    int id;
    char type[31];
    int year;
    struct car *next;
} Car;
```

(5)

מס' מחברת: 116 מס' ת.ד.: 066192033

```
#include <stdio.h>
#include <stdlib.h>
```

```
car *add_car(car *cars_list, car *to_add)
```

```
{
```

```
    car *a;
    int i=0;
```

```
    a = (car *) malloc(sizeof(car));
```

```
    if (a == NULL)
```

```
    {
```

```
        printf("allocation fail\n");
```

```
        exit(1);
```

```
    }
```

```
    a->id = to_add->id;
```

```
    while (to_add->(type+i) != '\0')
```

```
    {
```

```
        a->(type+i) = to_add->(type+i);
```

```
        i++;
```

```
    }
```

```
    a->(type+i) = '\0';
```

```
    a->year = to_add->year;
```

```
    a->next = cars_list;
```

```
    return a;
```

```
}
```

2

8 1/2

מס' 3  
a -> type[i]

110

\*a -> type+i

130

מסוימת: 1/6 מסות: 0669002

```
*include <stdlib.h>
*include <string.h>
*include <stdio.h>
```

ד  
→  
הוספת  
רכב  
לרשימה  
הרכבים  
הקיימים  
במערכת

10

```
car *ready_cars (car *cars, car *working)
```

```
int flag=0;
```

```
car *head=NULL, *ac=cars, *aw=working, *first=working;
```

```
while (ac!=NULL)
```

```
while (aw!=NULL)
```

```
if ((aw->id==ac->id) && (strcmp(aw->type,ac->type)==0) && (aw->year==ac->year))
```

```
{
    flag=1;
    break;
}
```

```
aw=aw->next
```

```
if (flag==1)
    head=add_cars(head,ac);
```

```
flag=0;
aw=first;
ac=ac->next;
```

```
return head;
```

הוספת  
רכב  
לרשימה  
הרכבים  
הקיימים  
במערכת  
②



שאלה מס' 4 (25 נקודות)

24

סעיף א' (7 נקודות):  
לפניכם התוכנית הבאה:

✓  
7

```

#include<stdio.h>
#include<string.h>

#define SIZE 30

int main()
{
    char    str[SIZE], new_str[SIZE];

    int     i, len;

    printf("Please enter 20 letters\n");
    scanf("%s", str);
    len=strlen(str);
    for(i=0; i<len; i++)
        new_str[i]=str[len-1-i];
    printf("The result is: %s\n", new_str);

    return 0;
}

```

התוכנית הזו פועלת באופן לא תקין, גם כשהמשתמש מכניס קלט תקין כפי שדרוש. מה לדעתכם התוכנית אמורה לעשות? מהי הבעיה בתוכנית הזו וכיצד ניתן לתקן אותה? רישמו את השורה/ות שיש לשנות/להוסיף (במקרה של תוספת ציינו איפה יש להוסיף אותה).

היא לא תיקן את שם החדש (str) והוא לא  
אינה הוספה - new\_str (וליה "ab" str="ka" new\_str)

באם לא הוסף המיקום החדש של האות 'א' וכל האותיות שאחריו  
וכיון שהפסגה היא 'א' וכל האותיות שאחריה הן 'ב' וכל האותיות שאחריהן הן 'ג' וכו'  
ואם יקפס 'א' אחר 'ב' וכל האותיות שאחריה הן 'ג' וכו' וכל האותיות שאחריהן הן 'ד' וכו'  
הוא לא שרר את האות 'א' היא : new\_str[i]='א'; תקינה שלב הבא

בשני הסעיפים הבאים של השאלה אנו מתייחסים למבנה (structure) שמוגדר באופן הבא:

```

struct _Data
{
    float result;
    struct _Data *next;
};
typedef struct _Data Data;

```

float

סעיף ב' (6 נקודות):  
נתונה הפונקציה הבאה:

✓ 5

```
Data *f(Data *info)
{
    Data item;
    item.result = info->result;
    item.next = info->next;
    return &item;
}
```

מה לדעתכם הפונקציה הזו אמורה לעשות? איזו בעיה יש בה? הסבירו בקצרה, ורישמו את הפונקציה לאחר תיקון הבעיה.

הפונקציה מקבלת שני נקודות זיכרון (Data \*info) (Data \*f) שהיא חוזרת עליהן (Data \*item) ומקצת זיכרון חדש. היא מנסה להעתיק את הנתונים מהנקודה info לנקודה item, אך היא לא מעתיקה את next. לכן, הפונקציה חוזרת עם נקודה חדשה, אך הנתון next לא נשמר. זה גורם לרצף של נקודות זיכרון שכל אחת מהן מכילה את הנתונים מהנקודה info, אך הנתון next הוא תמיד NULL. זה גורם לרצף של נקודות זיכרון שכל אחת מהן מכילה את הנתונים מהנקודה info, אך הנתון next הוא תמיד NULL. זה גורם לרצף של נקודות זיכרון שכל אחת מהן מכילה את הנתונים מהנקודה info, אך הנתון next הוא תמיד NULL.

\* פונקציה חוזרת על עצמה

```
Data *f(Data *info)
```

```
    Data item;
    item.result = info->result;
    item.next = info->next;
    return item;
```

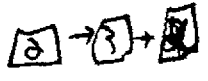
הפונקציה חוזרת על עצמה  
מחזירה את הנתון  
הנתון next הוא תמיד NULL  
זה גורם לרצף של נקודות זיכרון  
שכל אחת מהן מכילה את הנתונים מהנקודה info

סעיף ג' (6 בקורות):  
נתונה הפונקציה הבאה:

✓ 6

```
int secret(Data *info)
{
    int answer;
    if ((info == NULL) || (info->next == NULL))
        return 1;

    answer = (info->result == (info->next->result));
    return (answer && secret(info->next));
}
```



מה לדעתכם הפונקציה הזו עושה? נמקו בקצרה.

הפונקציה מחזירה 1 אם כל האנשים שהיא מקבלת כפרמטרים הם זוגיים, אחרת מחזירה 0. (אם יש אנשים שאינם זוגיים, אז היא איננה מקבלת אותם כלל.)

\_\_\_\_\_

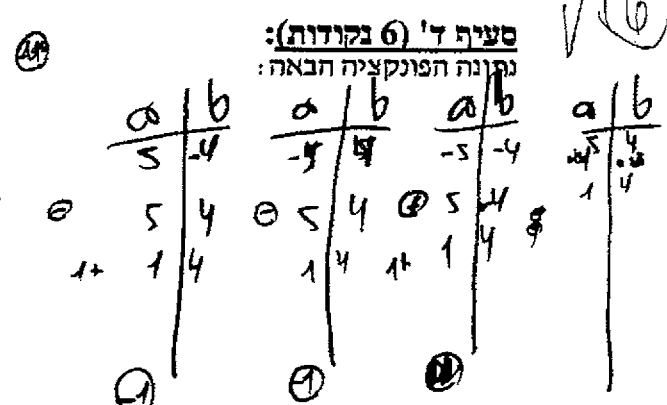
\_\_\_\_\_

```
int secret(int a, int b)
{
    if (a < 0)
        return -secret(-a, b);

    if (b < 0)
        return -secret(a, -b);

    if (a < b)
        return 0;

    return 1 + secret(a-b, b);
}
```



מה הפונקציה הזו מחזירה? האם היא תפעל בהצלחה עבור כל זוג מספרים שיועבר אליה? נמקו בקצרה.

הפונקציה מחזירה 1 אם a < b, אחרת מחזירה 0. (אם a < b, אז היא מחזירה 1 + secret(a-b, b).)

הפונקציה מחזירה 1 אם a < b, אחרת מחזירה 0. (אם a < b, אז היא מחזירה 1 + secret(a-b, b).)

סמסטר אי תשס"ו  
מועד : ב' 10/8/2006  
משך הבחינה : שלוש וחצי שעות  
אין להשתמש בחומר עזר

## בחינה בקורס תכנות ב- C

ממצים : נורית הספל, אמיתי ערמון

מתרגלים : אמיר אפשטיין, שחר הרוסי, צור יצחקיאן, אולגה סורקין,  
עופר פסטרנק, נירה שפיר

### הנחיות כלליות

- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- בכל סעיף ניתן להשתמש בפונקציות שהתבקשתם לכתוב בסעיפים קודמים, גם אם לא פתרתם את הסעיפים הקודמים.
- ניתן להשתמש בפונקציות ספריה, אלא אם נאמר אחרת בשאלה (זכרו לבצע #include מתאים כאשר אתם משתמשים בפונקציית ספריה).
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בכל עמוד.

**בהצלחה!**

**לשימוש הבודקים:**

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
	ציון

## שאלה מס' 1 (25 נקודות)

### סעיף א' (5 נקודות):

כיתבו פונקציה בעלת המפרט (prototype) הבא :

```
int factor_power(int a, int b);
```

הפונקציה מחזירה את החזקה הגבוהה ביותר של  $b$  שהמספר  $a$  מתחלק בה ללא שארית (אם  $a$  לא מתחלק ב- $b$  אז יוחזר 0).

לדוגמא:

עבור  $a=16, b=2$ , יוחזר 4, כי זו החזקה הכי גבוהה של 2 שהמספר 16 מתחלק בה.

עבור  $a=45, b=5$ , יוחזר 1, כי זו החזקה הכי גבוהה של 5 שהמספר 45 מתחלק בה.

עבור  $a=15, b=30$ , יוחזר 0, כי 15 לא מתחלק ב-30.

### סעיף ב' (10 נקודות):

כיתבו פונקציה בעלת המפרט (prototype) הבא :

```
int num_of_factors(int n);
```

הפונקציה מקבלת מספר שלם חיובי, ומחזירה את מספר הגורמים בפירוק שלו לגורמים ראשוניים.

לדוגמא:

עבור המספר 16 יוחזר 4, כי הפירוק שלו לגורמים ראשוניים הוא:  $2 * 2 * 2 * 2$

עבור המספר 45 יוחזר 3, כי הפירוק שלו לגורמים ראשוניים הוא:  $3 * 3 * 5$

עבור המספר 17 יוחזר 1, כי הפירוק שלו לגורמים ראשוניים הוא: 17

### סעיף ג' (10 נקודות):

כיתבו פונקציה בעלת המפרט (prototype) הבא :

```
int *find_factors(int n, int *n_factors);
```

הפונקציה מקבלת מספר שלם חיובי  $n$ , ומחזירה מצביע למערך שמכיל את הפירוק של  $n$  לגורמים ראשוניים. כמו-כן, מספר הגורמים (כלומר גודל המערך) יוכנס למשתנה ש- $n\_factors$  מצביע עליו.

לדוגמא: עבור המספר 16 יוחזר מערך שיכיל את המספרים:  $2 \ 2 \ 2 \ 2$

במקרה זה למשתנה ש- $n\_factors$  מצביע עליו יוכנס הערך 4.

**הערות:** לא ניתן להניח שהמספר  $n$  יותר קטן ממספר קבוע כלשהו. כמו-כן, לא ניתן להניח שמספר הגורמים של  $n$  יותר קטן ממספר קבוע כלשהו.

## שאלה מס' 2 (25 נקודות)

### סעיף א' (12 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
int chr_count(char *str, char ch);
```

הפונקציה תחזיר את מספר ההופעות של התו ch במחרוזת str.

#### דוגמאות:

(1) התו a מופיע 3 פעמים במחרוזת cbadaca

(2) התו g מופיע 0 פעמים במחרוזת cbadaca

הערה: בסעיף זה אין להשתמש בפונקציות מתוך הספרייה string.h.

### סעיף ב' (13 נקודות):

מחרוזת str2 ניקראת "אנגרמה" של מחרוזת str1 אם ניתן לקבל את str2 ע"י שינוי סדר התווים (פרמוטציה) של str1 (ראו דוגמאות להלן). כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int is_anagram(char *str1, char *str2);
```

הפונקציה תחזיר 1 אם המחרוזת str2 היא אנגרמה של המחרוזת str1, ואחרת היא תחזיר 0.

#### דוגמאות:

(1) המחרוזת "eleven plus two" היא אנגרמה של המחרוזת "twelve plus eleven"

(2) המחרוזת abcd איננה אנגרמה של המחרוזת abcda

(3) המחרוזת abcda איננה אנגרמה של המחרוזת abcdb

הערה: בסעיף זה אין להשתמש בפונקציות מתוך הספרייה string.h.

## שאלה מס' 3 (25 נקודות)

### סעיף א' (5 נקודות):

הגדירו מבנה (structure) בשם Book, שייצג ספר. המבנה יכלול את שם הספר, שם הסופר, שנת הפרסום, וההכנסות ממכירת הספר. כמו-כן, המבנה יאפשר יצירת רשימה-מקושרת של מבני Book. ניתן להניח שאורך כל אחד מהשמות הוא 80 תווים לכל היותר.

### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Book *max_book(Book *book_list);
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מבני Book. היא מחזירה מצביע למבנה ברשימה שמייצג את הספר שההכנסות ממנו היו הגבוהות ביותר.

### סעיף ג' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
Book *update_list(Book *book_list, Book *new_book);
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מבני Book (book\_list) ומצביע לספר נוסף (new\_book). אם הספר הנוסף לא נמצא ברשימה, הפונקציה מוסיפה אותו בסוף הרשימה (אחרת לא יתבצע ברשימה שום שינוי). הפונקציה מחזירה מצביע לרשימה המעודכנת.

הערה: ניתן להניח שהשוואת שם הספר מספיקה כדי לקבוע אם ספרים הם זהים. כמו-כן, אין צורך ליצור עותק חדש של המבנה המייצג את הספר הנוסף כאשר מוסיפים אותו לרשימה.

## שאלה מס' 4 (25 נקודות)

בשני הסעיפים הראשונים של השאלה אנו מתייחסים למבנה (structure) שמוגדר באופן הבא:

```
struct _Info
{
    double result;
    struct _Info *next;
};
typedef struct _Info Info;
```

### סעיף א' (7 נקודות):

נתונה הפונקציה הבאה:

```
void f(Info item, double result, Info *ptr)
{
    item.result = result;
    item.next = ptr;
    return;
}
```

מה לדעתכם הפונקציה הזו אמורה לעשות? איזו בעיה יש בה? נמקו בקצרה.

---

---

---

---

---

### סעיף ב' (6 נקודות):

נתונה הפונקציה הבאה:

```
int secret(Info *head)
{
    int solution;
    if ((head == NULL) || (head->next == NULL))
        return 1;

    solution = (head->result <= (head->next->result));
    return (secret(head->next) && solution);
}
```

מה לדעתכם הפונקציה הזו עושה? נמקו בקצרה.

---

---

---

---

---



**סעיף ג' (6) נקודות):**

מה עושה הפונקציה הבאה? נמקו בקצרה.

```
void secret(char *s)
{
    if(*s == '\\0') return;
    secret(s+1);
    putchar(*s);
}
```

---

---

---

---

---

**סעיף ד' (6) נקודות):**

נתונה הפונקציה הבאה :

```
int secret(int n)
{
    while(n > 1) {
        if(n % 2)
            return 0;
        n /= 2;
    }
    return 1;
}
```

מה עושה הפונקציה? נמקו בקצרה.

---

---

---

---

---

## הצעת פתרון למבחן בתכנות למהנדסים – מועד ב', סמסטר א' תשס"ו

שאלה מס' 1

סעיף א':

```
#include<stdio.h>
#include<stdlib.h>
int factor_power(int a, int b)
{
    int power=0, current;
    if ((b == 0) || (b == 1))
    {
        printf("Wrong factor: %d\n", b);
        exit(1);
    }
    for(current=b; a%current==0; current*=b)
        power++;
    return power;
}
```

סעיף ב':

```
int num_of_factors(int n)
{
    int i, j, power, n_factors=0;
    for (i = 2; n != 1; i++)
    {
        power = factor_power(n, i);
        n_factors += power;
        for (j=0; j < power; j++)
            n = n/i;
    }
    return n_factors;
}
```

סעיף ג':

```
#include<stdio.h>
#include<stdlib.h>
int *find_factors(int n, int *n_factors)
{
    int i, j, power, *array, index=0;
    if (n==1)
    {
        printf("No factorization of 1\n");
        exit(1);
    }
    *n_factors = num_of_factors(n);
    array = (int *) malloc ((*n_factors) * sizeof(int));
    if (array == NULL)
    {
        printf("Allocation failed\n");
        exit(1);
    }
    for(i=2; n!=1; i++)
    {
        power=factor_power(n,i);
        for (j=0; j < power; j++)
        {
            array[index]=i;
            index++;
            n /= i;
        }
    }
    return array;
}
```

## שאלה מס' 2

סעיף א':

```
int chr_count(char *str, char ch)
{
    int count=0;

    while(*str != '\0')
    {
        if (*str == ch)
            count++;
        str++;
    }
    return count;
}
```

סעיף ב':

```
int is_anagram(char *str1, char *str2)
{
    int i;

    for(i=0; str1[i] != '\0'; i++)
        if (chr_count(str1, str1[i]) != chr_count(str2, str2[i]))
            return 0;
    if (str2[i] != '\0') /* Different lengths */
        return 0;
    return 1;
}
```

## שאלה מס' 3

סעיף א':

```
struct _Book
{
    char name[81];
    char author[81];
    int year;
    double income;
    struct _Book *next;
}
typedef struct _Book Book;
```

סעיף ב':

```
Book *max_book(Book *book_list)
{
    Book *max;

    max = book_list;
    while(book_list != NULL)
    {
        if (max->income < book_list->income)
            max=book_list;
        book_list=book_list->next;
    }
    return max;
}
```

## סעיף ג':

```
#include<string.h>
#include<stdlib.h>
Book *update_list(Book *book_list, Book *new_book)
{
    Book *p_book;

    if (new_book == NULL)
        return book_list;
    if (book_list == NULL)
        return new_book;

    for(p_book = book_list; p_book != NULL; p_book = p_book->next)
    {
        if(strcmp(p_book->name, new_book->name)==0)
            return book_list;
        if (p_book->next==NULL)
            p_book->next=new_book; /* book_list will be returned */
    }
    /* in the next iteration */
}
```

(אפשר כמוכּן לפתור באופן דומה עם לולאת while במקום for)

## שאלה מס' 4:

### סעיף א':

הפונקציה אמורה לאתחל מבנה מסוג Info בערכים של ptr, result, אבל כיוון שמועבר אליה עותק של מבנה ולא כתובת של מבנה (והיא לא מחזירה שום דבר) השינויים שהיא מבצעת לא נשמרים, כלומר בפועל היא לא משנה שום דבר.

### סעיף ב':

הפונקציה מקבלת רשימה מקושרת של מבני Info ובודקת אם היא ממוינת בסדר עולה של ערכי ה-result. אם כן אז מוחזר 1 ואחרת 0. הבדיקה נעשית באופן רקורסיבי, כשכל פעם מושווה איבר אחד ברשימה לאיבר הבא אחריו, עד שמגיעים לסוף הרשימה.

### סעיף ג':

הפונקציה מקבלת מחרוזת ומדפיסה אותה הפוך (מהסוף להתחלה). זה משום שפקודת ההדפסה נמצאת אחרי הקריאה הרקורסיבית, כך שמגיעים אליה בסדר הפוך מאשר סדר הקריאות לפונקציה (קודם מגיעים להדפסה של התו האחרון בעקבות תנאי העצירה, אז מגיעים להדפסה של התו הקודם, וכך הלאה).

### סעיף ד':

הפונקציה בודקת אם המספר החיובי שהועבר אליה הוא חזקה של 2 (אם כן מוחזר 1 ואחרת 0). נסביר זאת בקצרה. כל עוד המספר גדול מ-1 בודקים אם הוא מתחלק ב-2 ללא שארית, ואם לא אז מחזירים 0 (כי אז יש לו גורם שאיננו 2). אם כן אז מחלקים אותו ב-2. אם הגענו ל-1 רק ע"י חלוקות ב-2 אז המספר הוא חזקה של 2.

סמטסר אי תשס"ה  
מועד: מיוחד 4/11/2005  
משך הבחינה: שלוש וחצי שעות  
חומר עזר: כל חומר עזר כתוב למעט  
ספרים (או צילום של ספרים)

## בחינה בקורס C למהנדסים

**מרצים:** אייל אבן-דר, רון ויין, אמיתי ערמון  
**מתרגלים:** אנגילה אנוש, שי גוטנר, נורית הספל, בן זנדבנק, דרור עירוני, לעם רוזיטי

### הנחיות כלליות

- במבחן זה 4 שאלות, ויש להשיב על כולן.
- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- ניתן להשתמש בפונקציות ספריה (בעיקר פונקציות קלט/פלט – זכרו לבצע `#include` מתאים).
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

**לשימוש הבודקים בלבד:**

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
92	ציון

**שאלה מס' 1 (25 נקודות)**

נתונה סדרת מספרים שלמים חיוביים.  $a_1, a_2, a_3, \dots, a_i, a_{i+1}, \dots$ . כל איבר  $a_i$  בסדרה מקיים:

$$a_i = \sum_{j=k}^t a_j$$

עבור  $k$  ו- $t$  כלשהם.

כתוב תוכנית המקבלת מספרים שלמים חיוביים כל עוד הם מקיימים את הגדרת הסדרה. ברגע שמגיע מספר אשר אינו מקיים את ההגדרה התוכנית מסתיימת.

הכלל:  $a_i = a_k + \dots + a_t$   
 כלומר:  $a_i > a_k$  ו- $a_i > a_t$

פתרון שאלה 1

```

for (i=0, d=0; i<n; i++)
{
    for (j=i; j<n-i; j++)
    {
        d=d+arr1[j];
        if (d==b)
            e=d;
    }
    if (e==b)
        arr1[n-i]=b;
}

```

$\rightarrow$   $arr2 = null$   
 $c=0;$

```

arr2 = (int*) malloc (sizeof(int)*n);
for (i=0; i<n; i++)
    arr2[i]=arr1[i];
n++;
if (d==b)
    c=0;
}

```

```

for (i=0; i<a; i++)
{
for (j=i; j<a-i; j++)
{
d = d + arr2[j];
if (d == b)
e = d;
}
}
if (e == b)
arr1[a] = b;
arr2 = arr1;


```

```

#include <stdio.h>
void main()
{
int *arr1, *arr2, a, b, c=1, n=2;
printf("Enter number n");
scanf("%d", &a);
while (c==1)
{
printf("Enter number n");
scanf("%d", &b);
arr1 = (int*) malloc(sizeof(int) * n+1);
if (arr1 == null)
c=0;

```

```

if (arr2 != null) {
arr1[0] = a;
for (i=0; i<n; i++)
arr1[i] = arr2[i];
}
free(arr2);

```

## שאלה מס' 2 (25 נקודות)

נתון מערך בגודל 10000 אשר כל איבריו נמצאים בתחום 1-100, כתבו תוכנית הממיינת את המערך כאשר ניתן לגשת לכל תא במערך פעמיים לכל היותר. אסור להשתמש במערך נוסף בגודל 10000.

פתרון שאלה 2

```
#include <stdio.h>
void main()
{
    int arr[101] = {0}, i, b_arr[10,000] = {0}, j=1, a=0;
    for (i=0; i<10,000; i++)
        arr[b_arr[i]] = arr[b_arr[i]] + 1;
    for (j=1; j<101; j++)
    {
        for (i=0; i<arr[j]; i++)
        {
            b_arr[a] = j;
            a++;
        }
    }
}
```





### שאלה מס' 3 (25 נקודות)

כתבו פונקציה המקבלת מחרוזת. רצף של תווים במחרוזת המסתיים ברווח מהווה מילה. הפונקציה תבדוק האם קיימת מילה המופיעה פעמיים במחרוזת. אם כן תחזיר 1 אם לא תחזיר 0. לדוגמא:  
 The The dog 12 is clever  
 יחזור 1

```
#include <string.h>
```

פתרון שאלה 3

```
int w_repeat(char *str)
{
    int n=0, s, count=1, c;
    char *point;
    s = strlen(str);
    for (i=0; i<s; i++)
    {
        if (str[i] == ' ')
            n++;
    }
    point = (char *) malloc (sizeof(char) * n);
    point[0] = str;
    for (i=1; i<n; i++)
    {
        for (; *str != '\0'; str++);
        point[i] = str+1;
    }
    for (i=0; i<n-1; i++)
    {
        while (point[i] != '-' && point[i] != '\0')
            point[i]++;
        point[i] = '\0';
    }
    for (i=0; i<n; i++)
    {
        for (j=1; j<n; j++)
        {
            c = strcmp (point[i], point[j]);
            if (c=0)
                count++;
        }
    }
    if (count == 2)
        return (1);
    return (0);
}
```

\* המחרוזת ' ' כלל אינה נחשבת למילה כי כן המחרוזת ריקה.

לדוגמה כן (כלל) point[i] = \*str

3

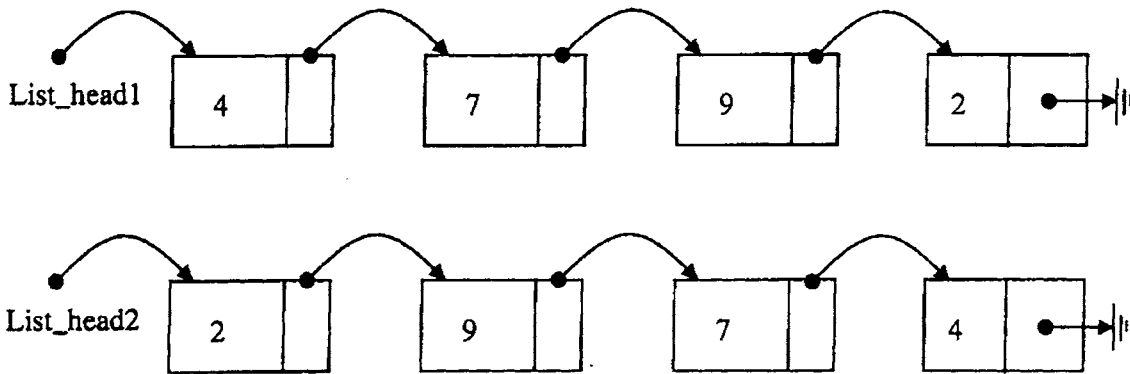
כנראה  
 המחרוזת ' ' כלל אינה נחשבת למילה כי כן המחרוזת ריקה.  
 (3)  
 למה נחשבת מילה  
 כלל אינה נחשבת למילה כי כן המחרוזת ריקה.

## שאלה מס' 4 (25 נקודות)

א. (7 נקודות)

הגדר טיפוס נתונים בשם `Item_t` שמשתנה מסוגו ישמש ליצוג אבר ברשימה מקושרת המחזיקה מספרים ממשיים (`double`).

ב. כתוב פונקציה המקבלת מצביעים לשתי רשימות מקושרות. הפונקציה תחזיר 1 אם הרשימות הן הופכיות זו לזו, 0 אם לא. רשימות מוגדרות להיות הופכיות אם האיברים של רשימה אחת מופיעים בסדר הפוך ברשימה השנייה למשל:



```

1. struct double {
    double n;
    struct double *next;
};
typedef struct double double;

```

2.

```
for (i=0; i<a; i++)
{
    arr1[i] = head_1->n;
    head_1 = head_1->next;
}

for (i=a-1; i>=0; i--)
{
    arr2[i] = head_2->n;
    head_2 = head_2->next;
}

for (i=0; i<a; i++)
{
    if (arr1[i] != arr2[i])
        return (0);
}
return (1);
}
```

✓

```

11. struct double {
    double n;
    struct double *next;
}

```

```

typedef struct double double

```

```

2. int flip (double *head1, double *head2)
{

```

```

    int i=0, a=0, *arr1, *arr2;
    double *head-1, *head-2;
    while (head1 != null)
    {
        head-1 = head1;
        head-2 = head2;

```

```

        i++;

```

```

        head1 = head1->next;

```

```

    }

```

```

    while (head2 != null)

```

```

    {

```

```

        a++;

```

```

        head2 = head2->next;

```

```

    }

```

```

    if (a != i)

```

```

        return (0);

```

```

    arr1 = (int*) malloc (sizeof(int)*a)

```

```

    arr2 = (int*) malloc (sizeof(int)*a)

```

סמסטר אי+ב' תשס"ה  
מועד: ב' 31/7/2005  
משך הבחינה: שלוש שעות  
חומר עזר: כל חומר עזר כתוב למעט  
ספרים (או צילום של ספרים)

## בחינה בקורס C למהנדסים

**ממצים:** אייל אבן-דר, רון ויין, אמיתי ערמון  
**מתרגלים:** אנג'לה אנוש, שי גוטנר, נורית הספל, בן זנדבנק, דרור עירוני, ליעם רודיטי

### הנחיות כלליות

- במבחן זה 4 שאלות, ויש להשיב על כולן.
- משקל כל שאלה 25 נקודות.
- בשאלה מס' 1 יש לענות בהסבר קצר על כל סעיף. בכל השאלות האחרות נדרש כפתרון קוד C ולא הסבר במילים.
- בשאלות 2-4 רצוי שהפתרון יהיה יעיל ככל האפשר.
- את הפתרונות יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- ניתן להשתמש בפונקציות ספריה (זכרו לבצע `#include` מתאים).
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

### **לשימוש הבודקים בלבד:**

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
	ציון

## שאלה מס' 1 (25 נקודות)

### סעיף א' (8 נקודות):

מה עושה הפונקציה הבאה?

אילו הנחות היא מניחה על המצביע שמועבר אליה ועל הטיפוס Data?

```
int secret (Data *ptr, double num)
{
    int val;

    if (ptr == NULL)
        return 0;
    val = (ptr->info == num) || secret(ptr->next, num);

    return val;
}
```

---

---

---

---

---

### סעיף ב' (9 נקודות):

נתונות פונקציה ותוכנית שמתמשת בה :

```
#include <stdio.h>

void f(int *a, int *b, int size)
{
    int *temp, i=0;
    while(i<size)
    {
        *(temp+i) = *(b+i);
        *(b+i) = *(a+i);
        *(a+i) = *(temp+i);
        ++i;
    }
}

int main()
{
    int a[] = {1,2,3,4,5,6,7,8,9};
    int b[] = {5,4,3,2,1,0,1,2};
    int i, min;
    min=sizeof(a);
    if (min > sizeof(b))
```

```
    min=sizeof(b);
    f(a,b,min);
    for(i = 0; i < min; i++)
        printf("%d %d\n", a[i], b[i]);
    return 0;
}
```

מה לדעתכם הפונקציה f אמורה לעשות?  
האם לדעתכם התוכנית תתפקד כראוי? אם יש בעיה/ות - כיצד ניתן לתקן זאת?

---

---

---

---

---

**סעיף ג' (8 נקודות):**

מה עושה הפונקציה הבאה?

איך ניתן לשנות בה שני תווים בדיוק (לא תווי רווח), כך שלא יחול שינוי בתיפקודה? (כלומר כך שלאחר השינוי בתווים האלה, עבור כל זוג מספרים שיועבר לפונקציה היא תחזיר בדיוק אותו מספר כמו לפני השינוי בתווים האלה)?

```
int secret(int a, int b)
{
    if (b < 0)
        return secret(-a, -b);
    if (b == 1)
        return a;
    else
        return a + secret(a, b - 1);
}
```

---

---

---

---

---

## שאלה מס' 2 (25 נקודות)

משחק הסודוקו מוגדר באמצעות לוח בגודל  $9 \times 9$  (כלומר 9 שורות ו-9 עמודות). בפתרון חוקי של המשחק, כל ספרה בין 1 ל-9 מופיעה פעם אחת בכל שורה, פעם אחת בכל עמודה ופעם אחת בכל תת-ריבוע מודגש (ראו שרטוט להלן).

כתבו פונקציה המקבלת הצעה לפתרון של משחק הסודוקו (כלומר מערך דו-מימדי מתאים) ובודקת האם הוא פתרון חוקי. על הפונקציה להיות בעלת המפרט (prototype) הבא :

```
int is_legal_sudoku(int board[9][9]);
```

ניתן להניח שהמערך המועבר לפונקציה הוא אכן בגודל  $9 \times 9$ , ושכל תא בו מכיל מספר בין 1 ל-9. בשרטוט שלהלן יש דוגמא לפתרון חוקי :

1	2	3	4	5	6	7	8	9
9	8	7	1	2	3	4	5	6
4	5	6	9	8	7	1	2	3
2	3	1	5	6	4	8	9	7
8	7	9	2	3	1	5	6	4
5	6	4	8	7	9	2	3	1
3	1	2	6	4	5	9	7	8
7	9	8	3	1	2	6	4	5
6	4	5	7	9	8	3	1	2



### שאלה מס' 3 (25 נקודות)

כתבו פונקציה המקבלת מחרוזת שמכילה מילים וספרות, ומחזירה מחרוזת חדשה, שבה כל ספרה מוחלפת במילה האנגלית המתאימה (ראו דוגמא להלן).

לדוגמא, אם המחרוזת שמועברת לפונקציה היא:

"I am number 1, you are number 2"

אז המחרוזת שמוחזרת צריכה להיות:

"I am number one, you are number two"

הערה: לא ניתן להניח חסם על אורכי המחרוזות.

## שאלה מס' 4 (25 נקודות)

### סעיף א' (5 נקודות):

הגדירו מבנה (structure) בשם Date, שמייצג תאריך, וכולל יום, חודש, ושנה. הגדירו מבנה נוסף בשם Birthday, שמייצג יום-הולדת, וכולל שם ותאריך-לידה, כך שניתן יהיה ליצור רשימה מקושרת של מבני Birthday.

### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
Birthday *date_subset(Birthday *ptr, Date day);
```

הפונקציה מקבלת מצביע לרשימת ימי-הולדת ותאריך, ומחזירה רשימה חדשה, שכוללת רק את ימי-ההולדת שזהו תאריך-הלידה שלהם. הרשימה המקורית לא משתנה. **הערה: לא ניתן** להניח שהרשימה המקורית ממויינת באופן כלשהו. ניתן להניח שהיא מסתיימת ב-NULL.

### סעיף ג' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
void extract_date(Birthday *ptr, Date day);
```

הפונקציה מקבלת רשימת ימי-הולדת ותאריך. הפונקציה **מדפיסה** את שמותיהם של מי שנולדו בתאריך-הלידה הזה, **ומוחקת מהרשימה** את ימי-ההולדת האלה. **שוב, לא ניתן** להניח שהרשימה המקורית ממויינת, **וניתן להניח** שהיא מסתיימת ב-NULL.

## פתרון למבחן בתכנות למהנדסים – מועד ב' תשס"ה, 31/7/05

(מאת אמתי ערמון)

### שאלה מס' 1:

סעיף א':

הפונקציה מניחה שהטיפוס Data הוא מבנה, שמכיל שדה מטיפוס double בשם info ושדה מטיפוס Data\* בשם next. כמו-כן, הפונקציה מניחה שהמזביע שמועבר אליה מצביע על רשימה מקושרת של מבני Data, ושהאיבר האחרון ברשימה מצביע ל-NULL. הפונקציה בודקת אם הערך num שמועבר אליה נמצא ברשימה. אם כן, מוחזר 1, ואחרת 0.

סעיף ב':

הפונקציה אמורה להחליף בין size הערכים הראשונים של המערכים a ו-b. אולם היא צפויה לגרום לתעופה, כיוון ש-temp מצביע לזיכרון שלא הוקצה. יש להוסיף לה כפקודה ראשונה:

```
temp=(int *) malloc (size*sizeof(int));
```

כמו-כן, בתוכנית יש להוסיף לפני הקריאה לפונקציה את השורה:

```
min=min/sizeof(int);
```

אחרת תיגרם תעופה עקב חריגה מגבולות המערך.

בנוסף, אין `#include<stdlib.h>`.

סעיף ג':

הפונקציה מחזירה את הערך של  $a*b$ . ניתן למשל לכתוב: `if (b==0) return 0;` או להחליף בין a ל-b בשורה הראשונה, מבלי שהתיפקוד של הפונקציה ישתנה.

### שאלה מס' 2:

להלן אפשרות לפתרון (הערות לא נדרשו במבחן):

```
void zero_array(int *array, int size) /* zeroing an int array */
{
    int i;
    for(i=0; i<size; i++)
        array[i]=0;
}

int check_legal_sudoku(int board[9][9])
{
    int row_digits[10], col_digits[10], sq_digits[10], i, j;

    for(i=0; i<9; i++)
    {
        zero_array(row_digits,10); /*Initially no digit appeared. */
        zero_array(col_digits,10); /*Afterwards the value will be 1 */
        zero_array(sq_digits,10); /*for the digits which appeared. */

        for(j=0; j<9; j++)
        {
            /* Checking row i and updating the row digits array*/
            if (row_digits[board[i][j]]) return 0;
            row_digits[board[i][j]]++;

            /* Checking column i and updating the columns array*/
            if (col_digits[board[j][i]]) return 0;
            col_digits[board[j][i]]++;

            /* Checking a 3x3 square and updating the square array*/
            if (sq_digits[board[3*(i/3)+j/3][3*(i%3)+j%3]]) return 0;
            sq_digits[board[3*(i/3)+j/3][3*(i%3)+j%3]]++;
        }
    }

    return 1;
}
```

**שאלה מס' 3:**  
להלן פתרון אפשרי:

```
#include <stdio.h>
#include <string.h>

int is_digit(char c) {return ((c >= '0') && (c <= '9'))}

char *Digits2Words(char *str)
{
    char *digit_words[10]={"zero","one", "two", "three", "four",
                          "five", "six", "seven", "eight", "nine"};

    char *new_str; /* The new string */
    int i, j=0, k, word_len, len_to_add=0, len=strlen(str);

    /* Calculating the required extra length for the new string */
    for (i=0; i<len; i++)
        if (is_digit(str[i]))
            len_to_add += (strlen(digit_words[i-'0'])-1);

    /* Allocating the new string (including the '\0') */
    new_str=(char *) malloc ((len+len_to_add+1)*sizeof(char));
    if (new_str == NULL)
    {
        printf("String allocation failed\n");
        exit(1);
    }

    /* Copying the string and changing digits to words */
    for(i=0; i<len; i++)
    {
        if (! is_digit(str[i])) /* If it's not a digit - just copy*/
        {
            new_str[j]=str[i];
            j++;
            continue;
        }

        /* If it's a digit - copy the relevant word */
        word_len=strlen(digit_words[i-'0']);
        for(k=0; k < word_len; k++)
        {
            new_str[j]=digit_words[i-'0'][k];
            j++;
        }
    }

    new_str[len+len_to_add]='\0'; /* '\0' at the end */
    return new_str;
}
```

**שאלה מס' 4:**  
סעיף א': [התקבל גם שימוש במערך עבור השם]

```
typedef struct _Date
{
    int    day;
    int    month;
    int    year;
} Date;
```

```

typedef struct _Birthday
{
    char *name;
    Date day;
    struct _Birthday* next;
} Birthday;

```

סעיף ב':

```

#include <stdlib.h>
#include <string.h>

int equal_dates(Date a, Date b)
{
    return ((a.year == b.year) && (a.month == b.month)
        && (a.day == b.day));
}

Birthday *date_subset (Birthday* ptr, Date day)
{
    Birthday *new_head=NULL, *new_ptr, *last_ptr;

    while (ptr != NULL)
    {
        if (equal_dates(ptr.date, day)) /* It's the required date */
        {
            /* Allocating a new Birthday */
            new_ptr=(Birthday *) malloc (sizeof(Birthday));
            if (new_ptr==NULL)
            {
                printf("Allocation failed.\n");
                exit(1);
            }
            new_ptr->next=NULL;

            /* Copying the date */
            (new_ptr->date).year=(ptr->date).year;
            (new_ptr->date).month=(ptr->date).month;
            (new_ptr->date).day=(ptr->date).day;

            /* Copying the name */
            new_ptr->name=(char *)malloc((strlen(ptr->name)+1)
                *sizeof(char));
            strcpy(new_ptr->name, ptr->name);

            /* Updating the new list */
            if (new_head==NULL) /* If it's the first in the new list */
                new_head=new_ptr;
            else
                last_ptr->next=new_ptr; /* Adding at the end */

            last_ptr=new_ptr; /* It's the last element of the new list.*/
            ptr=ptr->next; /* Moving to the next element of the */
        } /* original list. */
    }
    return new_head;
}

```

```
#include <stdlib.h>
#include <stdio.h>

void extract_date (Birthday *ptr, Date day)
{
    Birthday *prev_ptr=ptr;

    while(ptr != NULL)    /* A loop over the linked list */
    {
        if (equal_dates(ptr->date, day)) /*It's the required date*/
        {
            /* Printing and freeing the name */
            printf("%s\n", (ptr->date).name);
            free((ptr->date).name);

            if (ptr==prev_ptr) /* The first birthday in the list*/
            {
                prev_ptr=ptr->next;
                free(ptr);
                ptr=prev_ptr;
            }
            else /* Not the first birthday in the list */
            {
                prev_ptr->next=ptr->next;
                free(ptr);
                ptr=prev_ptr->next;
            }
        }
        else /* going forward without printing or deleting */
        {
            prev_ptr=ptr;
            ptr=ptr->next;
        }
    }

    return;
}
```

שאלה מס' 1 (25 נקודות)

סעיף א' (5 נקודות):

בסעיף זה של השאלה אנו מתייחסים למבנה (structure) שהוגדר באופן הבא:

```
struct _Student
{
    char *name; /* The student's name. */
    float average; /* The student's average grade. */
};
typedef struct _Student Student;
```

כמו-כן נתונה לנו הפונקציה:

```
void init_student (Student *p_student,
                  char *name, float average)
{
    p_student->name = name;
    p_student->average = average;
    return;
}
```

התכנית הבאה אמורה לקלוט שמות וממוצעים של סטודנטים, ולאחר מכן להדפיס את השם והממוצע של כל סטודנט:

```
#include<stdio.h>
#define N_STUDENTS 30
#define MAX_LEN 40

int main ()
{
    Student class[N_STUDENTS];
    char name[MAX_LEN + 1];
    float average;
    int i;

    printf("Enter the names and averages\n");
    for(i = 0; i < N_STUDENTS; i++)
    {
        scanf ("%s %f", name, &average);
        init_student (&(class[i]), name, average);
    }

    for (i = 0; i < N_STUDENTS; i++)
        printf("%s: %f", class[i].name,
              class[i].average);

    return 0;
}
```

סעיף 2 →

סעיף 2 →  
סעיף 3 →

20 1  
20 2  
25 3  
23 4

התכנית לא מתפקדת כראוי, למרות שהקלט שהיא מקבלת תקין (כלומר כל השמות שהוכנסו אינם ארוכים מ-40 תווים וכל הציונים הם חוקיים). מה הבעיה?

הבעיה היא שרק  $\sqrt{\text{scanf}}$  איתנו וישנים כל פעם

אם data לא יקרא name אזי הנתונים לא יתקבלו

init-student, משהוא לא יקרא data אזי הנתונים לא יתקבלו

אם לא יקרא name הנתונים לא יתקבלו

ישננים של הנתונים אלא הנתונים לא יתקבלו

הוא לא היה על הסטודנטים. בסוף כל קריאה של הנתונים הוא שיער ב' (5 נקודות): שם הסטודנטים יתקבלו ויהיה אולי שם, תיקנו את סעיף אי ללא שינוי המבנה (כתבו רק את השורות שמשתנות או מתווספות).

class name [N-STUDENTS][MAX\_LEN+1];

1 char name [N-STUDENTS][MAX\_LEN+1];

2 scanf ("%s %f", &name[i], &average);

3 init-student (&class[i], &name[i], average);

סעיף ג' (5 נקודות):

מה עושה הפונקציה הבאה? כמו-כן, איזו הנחה היא מניחה על המערך שמועבר אליה?

```
int func (int *arr, int n,
          int place, int val)
```

```
{
    int i;

    if (place < 0 || place > n)
        return 0;

    for (i = n; i > place; i--)
        arr[i] = arr[i - 1];
    arr[place] = val;

    return 1;
}
```

3  
הנתונים יתקבלו  
זקף  $h+1 \leq$

הפונקציה "מניחה" את הנתונים באינדקס הנתונים באמצעות

אילו וקטורים "place" מכניסה את האינדקס "val".

(האינדקס האחרון קטן (נרס). הפונקציה היא שאינה

פונקציה את האינדקס האחרון קטן ומסמך את האינדקס האחרון

הפונקציה מניחה את הנתונים באינדקס place אילו קטן מ-8 ואת





סעיף ה' (5 בקודות):

מה עושה הפונקציה הבאה?

כמו-כן, איזו הנחה היא מניחה על המחרוזת שמועברת אליה?

```
#include <string.h>
int secret (char *str)
{
    int    val, len;

    if (str[0]== '\0')
        return 0;

    len = strlen (str);
    val = str[len - 1] - '0';
    str[len - 1] = '\0';
    return (val + secret(str)*10);
}
```

הפונקציה מקבלת מחרוזת ומחזירה מספר  
ס'ן 0-9 (זאת קצתם התוודע סבאחאללה  
יהיו רק ספרות בין 0-9) ומחזירה את  
המספר שנוצא קמחולת המספר integer.

## שאלה מס' 2 (25 נקודות)

### סעיף א' (17 נקודות):

כתבו תכנית הקולטת מספרים שלמים מן המשתמש, עד אשר מתקבל מספר שלילי. על התכנית להדפיס את עשרים המספרים המקסימליים מבין אלה שנקלטו, בסדר כלשהו (כולל חזרות, כלומר אם למשל המספר המקסימלי הופיע שלוש פעמים אז הוא יודפס שלוש פעמים). אם הסדרה מכילה פחות מעשרים מספרים אז יש להדפיס את כל המספרים שנקלטו, לא כולל המספר השלילי.

ניתן להניח כי הקלט תקין.

לדוגמא, אם הקלט הוא:

21 20 19 18 17 16 15 14 10 11 12 13 1 2 3 4 5 6 7 8 9 22 23 23 -1

אז הפלט יהיה למשל:

21 20 19 18 17 16 15 14 10 11 12 13 5 6 7 8 9 22 23 23

### סעיף ב' (8 נקודות):

רוצים לשנות את התכנית מסעיף א', כך שהיא תדפיס את  $n$  המספרים המקסימליים, כאשר  $n$  מתקבל כקלט מהמשתמש בתחילת התכנית (כלומר יתכן מספר חיובי כלשהו, ולא דווקא 20) ואחר כך מתקבלת סדרת המספרים. כתבו רק את חשורות שנוספות או משתנות בתכנית מסעיף א' (אם לא פתרתם את סעיף א', כתבו את כל התכנית). ניתן להניח כי הקלט תקין.

```
#include <stdio.h>
#define MAX_NUM 20

int main()
{
    int arr[MAX_NUM];
    int num, i=0, k=0, end;
    for (i=0; i<MAX_NUM; i++)
        arr[i] = -1;
    i=0;
    scanf("%d", &num);
    while (num >= 0)
    {
        if (i < MAX_NUM)
            arr[i] = num;
        else
        {
            end = 0; k = 0;
            while ((k < MAX_NUM) && (!end))
            {
                if (num > arr[k])
                {
                    arr[k] = num;
                    end = 1;
                }
                k++;
            }
        }
        i++;
        scanf("%d", &num);
    }
}
```

הערות ידניות:

- בסעיף א' (17 נקודות) - יש להדפיס את עשרים המספרים המקסימליים.
- בסעיף ב' (8 נקודות) - יש לשנות את התכנית כדי להדפיס את  $n$  המספרים המקסימליים.
- הקלט הוא מספרים שלמים, עד אשר מתקבל מספר שלילי.
- הפלט הוא המספרים המקסימליים, בסדר כלשהו (כולל חזרות).

Page 100

```

for (i=0; i<MAX_NUM; i++)
{
    if (arr[i] >= 0)
        printf("%d ", arr[i]);
}
return (0);
}

```

2130k, 17'82? 83k (P)  
 2703m 20747 nks \*

```

#include <stdio.h>
#include <stdlib.h>

```

```

void main()

```

```

{
    int num, i=0, k=0, end;
    int *arr;
    int MAX_NUM;

    scanf("%d", &MAX_NUM);
    arr = (int *) malloc(sizeof(int)*MAX_NUM);
    if (arr == NULL)
    {
        printf("Allocation failed\n");
        return (1);
    }
}

```

```

free(arr);
return(0);
}

```

## שאלה מס' 3 (25 נקודות)

### סעיף א' (17 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int is_similar_str (char *str1, char *str2);
```

הפונקציה בודקת אם שתי מחרוזות שמועברות אליה הן "דומות", כפי שמוגדר להלן, ומחזירה 1 אם כן, ואחרת 0. מחרוזות str1 ו-str2 נחשבות "דומות" אם מתקיים אחד מן התנאים הבאים:

(1) המחרוזות זהות.

(2) המחרוזות שונות זו מזו בתו אחד בלבד.

(3) המחרוזות שונות זו מזו רק בשני תווים סמוכים, שמופיעים בהן בסדר הפוך (כלומר אם מחליפים בין שני התווים האלה ב-str2 מקבלים את str1).

דוגמאות למחרוזות דומות:

abcd	best	ba	water
abcd	bets	ab	woter

הערה: שימו לב שמחרוזות שאורכיהן שונים הן בהכרח לא דומות. כמו-כן, מחרוזות ששונות זו מזו ביותר משני תווים הן בהכרח לא דומות.

### סעיף ב' (8 נקודות):

כתבו תוכנית שקולטת מהמשתמש שתי מחרוזות (שתיהן באורך 80 לכל היותר), משתמשת בפונקציה מסעיף א' כדי לבדוק אם הן דומות, ומדפיסה הודעה מתאימה (אתם רשאים להשתמש בפונקציה is\_similar\_str גם אם לא פתרם את סעיף א').

הערה: אתם יכולים להניח כי הקלט שהמשתמש מכניס הוא תקין.

```
#include <string.h>

int is_similar_str(char *str1, char *str2)
{
    int diff = 0, index;
    int i, n;
    if (strlen(str1) != strlen(str2))
        return 0;
    n = strlen(str1);
    for (i = 1; i < n; i++)
    {
        if (str1[i] != str2[i])
        {
            diff++;
            if (diff == 1)
                index = i;
        }
    }
}
```

Ⓔ

```

    if ((diff == 0) || (diff == 1))
        return (1);
    if (diff == 2)
        if (str1[index] == str2[index+1])
            if (str1[index+1] == str2[index])
                return (1);
    return (0);
}

```

~~string str1, str2, str3, str4, str5, str6, str7, str8, str9, str10, str11, str12, str13, str14, str15, str16, str17, str18, str19, str20, str21, str22, str23, str24, str25, str26, str27, str28, str29, str30, str31, str32, str33, str34, str35, str36, str37, str38, str39, str40, str41, str42, str43, str44, str45, str46, str47, str48, str49, str50, str51, str52, str53, str54, str55, str56, str57, str58, str59, str60, str61, str62, str63, str64, str65, str66, str67, str68, str69, str70, str71, str72, str73, str74, str75, str76, str77, str78, str79, str80, str81, str82, str83, str84, str85, str86, str87, str88, str89, str90, str91, str92, str93, str94, str95, str96, str97, str98, str99, str100~~

①

```

#include <stdio.h>
#include <string.h>

#define MAX 80
int is_similar_str(char *str1, char *str2);
int main()
{
    char str1 [MAX+1];
    char str2 [MAX+1];

    printf("Type 2 strings");
    scanf("%s", str1);
    scanf("%s", str2);
    if (is_similar_str(str1, str2))
        printf("Similar!");
    else
        printf("No similar!");

    return (0);
}

```

## שאלה מס' 4 (25 נקודות)

### סעיף א' (7 נקודות):

הגדירו שני מבנים (structures): האחד, בשם Point אמור לייצג נקודה במישור בעלת שתי קואורדינטות ממשיות. השני, בשם Polygon מייצג מצולע (פוליגון) – המאופיין ע"י רשימה מקושרת של הקודקודים שלו, וכן ע"י שדה המציין את מספר הקודקודים ברשימה. אתם רשאים להגדיר מבני-עזר נוספים.

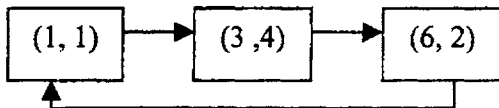
### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

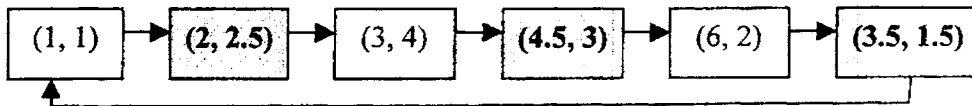
```
void add_polygon_vertices (Polygon *poly);
```

הפונקציה מקבלת מצביע לפוליגון והיא מכפילה את מספר הקודקודים בפוליגון על-ידי כך שהיא מוסיפה קודקוד חדש בין כל שני קודקודים קיימים. הקודקוד החדש הוא נקודת האמצע בין שני הקודקודים הקיימים. עליכם להניח שרשימת קודקודי המצולע היא "רשימה מקושרת מעגלית", כלומר האיבר האחרון ברשימה מצביע על האיבר הראשון שלה (ואיננו מכיל את הערך NULL).

לדוגמה, אם נתון המצולע הבא כקלט (משולש במקרה שלנו):



הפונקציה הופכת את המצולע להיות:

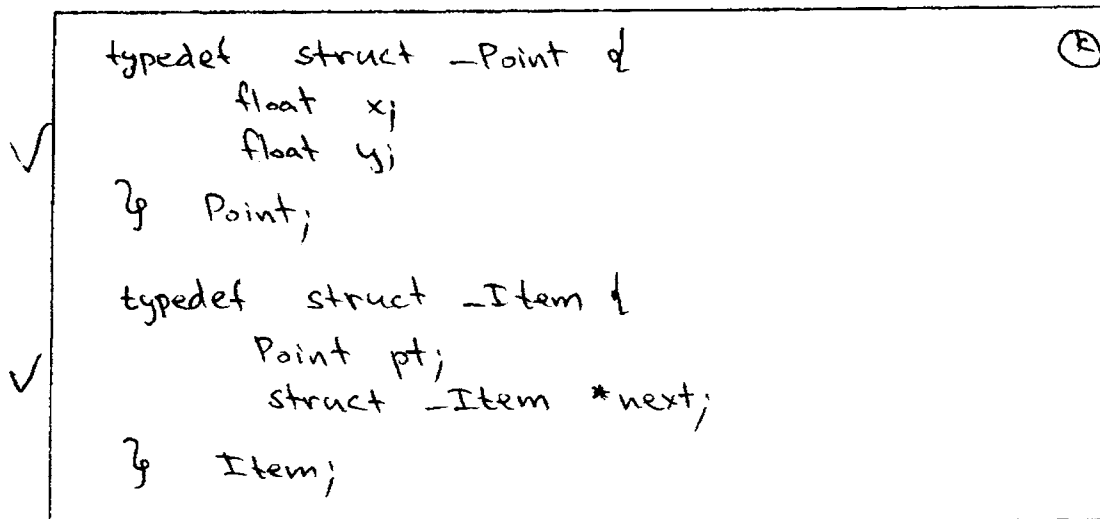


### סעיף ג' (8 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
void reset_polygon (Polygon *poly);
```

על הפונקציה "לנקות" את המצולע, כלומר לאפס את מספר הקודקודים שלו ולשחרר את רשימת הקודקודים שלו. עליכם להניח שרשימת הקודקודים היא מעגלית, כלומר האיבר האחרון ברשימה מצביע על האיבר הראשון שלה (ואיננו מכיל את הערך NULL).



```

typedef struct _Polygon {
    Item *head;
    int num;
} Polygon;

```

```

#include <stdlib.h>
void add_polygon_vertices (Polygon *poly)
{
    int i=0;
    Item *p, *tmp, *new;
    int n;

    p = poly->head;
    n = poly->num;
    while (i < n)
    {
        new = (Item*) malloc (sizeof(Item));
        tmp = p->next;
        p->next = new;
        new->next = tmp;
        new->pt.x = (p->pt.x + tmp->pt.x) / 2;
        new->pt.y = (p->pt.y + tmp->pt.y) / 2;
        p = new->next;
        i++;
    }
    poly->num = 2 * n;
    poly->num = 2 * n;
}

```



```
#include <stdlib.h>
```

```
void reset-polygon (Polygon *poly) ①
```

```
{
```

```
    int i=0;
```

```
    Item *p, *tmp;
```

```
    int n;
```

```
while (i < n)
```

```
    p = poly->head;
```

```
    n = poly->num;
```

```
    while (i < n)
```

```
    {
```

```
        tmp = p->next;
```

```
        free(p);
```

```
        p = tmp;
```

```
        i++;
```

```
    }
```

```
    poly->num = 0;
```

```
}
```

```
poly->head = NULL
```

```
    -2 > 0n
```

סמטסר אי תשס"ד  
מועד : ב' 04/04/2004  
משך הבחינה : ארבע שעות  
חומר עזר : כל חומר עזר כתוב למעט  
ספרים (או צילום של ספרים)

## בחינה בקורס C למהנדסים

מרצים : אייל אבן דר, רון ויין  
מתרגלים : נורית הספל, בן זנדבנק, דרור עירוני, אפי פוגל, ליעם רודיטי

### הנחיות כלליות

- במבחן 4 שאלות. יש להשיב על כל השאלות.
- משקל כל שאלה 25 נקודות.
- בשאלה מס' 1 יש לענות בהסבר קצר על כל סעיף. לכל השאלות האחרות נדרש כפתרון קוד (רצוי שעובד) ולא הסבר במילים.
- בשאלות 2-4 על הפתרון להיות יעיל ככל האפשר.
- את הפתרון יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- מותר (ואף רצוי) להשתמש בפונקציות ספרייה, אלא אם נאסר על כך במפורש.
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

#### לשימוש הבודקים בלבד :

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
	ציון

## שאלה מס' 1 (25 נקודות)

### סעיף א' (5 נקודות):

נתונה התכנית הבאה, המתרגמת ציונים בפורמט ה"אמריקאי" לציונים "ישראליים":

```
#include <stdio.h>
#include <ctype.h>

int main()
{
    char grade;
    printf ("Enter your grade in letters (A-F):\n");
    scanf ("%c", &grade);
    grade = toupper(grade);

    switch(grade)
    {
    case 'A':
        printf("Your grade is above 90\n");
    case 'B':
        printf("Your grade is above 80\n");
    case 'C':
        printf("Your grade is above 70\n");
    case 'D':
        printf("Your grade is above 60\n");
    case 'F':
        printf("Your grade is below 60\n");
    default:
        printf("Your grade is illegal!\n");
    }
    return 0;
}
```

מה הבעיה בתכנית זו ואיך ניתן לתקנה?

**פתרון:** חסר break אחרי כל case ולכן אם התוכנית תיתקל באחד ה-cases היא "תגלוש" ותבצע את כל ה-cases שאחריה. הפיתרון הוא להוסיף break אחרי כל שורת printf פרט ל-default.

**סעיף ב' (5 נקודות):**

נתונה התוכנית הבאה, המקבלת מחרוזת וסופרת כמה אותיות יש בה (תווים המייצגים אות לטינית) וכמה ספרות (התווים 0-9):

```
#include <ctype.h>

void count_chars (const char *s,
                  int n_letters, int n_digits)
{
    n_letters = 0;
    n_digits = 0;

    while (*s != '\0')
    {
        if (isalpha(*s))
            n_letters++;
        else if (isdigit(*s))
            n_digits++;

        s++;
    }
    return;
}

int main(void)
{
    char *s="If I go insane, please don't 117700";
    int l,d;
    count_chars(s,l,d);
    printf("There are %d letters and %d digits\n", l,d);
    return 0;
}
```

מה הבעיה בתוכנית זו ואיך ניתן לתקנה?

**פתרון:** אנו מעבירים את המשתנים l ו-d לפונקציה count\_chars ולא את הכתובות שלהם ולכן הפונקציה לא תשנה אותם ונקבל פלט חסר משמעות. על מנת לתקן, יש להעביר את הכתובות של l ו-d ולשנות את הפונקציה count\_chars כך שתקבל שני מצביעים ל-int וכך תוכל לשנותם.

**סעיף ג' (5 נקודות):**  
מה עושה הפונקציה הבאה:

```
void secret (char *s, char c)
{
    char    *dest;

    dest = s;
    while (*s)
    {
        if (*s != c)
        {
            *dest = *s;
            dest++;
        }
        s++;
    }
    *dest = '\0';

    return;
}
```

**פתרון:** הפונקציה מקבלת מחרוזת ותו, ומוחקת את כל מופעי התו מהמחרוזת.

**סעיף ה' (5 נקודות):**  
מה עושה הפונקציה הבאה:

```
int func (const int *arr, int n)
{
    int    a, b;

    if (n == 1)
        return (arr[0]);

    a = func (arr, n/2);
    b = func (arr + n/2, n - n/2);
    if (a > b)
        return a;
    else
        return b;
}
```

**פתרון:** פונקציה זו מקבלת מערך של מספרים שלמים ואת גודלו, ומחשבת את האיבר המקסימלי במערך בצורה רקורסיבית.

## שאלה מס' 2 (25 נקודות)

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
int is_legal_exp (const char *s);
```

הפונקציה מקבלת מחרוזת שאמורה לייצג ביטוי אריתמטי, ועליה לבדוק את חוקיות הביטוי. למעשה, עליה לבצע שתי בדיקות :

1. לוודא שהמחרוזת מכילה רק תווים חוקיים. התווים החוקיים הם כל אחד מבין: " 0123456789.+-\*/() ". כלומר: ספרות, נקודה עשרונית, ארבעת האופרטורים האריתמטיים (חיבור, חיסור, כפל וחילוק), סוגריים וכן התו רווח.
  2. המחרוזת מכילה קינון (nesting) חוקי של סוגריים.
- אם שני התנאים מתקיימים, הפונקציה מחזירה 1 (כלומר המחרוזת חוקית) ואחרת היא מחזירה 0.

לדוגמה, המחרוזות הבאות הן חוקיות :

```
"12.4", "3 +4.5", "(3 + 1.5)/2",  
"((3+4)/(2*(5.6-2)))"
```

המחרוזת הבאה איננה חוקית, כיוון שהיא מכילה תווים לא-חוקיים :

```
"34/y * (3 + 6.25)*x"
```

המחרוזות הבאות אינן חוקיות, משום שקינון הסוגריים בהן אינו חוקי :

```
"(7 + 9.34)*2.3 + 65.4/(12 + 6",  
"((1 + 2)*4 + (7 + 8))*3) + 12"
```

**פתרון:**

```

#include<stdio.h>
#include<string.h>

/* This function gets an expression containing      */
/* brackets and checks whether the nesting of      */
/* brackets is legal. Return 1 if yes, 0 if not. */
int isNested(const char *s)
{
    /* counting opening and closing brackets */
    int counter=0;
    /* We increment the counter by 1 when we see */
    /* an opening bracket and decrement by 1     */
    /* when we encounter a closing bracket.     */
    /* The counter should never be negative and it */
    /* should be 0 when we finish.              */
    while(*s) {
        if(*s == '(') counter ++;
        if(*s == ')') counter--;
            if(counter < 0) return 0;
        s++;
    }
    if(counter == 0) return 1;
    return 0;
}

int is_legal_exp(const char *s)
{
    const char *alnum = "0123456789.+-* /() ";
    if(!isNested(s)) return 0; /* First check nesting */
    /* If we're here, the expression is nested. */
    /* Check that all the chars are legal. */
    while(*s) {
        /* We check whether each character appears in */
        /* the string alnum. We can use strchr. Strchr */
        /* returns NULL if the char is NOT in the string. */
        if(!strchr(alnum,*s)) return 0;
        s++;
    }
    /* If we're here, all the characters are legal. */
    return 1;
}

```

## שאלה מס' 3 (25 נקודות)

### סעיף א' (15 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
int is_unique_array (const int *arr, int n, int m);
```

הפונקציה מקבלת מערך בשם `arr` של מספרים שלמים בגודל `n`, כאשר נתון שכל אחד מן האיברים במערך הוא מספר אי-שלילי החסום על-ידי `m`: כלומר כל איברי המערך הם מספרים שלמים בין 0 ל-`m-1`. על הפונקציה לבדוק האם כל איברי המערך שונים זה מזה: אם כן, היא מחזירה 1, ואחרת (כלומר יש לפחות זוג אחד של איברים במערך שיש להם אותו ערך) היא מחזירה 0.

כדי לקבל את מלוא הנקודות על הפונקציה שאתם כותבים לעבור רק פעם אחת על כל תא במערך `arr` (רמז – שימו לב לפרמטר `m` ומשמעותו, וחשבו כיצד ניתן לנצל אותו).

### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא :

```
int median_array (const int *arr, int n, int m);
```

על הפונקציה להחזיר את החציון של המערך, כלומר להחזיר מספר שחצי מאיברי המערך גדולים ממנו וחצי קטנים או שווים לו.

כדי לקבל את מלוא הנקודות על הפונקציה שאתם כותבים לעבור רק פעם אחת על כל תא במערך `arr` (רמז – השתמשו בטכניקה דומה לזו בה השתמשתם בסעיף הקודם).



פתרון סעיף א':

```
#include<stdio.h>
#include<stdlib.h>

/* We get an array of integers whose size is n. */
/* We know that the numbers in it are bounded by */
/* 0..m-1. We return 1 if no element appears more */
/* than once and 0 otherwise. */
int is_unique_array(int *arr, int n, int m)
{
    /* Use a frequency array. Since we know the range */
    /* of the elements is no more than 0..m-1, the of */
    /* the auxiliary array can be m. We allocate it */
    /* dynamically since m is a variable. */
    int *aux;
    int i; /* an index variable */

    /* allocate array. We will use calloc because we */
    /* want the array to be reset to 0. */
    aux = (int *)calloc(m, sizeof(int));
    if(aux == NULL) { /* verify allocation */
        printf("memory allocation failed\n");
        exit(1);
    }

    /* Go over the array and update the frequency array */
    for(i=0;i<n;i++)
        aux[arr[i]]++;

    /* Go over the frequency array and if a number */
    /* repeats more than once, return 0. */
    for(i=0;i<m;i++)
        if(aux[i] > 1) {
            free(aux);
            return 0;
        }

    /* If we're here, the array is unique. */
    free(aux);
    return 1;
}
```

פתרון סעיף ב' :

```
#include<stdio.h>
#include<stdlib.h>

/* We get an array of integers whose size is n. */
/* We know that the numbers in it are bounded by */
/* 0..m-1. We calculate the median of the array. */
int median(int *arr, int n, int m)
{
    int *aux;
    int i; /* an index variable */
    int count = 0; /* A counter to calculate the median */

    aux = (int *)calloc(m, sizeof(int));
    if(aux == NULL) { /* verify allocation */
        printf("memory allocation failed\n");
        exit(1);
    }

    /* Go over the array and update the frequency array */
    for(i=0;i<n;i++)
        aux[arr[i]]++;

    /* Go over the frequency array */
    /* and count the elements we encountered */
    for(i=0;i<m;i++) {
        count += aux[i];
        if(count >= n/2) break;
    }
    /* If we're here, the array is unique. */
    free(aux);
    return i;
}
```

## שאלה מס' 4 (25 נקודות)

נגדיר מבנה למימוש תור של מספרים בגודל 100. על תור זה לתמוך בפעולות הבאות: הוספת איבר לסוף התור, הוצאת איבר מראש התור (האיבר שיוצא הוא זה שהוכנס ראשון). את איברי התור ניתן לשמור במערך סטטי, כאשר יש לשמור בנוסף שני אינדקסים: `first` המציין את מיקומו של האיבר הראשון בתור, ו-`last` המציין את מיקומו של המקום הפנוי הראשון לאחר האיבר האחרון בתור.

לדוגמה, אם הכנסנו לתור ריק את האיברים 3, 52, 4, 56, 79 (בסדר זה, משמאל לימין), אז התור יראה כך:

0	1	2	3	4	5	6
3	52	4	56	79	?	?

98	99
?	?

(`first = 0, last = 5`)

...

98	99
?	?

...

כעת, אם נשלוף איבר מהתור, הוא יראה כך:

0	1	2	3	4	5	6
3	52	4	56	79	?	?

98	99
?	?

(`first = 1, last = 5`)

...

98	99
?	?

...

ואם נכניס את האיבר -1, התור יראה כך:

0	1	2	3	4	5	6
3	52	4	56	79	-1	?

98	99
?	?

(`first = 1, last = 6`)

...

98	99
?	?

...

### סעיף א' (5 נקודות):

הגדירו מבנה בשם `Queue` המייצג תור כפי שתארנו לעיל. את גודל התור (100 במקרה שלנו) הגדירו על-ידי פקודת `#define` מתאימה.

### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int enqueue (Queue *q, int val);
```

הפונקציה מקבלת תור ואיבר חדש בשם `val`. אם התור איננו מלא, על הפונקציה להוסיף את האיבר בסוף התור ולהחזיר 1, אחרת עליה להחזיר 0.

### סעיף ג' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int dequeue (Queue *q, int *p_val);
```

אם התור איננו ריק, על הפונקציה לתת לשלוף את האיבר הראשון מן התור ולשים את ערכו אל המשתנה ש-`p_val` מצביע עליו – במקרה זה, על הפונקציה להחזיר 1. אחרת, אם התור ריק על הפונקציה פשוט להחזיר 0 (ולא לשים שום ערך אל `p_val`).

פתרון כל השאלה:

```
#include<stdio.h>

#define QSIZE 100

typedef struct queue {
    int q[QSIZE];
    int first;
    int last;
} Queue;

/* Insert an element to the queue */
int enqueue(Queue *qu, int val)
{
    if(qu->last == QSIZE) /* Queue is full */
        return 1;
    /* Insert element at end of queue */
    qu->q[qu->last] = val;
    qu->last++;
    return 0;
}

int dequeue(Queue *qu, int *p_val)
{
    if(qu->last <= qu->first) /* Queue is empty */
        return 0;
    *p_val = qu->q[qu->first]; /* Put first value in queue */
    qu->first++;
    return 1;
}
```

סמיסטר : ב' תשס"ד  
מועד : ב' 20/08/2004  
משך הבחינה : שלוש שעות  
חומר עזר : כל חומר עזר  
כתוב למעט ספרים

## בחינה בקורס C למהנדסים

מרצים : אייל אבן דר, גיורא אונגר  
מתרגל : דורון חן

### הנחיות כלליות

- במבחן 4 שאלות.
- משקל כל שאלה 25 נקודות.
- על הפתרון להיות יעיל ככל האפשר.
- את הפתרון יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטיוטה בלבד.
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

לשימוש הבודקים בלבד :

	תעודת זהות
	מספר מחברת
	שאלה 1
	שאלה 2
	שאלה 3
	שאלה 4
	ציון

## שאלה מס' 1 (25 נקודות)

כתבו תוכנית הקולטת מהמשתמשת מספר חיובי,  $n$ , ולאחריו  $n$  מספרים. בשלב הראשון על התוכנית להדפיס את כל המספרים אשר מהווים **מקסימום** **מקומי**. מקסימום מקומי הוא מספר אשר גדול שווה מקודמו ומהמספר שבא אחריו.

בשלב השני על התוכנית להדפיס את כל המספרים אשר מהווים מקסימום מקומי מבין המספרים שהודפסו בשלב הראשון:  
בשלב השלישי יודפסו כל המספרים אשר מהווים מקסימום מקומי מבין המספרים שהודפסו בשלב הקודם. יש להמשיך בתהליך זה עד אשר ישאר מספר אחד בלבד, הלא הוא **המקסימום הגלובלי**.

הערה: יתכן שהמספר שהינו המקסימום הגלובלי יופיע יותר מפעם אחת ברשימת המספרים אשר נקלטו מהמשתמשת.

דוגמה:

קלט:

1 4 2 100 23 24 22 1 17 6 19

פלט:

4 100 24 17 19

100 19

100

### שאלה מס' 2 (25 נקודות)

מחרוזת באורך  $n$ , עבור  $n$  זוגי, תיקרא אנטי-פלינדרום אם התו ה- $i$  של המחרוזת והתו ה- $(n-i-1)$  הם בעלי קודי `ascii` הרחוקים זה מזה ב-5 לפחות. לדוגמה:

המחרוזת `dbcawuys` הינה אנטי-פלינדרום  
המחרוזת `yaqbdg` איננה אנטי-פלינדרום

הערות:

- מחרוזת באורך אי-זוגי בהכרח איננה אנטי-פלינדרום.
- מחרוזת ריקה איננה אנטי-פלינדרום.
- בשאלה זו אין להשתמש בפונקציות מהספרייה `string.h`

### סעיף א' (10 נקודות):

כתבו פונקציה בשם `IsAntiPalindrom`, המקבלת מחרוזת ומספר המכיל את אורכה (קרי מספר התווים במחרוזת). הפונקציה מחזירה 1 אם המחרוזת היא אנטי-פלינדרום ו-0 אחרת. עליכם לכתוב את הפונקציה מבלי להשתמש ברקורסיה.

### סעיף ב' (15 נקודות):

כתבו את אותה הפונקציה, הפעם ע"י שימוש ברקורסיה, מבלי להשתמש בלולאות.

### שאלה מס' 3 (25 נקודות)

ברשימת הטיסות הממריאות בשדה התעופה, מיוצגת טיסה ע":

- מספר סידורי (מספר שלם בן 4 ספרות)
- יעד הטיסה (מחרוזת)
- שמה של חברת התעופה (מחרוזת)
- שעת ההמראה, המורכבת משעות ודקות (ייצוג לפי בחירתכם)

הערות:

- ניתן להניח כי כל הטיסות ברשימה חוקיות, ובפרט שהשעה בתוכן חוקית.
- בהדפסות, חובה לשמור במדויק על הפורמט שיוזגם להלן. יש להדפיס את הטיסות לפי סדר הופעה ברשימה שניתנה, ואין למיין אותן לפי שדה כלשהו.
- בשאלה זו מותר להשתמש בפונקציות מהספרייה `string.h`.

### סעיף א' (5 נקודות):

הגדירו מבנה בשם `struct Departure` המייצג טיסה ממריאה.

### סעיף ב' (10 נקודות):

כתבו פונקציה המקבלת מחרוזת המכילה שם של חברת תעופה, וכן (בתוך מערך) רשימה של טיסות ממריאות. על הפונקציה להדפיס את כל הטיסות של חברה זו, אשר מופיעות ברשימה. כל טיסה צריכה להיות מודפסת בשורה אחת נפרדת (דוגמה להלן). אם ברשימה אין אף טיסה של החברה, על הפונקציה להדפיס הודעה מתאימה (דוגמה להלן).

דוגמה לפלט:

נניח ששם החברה שניתן לפונקציה הוא "El-AI". אזי, אם יש 3 טיסות של אל-על הפלט יהיה:

8827, El-AI, Rome, 13:24

3924, El-AI, Bangkok, 15:43

5672, El-AI, Paris, 9:12

לעומת זאת, אם ברשימה אין אף טיסה של אל-על, תודפס ההודעה הבאה:

Sorry, El-AI has no flights on the list

### סעיף ג' (10 נקודות):

שתי טיסות מוגדרות כזהות אך ורק אם כל השדות שלהן זהים. כתבו פונקציה המקבלת (בתוך מערך) רשימה של טיסות ממריאות, ובודקת האם יש ברשימה שתי טיסות זהות. על הפונקציה להדפיס "yes" ולהחזיר 1 אם יש ברשימה 2 טיסות זהות. אחרת (קרי, אם כל הטיסות שונות זו מזו), עליה להדפיס "no" ולהחזיר 0.



**שאלה מס' 4 (25 נקודות)**

שאלה זו עוסקת בקוד ל"הצפנה" של מספרים. יהי  $P$  "מערך פענוח" המתאר כיצד לפענח כל ספרה במספר אשר הוצפן על-פי הקוד. מערך הפענוח הינו בגודל 10 ומייצג תמורה (פרמוטציה) של הספרות 0 עד 9. דוגמה: נניח שבידינו  $P$ , מערך הפענוח הבא

1	5	3	8	0	7	2	6	4	9
$P[0]$	$P[1]$	$P[2]$	$P[3]$	$P[4]$	$P[5]$	$P[6]$	$P[7]$	$P[8]$	$P[9]$

$P$  מתאר, כאמור, איך לפענח כל ספרה במספר אשר הוצפן לפי הקוד. למשל, אם המספר לאחר שהוצפן הוא 83257, אזי הפענוח ייתן את המספר 48376. זאת, כיוון שהספרה 8 מיוצגת ע"י התא  $P[8]$  ולכן תפוענח להיות 4, הספרה 3 מיוצגת ע"י התא  $P[3]$  ולכן תפוענח להיות 8, וכך הלאה.

**סעיף א' (9 נקודות):**

כתבו פונקציה המקבלת את מערך הפענוח,  $P$ , ומספר לאחר שהוצפן. הפונקציה מחזירה מספר שלם המייצג את המספר המפוענח. אב-הטיפוס (prototype) של הפונקציה הוא:

```
int Decipher(int P[], unsigned int iCiphredNum);
```

רמז: מומלץ להשתמש ברקורסיה.

**סעיף ב' (9 נקודות):**

בהינתן מערך פענוח  $P$ , נסמן ב-  $invP$  את "מעריך הצפנה", המתאר כיצד להצפין כל ספרה. בהמשך לדוגמה לעיל, מעריך ההצפנה ייראה כך:

4	0	6	2	8	1	7	5	3	9
$invP[0]$	$invP[1]$	$invP[2]$	$invP[3]$	$invP[4]$	$invP[5]$	$invP[6]$	$invP[7]$	$invP[8]$	$invP[9]$

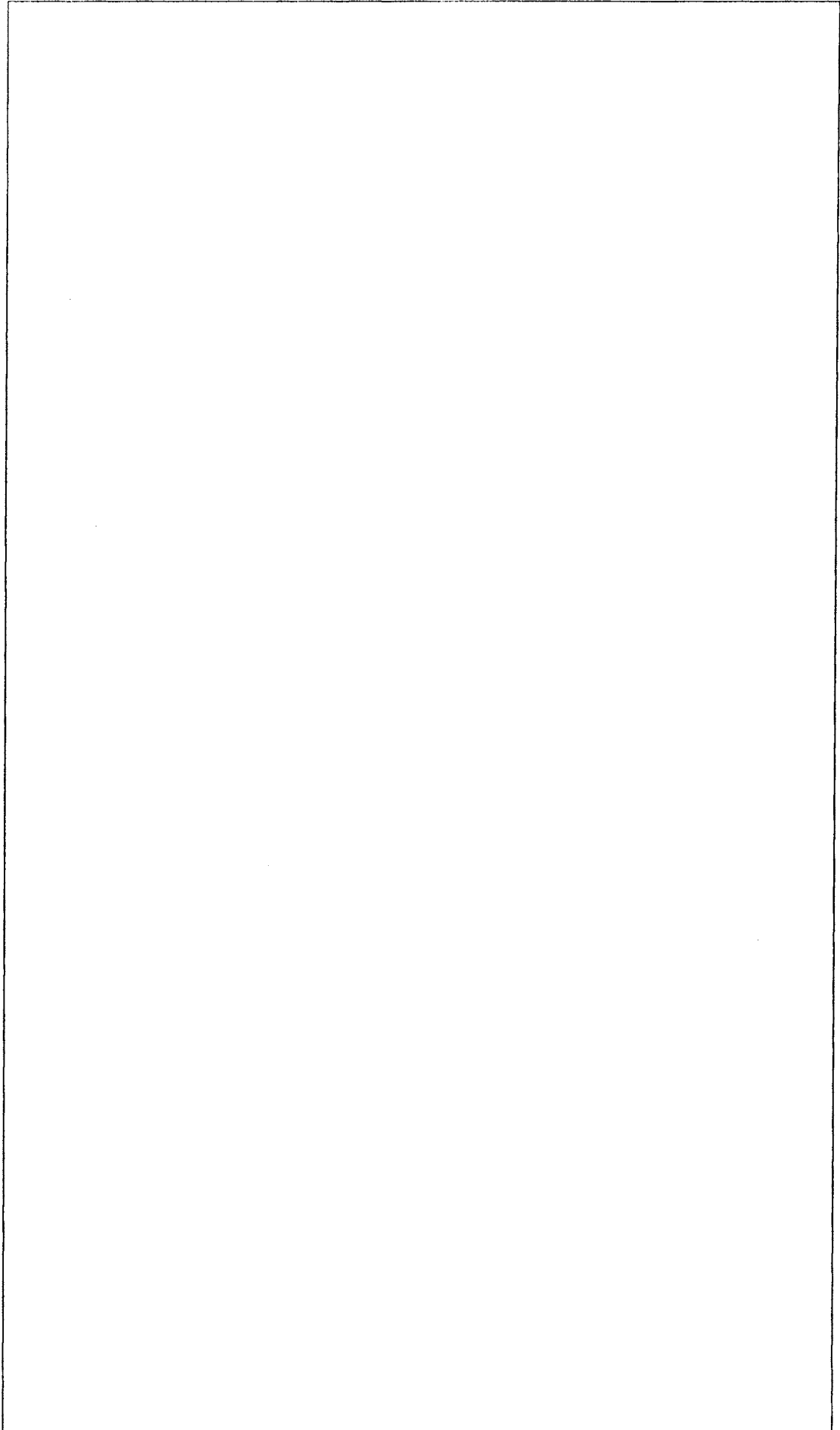
זהו מעריך ההצפנה, משום שאם 0 מפוענח להיות 1, הרי ש-1 מוצפן ל-0. אם 1 מפוענח להיות 5, הרי ש-5 מוצפן ל-1, וכך הלאה.

כתבו פונקציה המקבלת את  $P$ , מעריך הפענוח, ומחזירה את  $invP$ , "מעריך ההצפנה". רמז: קיים פתרון יעיל וקצר. שימו לב כי **אין צורך** למלא את ערכי מעריך הפענוח על פי הסדר.

**סעיף ג' (7 נקודות):**

כתבו פונקציה המקבלת את מעריך הפענוח,  $P$ , ושני מספרים אשר הוצפנו. על הפונקציה לבצע את הפעולות הבאות:

1. לפענח את שני המספרים בעזרת מעריך הפענוח.
  2. להחזיר את סכומם של שני המספרים המפוענחים, כשהוא מוצפן על-פי הקוד.
- ניתן להיעזר בפונקציות מהסעיפים הקודמים גם אם לא פתרתם אותם.





סמטסר: ב' תשס"ד  
מועד: א' 22/06/2004  
משך הבחינה: שלוש שעות  
חומר עזר: כל חומר עזר  
כתוב למעט ספרים

## בחינה בקורס C למהנדסים

מרצים: אייל אבן דר, גיורא אונגר  
מתרגל: דורון חן

### הנחיות כלליות

- במבחן 4 שאלות.
- משקל כל שאלה 25 נקודות.
- על הפתרון להיות יעיל ככל האפשר.
- את הפתרון יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כתיבה בלבד.
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

#### לשימוש הבודקים בלבד:

	תעודת זהות
	מספר מחברת
	שאלה 1
24	
23	שאלה 2
24	שאלה 3
24	שאלה 4
95	ציון

## שאלה מס' 1 (25 נקודות)

### סעיף א' (7 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int DoesContainDigit(int iNum, const int iDigit);
```

הפונקציה מקבלת מספר (iNum) וספרה (iDigit) ובודקת האם המספר כולל את הספרה. אם כן, הפונקציה מחזירה 1, אחרת 0. במקרה ש-iDigit איננו ספרה בודדת, הפונקציה מחזירה 0.

### סעיף ב' (9 נקודות):

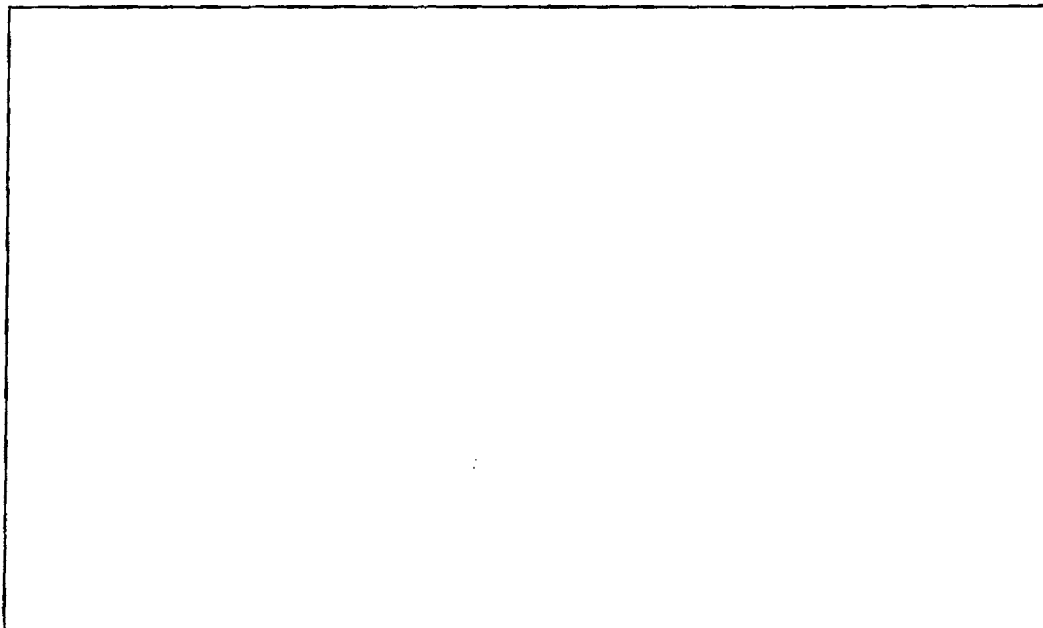
כתבו פונקציה לקליטת מספרים מהמשתמשת:

על הפונקציה לקלוט מהמשתמשת מספר שלם,  $h$ , ולאחר מכן לקלוט  $h$  ערכים שלמים. על הפונקציה לקבל פרמטר ולמלא בו את מספר הערכים שנקלטו ( $h$  לעיל). כמו כן, על הפונקציה להחזיר מצביע למקום שהוקצה בזיכרון עבור המספרים שנקלטו.

ניתן להניח כי  $h < 0$ , אך אין להניח כי הוא קטן מערך מסוים. כמו כן ניתן להניח כי הקלט תקין, קרי שהמשתמשת הקלידה  $h$  ערכים שלמים חוקיים.

### סעיף ג' (9 נקודות):

כתבו פונקציה `main()`, הקולטת מספרים מהמשתמשת באמצעות פונקציה נפרדת, כמוסבר בסעיף (ב). לאחר מכן, פונקציה `main()` מדפיסה את כל המספרים מביניהם אשר מכילים את הספרה 5 ו/או מתחלקים ב-5 ללא שארית, וכן לבסוף את סכומם של כל המספרים הללו. הערה: ניתן להשתמש בפונקציות מהסעיפים הקודמים גם אם לא פותרתם אותם.



```
int DoesContainDigit (int iNum, const int iDigit)
```

```
{ int tmp; int
```

```
while (iNum >= 10)
```

(k)

```
{ tmp = iNum % 10;
```

```
if (tmp == iDigit)
    return 1;
```

```
iNum = iNum / 10;
```

```
}
```

```
if (iNum == iDigit)
    return 1;
```

```
return 0;
/* else if (iDigit == 0) */
```

```
return 0;
```

```
}
```

/\* 501 -> 50  
LSB 001  
MSB 501  
501 -> 50  
MSB 501  
LSB 001  
\*/

/\* 501 -> 50  
MSB 501  
LSB 001  
\*/

6

2020/02/27 (2)

```
int* input_user(int *ip)
```

```
{  
    int n, i;
```

```
    int *p;
```

```
    printf("please enter a positive number\n");
```

```
    scanf("%d",&n);
```

```
    printf("please enter %d number:\n", n);  
    for(i=0; i<n; i++)  
        scanf("%d", &p[i]);
```

```
    p = (int*) malloc(n * sizeof(int));
```

```
    if(p == NULL)
```

```
    {  
        return printf("Memory out of Memory");  
        return NULL;
```

```
    }
```

```
    printf("please enter %d numbers\n", n);
```

```
    for(i=0; i<n; i++)
```

```
    {  
        scanf("%d", p+i);
```

```
    }
```

```
    *ip = n;
```

```
    return p;
```

```
}
```

9

```
int main ()
```

②

```
{ int *p;
```

```
int num; sum=0, status=0;
```

```
p = inputUser(&num);
```

```
if (p == NULL)
```

```
{ printf("out of memory");  
return 1;  
}
```

/\* p is a pointer  
of type int  
\*/

```
for (i=0; i < num; i++, status=0)
```

```
{ if (Does Contain Digit (*(p+i), 5))
```

```
{ printf("the number %d contain the digit 5\n", *(p+i));  
status = 1; }
```

```
if ((*(p+i)%5) == 0)
```

```
{ printf("the number %d divid. with 5\n", *(p+i));  
status = 1; }
```

~~status = 1; }~~

```
if (status)
```

```
sum += *(p+i);
```

/\* sum is a  
variable of  
type int  
\*/

```
printf("the sum of the numbers is %d\n", sum);
```

```
return 0;
```

7/17/25

שאלה מס' 2 (25 נקודות)

כתבו פונקציה בעלת המפרט (prototype) הבא:  
 int Convert (const char \*s, double \*p\_value);

הפונקציה תקבל מחרוזת ותוודא כי היא מהצורה הבאה:  
 "I lost \*NUMBER\* nis"  
 אם היא אכן מהצורה הנ"ל על הפונקציה להחזיר אחד ולמלא ב-p\_value את המספר שהופיע במחרוזת (קרי, הערך שסומן לעיל ע"י \*NUMBER\*).  
 אחרת על הפונקציה להחזיר 0. ניתן להניח כי המספר הוא חיובי.

דוגמא:

```
Convert("I lost 100.78 nis", &money);
```

תכניס לתוך money 100.78 ותחזיר 1. לעומת זאת

```
Convert("I lose 100.78 nis", &money);
```

```
Convert("I lost 7fd6 nis", &money);
```

וכמו כן

יחזירו 0, כי המחרוזות שהועברו אינן עומדות בדרישות.

מחזיר 0 כי לא נמצא מספר  
 #include <string.h>  
 #include <ctype.h>

```
int Convert (const char *s, double *p_value)
{
    char S1[8], S2[5];
    double dmoney, sheckl, agorot, dj;
    int  ick=0, ij, index_s, index_fin, from;

    for (i=0; i<7 && S[i]!='\0'; i++)
    {
        S1[i]=S[i];
    }
    S1[i]='\0';
}
```

/\* המספר של מילת מס  
 /\* - nis - S2, Ilost - S1  
 /\* נחזיר כאלו דגול נכון  
 ונסים במק S1  
 /\* Ilost מקום



מס' מחברו strcmp של המילים /

False בלתי נכונה /

```
if (strcmp(s1, "I_lost_"))
```

at I קצת /  
/ \* שלים /

```
return 0;
```

```
index_s = i;
```

/ \*

המחרון המחרון /

```
while (for (isdigit(s[i]) && s[i] != '\0'))  
    i++;
```

```
if (s[i] == '\0')  
    return 0;
```

```
index_fin = i - 1;
```

/ \* המחרון המחרון /

```
for (j = 1; index_s <= index_fin; index_fin--)
```

```
{  
    check += (s[index_fin] - '0') * j;  
    j *= 10;  
}
```

```
if (s[i] == '0')
```

```
    if (! (isdigit(s[i+1]))  
        return 0;
```

/ \* 105. Found by myself \* /

```
    i++;
```

/ \* המחרון המחרון /

```
for (dj = 0.1; isdigit(s[i]) && s[i] != '\0'; i++)
```

```
{  
    aprot += (s[i] - '0') * dj;
```

```
    dj *= 0.1;
```

/ \* המחרון המחרון /



שאלה מס' 3 (25 נקודות)

ה- prototype הבא הינו של פונקציה המקבלת שתי מחרוזות באורך שווה ובודקת האם הן זהות.

```
int AreStringsIdentical( const char* str1, const char* str2 );
```

ניתן להניח כי המחרוזות חוקיות. על הפונקציה להחזיר 1 אם הן זהות ו-0 אחרת.

סעיף א' (12 נקודות):

ממשו את הפונקציה מבלי להשתמש ברקורסיה.

סעיף ב' (13 נקודות):

ממשו את הפונקציה באמצעות רקורסיה.

```

int AreStringsIdentical (const char* str1, const char* str2)
{
    int i=0;
    for (; str1[i] == str2[i] && str1[i] != '\0'; i++)
    {
        if (str1[i] == '\0')
            return 1;
    }
    // else
        return 0;
}

```

*Handwritten notes:*

- \*/ נכון כי התנאי מסתובב סוגר
- \*/ כי התנאי שווים
- \*/ כי התנאי אמת
- 11 / 12

```

int AreStringsIdentical(const char* str1, const char* str2)
{
    if (*str1 == '\0')           /* אם הסוף של str1
        return 1;                * אז הם זהים
    if (*str1 != *str2)         /* אם האות הראשונה שונה
        { return 0; }           * אז הם לא זהים
    else
        { return(AreStringsIdentical(str1+1, str2+1)); }
}

```



$\frac{13}{13}$

שאלה מס' 4 (25 נקודות)

סעיף א' (7 נקודות):

הגדירו מבנים (structures) לייצוג נקודה וישר במישור.

סעיף ב' (9 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int Intersect(const line_t *line1, const line_t *line2, point_t *point);
```

הפונקציה מקבלת שני ישרים ומצביע לנקודה, מחזירה 1 אם הם נחתכים ו-0 אחרת, וממלאת את המצביע לנקודה בנקודת החיתוך של הישרים.

סעיף ג' (9 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int PrintIntersections(const line_t *arr_line, int size);
```

על הפונקציה להדפיס את כל נקודות החיתוך של הישרים, ולהחזיר את מספר נקודות החיתוך.

אם שלושת הקווים הם:  $y = 2x$ ,  $y = x + 5$ ,  $y = x$  : על הפלט להראות כך:

The lines  $y = 2x + 0$  and  $y = x + 5$  intersects in (5, 10)

The lines  $y = 2x + 0$  and  $y = x + 0$  intersects in (0, 0)

```
typedef struct point {
    double x;
    double y;
} point_t;

typedef struct line {
    double a;
    double b;
} line_t;
```

(4

7

12

int intersect (const line t \* line1, const line2, line2, point\_t \* point)

{

if (line1->a == line2->a)

if (line1->b == line2->b)

{ printf("they are the same\n");

return 0;

}

return 0;

}

point->x = (line2->b - line1->b) / (line1->a - line2->a);

point->y = (point->x \* line1->a + line1->b);

return 1;

}

אם שני קווים זהים  
אם שני קווים מקבילים  
/ \* > / \*  
אם שני קווים  
אם שני קווים

אם שני קווים זהים

/ \*  
אם שני קווים זהים

$$x = \frac{b_2 - b_1}{a_1 - a_2}$$

y → a<sub>1</sub>x + b<sub>1</sub>

a<sub>2</sub>x + b<sub>2</sub>

a<sub>1</sub>x + b<sub>1</sub>

אם שני קווים זהים

\*

```
int printIntersections (const line_t *arr_line, int size)
```

```
{ point_t p1;
```

```
int i, j, check = 0;
```

```
for (i=0; i < size; i++)
```

```
for (j=i+1; j < size; j++)
```

```
{ if (intersect (arr_line [i], arr_line [j], p1))
```

```
printf ("the line y = %gX + %g and y = %gX + %g intersect in (%g, %g)\n",
```

```
arr_line [i].a, arr_line [i].b, arr_line [j].a, arr_line [j].b, p1.x, p1.y);
```

```
check ++;
```

```
}
```

```
/* %g is given  
so no printf  
/* sooo b
```

```
printf ("the number of intersect points is %d\n", check);
```

```
return check;
```

```
}
```

סמטסר א' תשס"ד  
מועד: א' 17/02/2004  
משך הבחינה: שלוש וחצי שעות  
חומר עזר: כל חומר עזר כתוב למעט  
ספרים (או צילום של ספרים)

## בחינה בקורס C למהנדסים

**מרצים:** אייל אבן דר, רוני ויין  
**מתרגלים:** נורית הספל, בן זנדבנק, דרור עירוני, אפי פוגל, ליעם רודיטי

### הנחיות כלליות

- במבחן 5 שאלות. יש לפתור את שאלות 1-3 ולבחור שאלה אחת מבין 4,5 (לתלמידי הנדסת חשמל מומלץ לענות על שאלה מס' 5 ולתלמידי הנדסת מכונות, הנדסת תעשייה וניהול והנדסה ביו-רפואית מומלץ לענות על שאלה מס' 4).
- משקל כל שאלה 25 נקודות.
- בשאלה מס' 1 יש לענות בהסבר קצר על כל סעיף. לכל השאלות האחרות נדרש כפתרון קוד (רצוי שעובד) ולא הסבר במילים.
- בשאלות 2-5 על הפתרון להיות יעיל ככל האפשר.
- את הפתרון יש לכתוב במסגרות המסומנות לכל שאלה. במחברת הבחינה יש להשתמש כטייטה בלבד.
- ניתן להשתמש בפונקציות ספריה (בעיקר פונקציות קלט/פלט – זכרו לבצע `#include` מתאים) פרט למקומות בהם הדבר נאסר.
- ✖ זכרו לשחרר משאבים (זכרון או קבצים) לאחר סיום השימוש בהם!
- נא לכתוב בכתב קריא ולא מחובר.
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו.
- ✓ בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון.

### לשימוש הבודקים בלבד:

040792916	תעודת זהות
46	מספר מחברת
25	שאלה 1
25	שאלה 2
25	שאלה 3
25	שאלה 4/5
100	ציון



## שאלה מס' 1 (25 נקודות)

בשני הסעיפים הראשונים של השאלה אנו מתייחסים למבנה (structure) שהוגדר באופן הבא:

```
struct _Complex
{
    float    re; /* The real part. */
    float    im; /* The imaginary part. */
};
typedef struct _Complex Complex;
```

כמו-כן נתונה לנו הפונקציה:

```
void complex_set (Complex *p_z, float x, float y)
{
    p_z->re = x;
    p_z->im = y;
    return;
}
```

### סעיף א' (5 נקודות):

נתונה לנו התכנית הבאה:

```
int main ()
{
    Complex    *p_z;

    complex_set (p_z, 3, 4);

    /* The rest of the program comes here ... */

    return (0);
}
```

מה הבעיה שיש בתכנית (הבעיה עלולה לגרום ל"תעופה" של התכנית בזמן ריצה)?

בפונקציה `complex_set` מניחה כי  $p_z$  (משתנה) היא מערך של `Complex` (מקום טיפוס) וקצב נכונה. (מקום בסיסון) איר

מחזיק (מקום וז"ן) למעשה

אכן גודל שטוח לפונקציה `complex_set` נדרש למעשה

למטה וגודלה שצדד מקום למי יניח אכן נכונה

סגור יתור למקום "גודל" של גודל.



סעיף ג' (5 נקודות):

מה עושה הפונקציה הבאה:

```
void secret (const char* s1, char *s2)
{
    while (*s1) → while (*s1 != '\0') ← נגמל לזאק ←
    {
        *s2 = *s1;
        s1++;
        s2++;
    }
    *s2 = *s1; → *s2 = '\0';
}
```

בני מנכרד הדוגר. de מוכן מחוסר s1  
סס מחוסר s2 סס וכו' סס איגו ג' ו' ✓

סעיף ד' (5 נקודות):

נתונה לנו התכנית הבאה, שאמורה לקרוא מחרוזת מן המשתמש תו אחר תו עד שהמשתמש לוחץ על Enter תוך הגבלת אורך המחרוזת ל- 80 תווים לכל היותר, ואז להדפיס את המחרוזת על המסך.

```
#define MAX_STR 80

int main ()
{
    char    str[MAX_STR + 1]; /* The string. */
    int    l_str = 0;        /* Its length. */
    int    c;

    while ((c = getchar()) != '\n')
    {
        if (l_str == MAX_STR)
        {
            printf ("Error - too long.\n");
            return (1);
        }
        str[l_str] = c;
        l_str++;
    }
    (*)
    printf ("%s\n", str);
    return (0);
}
```

קראו את הסיפור

התכנית אכן מדפיסה את המחזורות על המסך, אולם בסוף המחזורות מופיעים מספר תווי "זבל" שהשתמש לא הכניס כקלט. מדוע? מה צריך להוסיף כדי שהתכנית תפעל כנדרש?

קוד למימוש כ המחזורות במקביל בסוף str  
 null-terminated, וכן גם printf היא יוצרת את  
 הסכום של כל האותיות בסוף str  
 כן למימוש של מספר האותיות (\*) את האותיות  
 str[2-str] = '\0';

סעיף ה' (5 נקודות):  
 מה עושה הפונקציה הבאה:

```
void my_func (double *vec, int n)
{
    double    min_val;
    int       i_min, i;

    if (n == 1)
        return;

    min_val = vec[0];
    i_min = 0;
    for (i = 1; i < n; i++)
    {
        if (vec[i] < min_val)
        {
            min_val = vec[i];
            i_min = i;
        }
    }

    vec[i_min] = vec[0];
    vec[0] = min_val;

    my_func (vec + 1, n - 1);
    return;
}
```

היא מקבלת מערך של מספרים (בין 1 ל-10) ומחזרת את המערך מחדש  
 כך שכל מספר יופיע במקומו הראשון במערך ומספרים אחרים יופיעו  
 במקומות אחרים.  
 (במקרה של מספרים זהים במערך יופיעו מספרים אחרים במקומות אחרים)  
 למימוש יוניץ איגור מניח, ואת הרישום של המערך יחזיר -1

## שאלה מס' 2 (25 נקודות)

שימו לב: בשאלה זו אין להשתמש בפונקציות ספריה סטנדרטיות.

### סעיף א' (12 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int convert (const char *s, int *p_value);
```

הפונקציה מחשבת את הערך המספרי שמייצגת מחרוזת הקלט s ומחזירה אותו דרך המשתנה s-p\_value מצביע עליו. על הפונקציה להחזיר 1 במקרה שמחרוזת הקלט מייצגת מספר שלם (למשל "1634" או "-834") ו-0 בכל מקרה אחר (למשל "ab23jd" או "14.67").

הערה: שימו לב שהתו הראשון במחרוזת יכול להיות '-' ואז היא מייצגת מספר שלילי. שאר התווים חייבים להיות ספרות. אתם רשאים להניח שמחרוזת הקלט איננה ריקה, וכן שהמספר המיוצג על-ידי המחרוזת איננו גדול מדי בערכו המוחלט (כלומר הוא ניתן לייצוג על-ידי משתנה מסוג int).

### סעיף ב' (13 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int calc_sum (const char *strs[], int n);
```

או לחילופין (עבור MAX\_LEN מספר כלשהו):

```
int calc_sum (const char strs[][MAX_LEN], int n);
```

הפונקציה מקבלת מערך של n מחרוזות בשם strs והיא מחשבת (ומחזירה) את סכום המספרים שמחרוזות אלה מייצגות, תוך התעלמות מן המחרוזות שאינן מייצגות מספר שלם.

```
int convert(const char *s, int *p_value) (10)
{
    int negative_option = 1;
    int i = 0, sum = 0;

    if (*s == '-') {
        negative_option = -1; // check for negative
        i++;
    }

    for (; s[i] != '\0'; i++) {
        if (s[i] <= '9' && s[i] >= '0') { // input valid?
            sum = sum * 10 + (s[i] - '0'); // input valid
        }
        else
            return 0; // input was not valid
    }
}
```

Page 142  
6/91

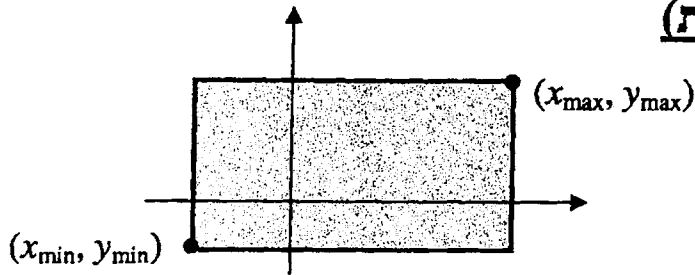
```

    *p-value = negative_option * sum;
    return 1; // declare success.
}

int calc_sum(const char strs[][Max_LEN], int n) (2)
{
    int sum = 0, i, temp;
    for (i = 0; i < n; i++) {
        if (convert(strs[i], &temp)) // if valid..
            sum += temp;
    }
    return sum;
}

```

### שאלה מס' 3 (25 נקודות)



#### סעיף א' (5 נקודות):

הגדירו שני מבנים (structures): האחד, בשם Point אמור לייצג נקודה במישור בעלת שתי קואורדינטות ממשיות. השני, בשם Rectangle מייצג מלבן המקביל לצירים – המאופיין ע"י הנקודה השמאלית-התחתונה שלו וע"י הנקודה הימנית-העליונה שלו (ראו ציור).

#### סעיף ב' (10 נקודות):

כתבו פונקציה בעלת המפרט (prototype) הבא:

```
int is_in_rect (const Rectangle *rect,  
               const Point *p);
```

הפונקציה מחזירה 1 אם הנקודה p מוכלת בתוך המלבן rect ו-0 אם היא אינה מוכלת בו.

#### סעיף ג' (10 נקודות):

כתבו תכנית הקוראת מן המשתמש, לפי הסדר:

- מספר שלם n.
  - n מלבנים (כל מלבן מאופיין על-ידי ארבעה מספרים ממשיים המייצגים את הקואורדינטות של פינות המלבן, כמתואר בסעיף א').
  - נקודה במישור p (הנקודה נתונה ע"י שתי קואורדינטות).
- על התכנית לאחסן את המלבנים שהיא קוראת במערך שהיא מקצה באופן דינאמי. על התכנית לספור כמה מבין n המלבנים מכילים את הנקודה p ולהדפיס מספר זה ככלט.

הערה: אתם יכולים להניח כי הקלט שהמשתמש מכניס הוא תקין.

```
typedef struct Point {  
    real x; float x;  
    real y; float y;  
} Point;
```

```
typedef struct Rectangle {  
    point p_min;  
    point p_max;  
} Rectangle;
```

```

int is_in_rect(const Rectangle *rect,
               const point *p)
{
    int is_in = 1; // assume point is in rectangle.
    if (p->x < rect->p_min.x || p->x > rect->p_max.x)
        is_in = 0;
    if (p->y < rect->p_min.y || p->y > rect->p_max.y)
        is_in = 0;
    return is_in;
}

```

✓  
 ✓

```

#include <stdio.h>
#include <stdlib.h>

```

```

int main() {
    Rectangle *rect;
    point p;
    int i, counter = 0;

    printf("enter a positive number (n)\n");
    scanf("%d", &n);
    /* allocate memory */
    rect = (Rectangle *) malloc(n * sizeof(Rectangle));
    if (rect == NULL) {
        printf("out of memory\n");
        return -1;
    }
}

```





```
/* input from user n rectangles */
```

```
for (i=0; i<n; i++) {
```

```
    printf("please enter the minimal x and y of rect\n");
```

```
    scanf("%f%f", &rect[i].p_min.x, &rect[i].p_min.y);
```

```
    printf("please enter the maximal x and y of rect\n");
```

```
    scanf("%f%f", &rect[i].p_max.x, &rect[i].p_max.y);
```

```
    printf("\n");
```

```
}
```

```
/* get a dot from user */
```

```
printf("insert a point (x and y)\n");
```

```
scanf("%f%f", &p.x, &p.y);
```

```
/* count how many rect's consume point */
```

```
for (i=0; i<n; i++)
```

```
    counter += is_in_rect(&rect[i], &p);
```

```
printf("point is in %d rectangles\n", counter);
```

```
free(rect);
```

```
return 0;
```

```
}
```



10/2/2017

שאלה מס' 4 (25 נקודות)

כתבו תכנית המקבלת בשורת הפקודה שלה (command line) את שני הארגומנטים הבאים:

1. שם של קובץ קלט.
  2. מחרוזת של תווים.
- כפלט, על התכנית להציג על המסך את תכנו של קובץ הקלט, לאחר שהסירה ממנו את כל התווים המופיעים במחרוזת הנתונה.

הערה: עליכם לבדוק שאתם מקבלים מספיק ארגומנטים בשורת הפקודה, וכן עליכם לוודא שאתם מצליחים לפתוח את הקובץ (אחרת עליכם להדפיס הודעת שגיאה ולסיים את התכנית).

לדוגמה, אם לתכנית שלנו קוראים del\_chars ומריצים אותה ע"י שורת הפקודה הבאה:

```
del_chars creep.txt Ie.m
```

בהנחה שקובץ הקלט creep.txt נראה כך:

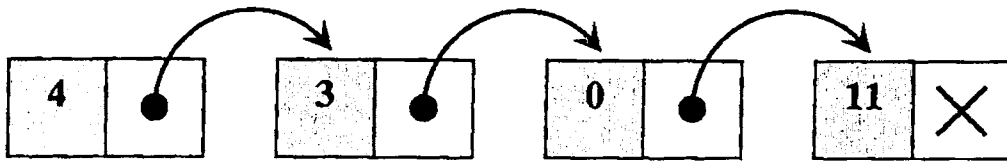
```
But I'm a creep, I'm a weirdo.  
What the hell am I doing here?  
I don't belong here.
```

אז הפלט על המסך יהיה (לאחר שמחקנו את כל המופעים של התווים I e . m):

```
But ' a crp, ' a wirdo  
What th hll a doing hr?  
don't blong hr
```



שאלה 5 (25 נקודות)



סעיף א' (5 נקודות):

הגדירו מבנה (structure) אשר ישמש לייצוג רשימה מקושרת אשר תכיל מספרים שלמים (כמו בציור).

סעיף ב' (10 נקודות):

כתבו פונקציה המקבלת מצביע לראש הרשימה ומחזירה את סכום המספרים ברשימה.

סעיף ג' (10 נקודות):

כתבו פונקציה המקבלת מצביע לראש הרשימה, אשר תשכפל את הרשימה ותחזיר מצביע לתחילת הרשימה המשוכפלת.

```
typedef struct mylist {  
    int n;  
    struct mylist *next;  
} mylist;
```

```
int listsum(mylist *Lhead) {  
    if (Lhead == NULL)  
        return 0;  
    // else it will do:  
    return Lhead->n + listsum(Lhead->next);  
}
```

```

        2. 2017) 23/10/2017 20:00 10 02 (d)
mylist *copylist(mylist *Lhead) {
    mylist *new_head;

    if (Lhead == NULL) // end of list?
                        // (or list empty)
        return NULL;

    new_head = (mylist *) malloc(sizeof(mylist));

    /* check for correct allocation */
    if (new_head == NULL) {
        printf("out of memory!\n");
        return NULL;
    }

    /* copy Lhead item */
    new_head->n = Lhead->n;
    new_head->next = copylist(Lhead->next);

    return new_head;
}

```

```

...
#include <stdlib.h>
... "malloc"

```

סמטסר א' תשס"ג  
 מועד: א' 04/02/2003  
 משך הבחינה: שלוש וחצי שעות  
 חומר עזר: דף אחד בכתב יד

## בחינה בקורס C למהנדסים

מרצים: אייל אבן דר, ליעם רודיטי  
 מתרגלים: אולגה סורקין, אתי עזרא, ד"ר גדי קימל.

### הנחיות כלליות

- במבחן 4 שאלות
- משקל כל שאלה 25 נקודות
- לכל השאלות נדרש כפתרון קוד (רצוי שעובד) ולא הסבר במילים
- על הפתרון להיות יעיל ככל האפשר
- את הפתרון יש לכתוב במסגרות המסומנות לכל שאלה
- אין להשתמש בפונקציות ספריה אלא אם הותר במפורש למעט בפונקציות קלט פלט
- נא לכתוב בכתב קריא ולא מחובר
- מומלץ לקרוא היטב את השאלון כולו בטרם תתחילו לענות עליו
- בדקו בסיום הבחינה כי עניתם על כל השאלות וכל הסעיפים וכן כי מילאתם את מספר תעודת הזהות ואת מספר המחברת בכל עמוד של השאלון

תעודת זהות	מספר מחברת
46	
20	שאלה 1
20	שאלה 2
21	שאלה 3
10	שאלה 4
	ציון

## שאלה 1 (25 נקודות)

כתבו תוכנית המקבלת כקלט שלושה מספרים שלמים. המספר הראשון יתאר את השעה של הזמן הנוכחי (hour), המספר השני יתאר את הדקה של הזמן הנוכחי (minute) והמספר השלישי יתאר מספר דקות (offset) שיש לחבר לזמן הנוכחי. התוכנית תדפיס את השעה והדקה לאחר חיבור offset דקות לזמן הנוכחי. שימו לב ש offset יכול להיות יותר גדול מ-60. ניתן להניח כי הקלט חוקי (כלומר hour בין 0 ל 23 ו minute בין 0 ל 59)

דוגמא:

קלט: 11 3 145

פלט: 13 28

(אם נחבר לשעה 11:03 דקות נקבל את השעה 13:28)

קלט: 23 52 9

פלט: 0 1

```
#include <stdio.h>
int main()
{
    int hour=0, minute=0, offset=0;
    int newhour=0, new_min=0, temp_min=0, temp_h=0;
    printf("Enter hour, minutes of current time and number of
    hours to add\n");
    scanf("%d %d %d", &hour, &minute, &offset);
    if (offset < 60)
    {
        if ((offset + minute) >= 60)
        {
            new_hour = hour + 1;
            new_min = offset + minute - 60;
        }
        else
        {
            new_hour = hour;
            new_min = offset + minute;
        }
    }
}
```

'kel' on 'm' ←  
 / 'new'  
 ' if (offset >= 60)

```

else
{
temp-h = offset / 60;
temp-m = offset / 60 offset / 60;
temp-h = offset / 60;
temp-m = offset % 60;
if ((temp-m + minutes) >= 60)
{
new-hour = hour + temp-h + 1;
new-min = temp-m + minutes - 60;
}
else
{
new-hour = temp-h + hour;
new-min = temp-m + minutes;
}
}
if (new-hour == 24) {
new-hour = 0;
}
printf("New time is %d %d\n", new-hour, new-min);
return 0;
}

```

for example, if  
 new-hour = 24  
 new-hour = 23

(-5)

rand() time() srand() I לא תוכלו להגדיר

**שאלה 2 (25 נקודות)**

כתבו תוכנית אשר מאתחלת מערך של int-ים בגודל 100 בעזרת הפונקציות rand(), srand(), time() למספרים בטווח 1-50. על התוכנית להדפיס את המערך. בנוסף על התוכנית להדפיס את המספר השכיח ביותר (המספר שמופיע הכי הרבה פעמים) ואת כל האיברים במערך הגדולים ממנו. במקרה של מספר איברים שכיחים ביותר יש לבחור אחד מהם באופן שרירותי. השימוש ב-srand() יהיה: srand((unsigned)time(NULL))

\* קבועי אקראיים בין 0-32,767 rand() מנימה int

81  
RAND  
-2

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

srand((unsigned)time(NULL));
int main()
{
    int arr[100], count[51] = {0};
    int i, j, temp = 0, arr_max = 0, max_freq = 0;
    for (i=0; i<100; i++) /* for */
        arr[i] = (rand() % 100) - 17;
    for (i=0; i<100; i++) /* for */
        printf("%d ", arr[i]);
    printf("\n");
    for (i=0; i<100; i++)
        count[arr[i]]++;
    for (i=0; i<51; i++) /* arr_max */
    {
        if (count[i] > max_freq)
        {
            max_freq = count[i];
            arr_max = i;
        }
    }
    printf("The most frequent number is %d\n", arr_max);
}
```





### שאלה 3 (25 נקודות)

#### סעיף א (10 נקודות):

כתבו את הפונקציה `char *strstr(char *s1, char *s2)` אשר מחזירה את כתובת המופע הראשון ב-`s1` של `s2` אם קיים כזה ו-`NULL` אחרת. בסעיף זה ניתן להשתמש ב-`strlen(cosnt char *)`.

#### סעיף ב (15 נקודות):

כתבו את הפונקציה `delete_word` המקבלת שתי מחרוזות `str` ו-`substr` ומוחקת את המופע הראשון של `substr` מ-`str`. אסור לפונקציה לשנות את המחרוזות הקיימות אלא להקצות מקום למחרוזת חדשה באמצעות `malloc` ולהחזירה. אם אין ב-`str` מופע של `substr` אז על הפונקציה להחזיר עותק של `str`. ניתן להשתמש בפונקציה מסעיף א, גם אם לא עניתם על סעיף זה.

דוגמא:

`str="abcdef"`

`substr="cd"`

המחרוזת שתוחזר מהפונקציה היא:

`"abef"`

```
(k) char *strstr (char *s1, char *s2)
{
    int length1 = strlen (s1), length2 = strlen (s2);
    int cut = 0;
    char *place = NULL;
    int i, j;
    for (i=0; i < length1; i++)
    {
        if (*s1 == *s2)
        {
            place = s1;
            for (j=0; j < length2; j++)
            {
                if (*s1 == *s2)
                    cnt++;
                s1++;
                s2++;
            }
            if (cut == length2)
                return place; → else
        }
    }
}
```

כאן, כאן  
s1 = aaab  
s2 = aab  
↑

3

←

s1 = place + 1;

if - 1 onim ←  
 for null ←  
 for null ←  
 if null ←  
 place be ←  
 NULL null, for

```

else
    s1++;
}
return place; ←
NULL place →
}
② char * delete_word (char *str, char * substr)
{
    char *place, *new;
    int i;
    place = strstr (str, substr);
    if (place == NULL)
    {
        new = (char *) malloc (strlen(str) * sizeof(char));
        for (i=0; i < strlen(str); i++)
        {
            *new = *str;
            new++;
            str++;
        }
    }
    else
    {
        new = (char *) malloc ((strlen(str) - strlen(substr)) * sizeof(char));
        while (*str != *place)
        {
            *new = *str;
            new++;
            str++;
        }
    }
}

```

if (new == NULL) return NULL;

if (new == NULL) return NULL;

(-1)



## שאלה 4 (25 נקודות)

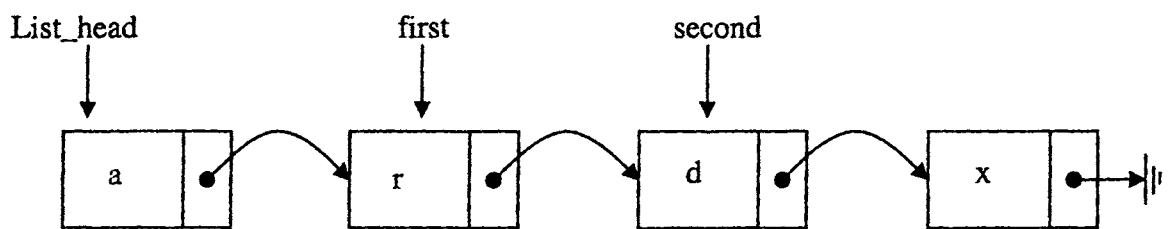
**סעיף א (3 נקודות):** הגדירו מבנה אשר ישמש ליצירת רשימה מקושרת המכילה char ומצביע לאיבר הבא ברשימה המקושרת.

**סעיף ב (10 נקודות):** כתבו פונקציה המקבלת char ומצביע לראש הרשימה ובודקת האם ה-char נמצא ברשימה. אם האיבר אכן ברשימה יש להחזיר את מיקומו הראשון ברשימה (int), אחרת יש להחזיר -1.

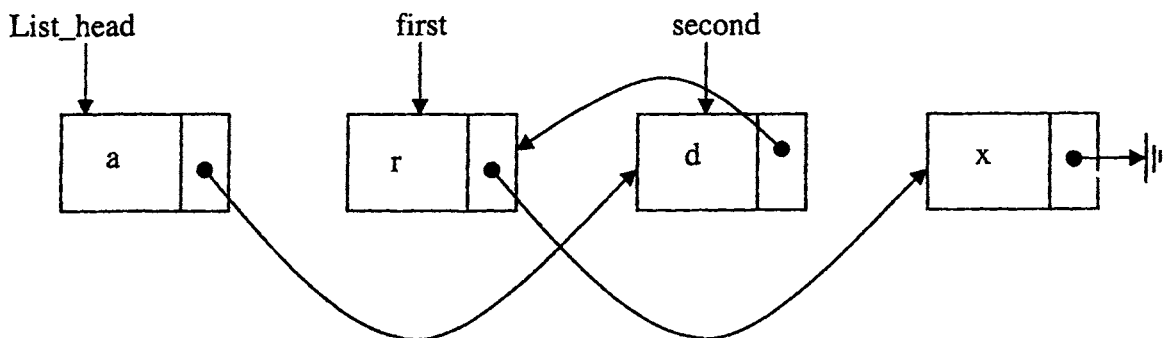
**סעיף ג (12 נקודות):** כתבו פונקציה בשם Swap אשר מקבלת שלושה מצביעים ומחזירה מצביע אחד.

`Item* Swap(Item* list_head, Item* first, Item* second)`

מצביע אחד הוא לראש רשימה של Items. ושני מצביעים לאיברים כלשהם ברשימה. הפונקציה תחזיר מצביע לראש הרשימה. ניתן להניח ששני המצביעים הנוספים אינם מצביעים לראש הרשימה. על הפונקציה להחליף בין שני האיברים הללו. אין לשנות את ה-char הנמצא בכל איבר ברשימה אלא לשנות את המצביעים. לדוגמא קלט לפונקציה:



הפונקציה תחזיר מצביע לראש הרשימה הבאה:



④ typedef struct Item

(k)

✓

```
{
  char ch;
  struct Item *next;
} Item;
```

(f) int where\_in\_list (char ch1, Item \*head)

```
{
  int cnt = 0;
  make
  Item *tail;
  tail = head;
  while (tail -> next != NULL)
  {
    cnt++;
    if (tail -> ch == ch1)
      return cnt;
    else
      tail = tail -> next;
  }
  if (tail -> next == NULL)
    return -1;
}
```

if tail  
is NULL  
return -1  
if tail -> ch == ch1  
return cnt

3

return -1  
return cnt

②

Item \*Swap (Item \*list-head, Item \*first, Item \*second)

```
{
    int place-first, place-second, i;
    Item *head1, *head2, *temp;

    temp = NULL;
    head1 = list-head;
    head2 = list-head;
    place-first = where-in-list (first->ch, list-head);
    place-second = where-in-list (second->ch, list-head);
    for (i=0, i < place-first, i++)
        head1 = head1->next;
    for (i=0, i < place-second, i++)
        head2 = head2->next;

    temp = head1;
    head1 = head2;
    head2 = temp;
    return (list-head);
}
```



מס' ת.ז.

מס' קורס : 0509.1821  
 סמסטר א'+ב' תשס"ב  
 בחינת מעבר מועד ב'  
 מועד הבחינה : 20.8.02  
 משך הבחינה : 3 שעות

**מבחן בקורס "תכנות להנדסה"**  
 ד"ר זוהר נאור, ד"ר ברק שוהם, עדי אבידור

- מותר השימוש בדף עזר אחד בלבד הכתוב בכתב יד
- יש לענות על כל השאלות
- התשובות לכל השאלות בשפת C
- **יש לענות על השאלות בגוף הבחינה. טיוטא תכתב במחברת.**

בחינה זו מכילה 6 עמודים (כולל עמוד זה).

בהצלחה!

4	3	2	1



**שאלה 1 (25 נקודות)**

א. (15 נקודות).

כתוב פונקציה המקבלת מספר שלם  $x$  ומחזירה 1 אם  $x$  מתחלק ב-7 וגם ב-19, ואפס אחרת.

ב. (10 נקודות).

כתוב פונקציה המקבלת מערך של מספרים שלמים, ומספר חיובי  $n$  המציין את מספר האברים במערך, ומדפיסה את כל אברי המערך המתחלקים גם ב-7 וגם ב-19. הנך רשאי להשתמש בפונקציה מסעיף א' גם אם לא פתרת אותו.

פתרון שאלה 2

שאלה 2 (25 נקודות)

כתוב תוכנית הקוראת את תוי הקלט ומדפיסה את האות השכיחה ביותר (כלומר האות שהופיעה הכי הרבה פעמים). אם יש יותר מאות אחת "הכי שכיחה", יש להדפיס רק אות אחת. הנח כי הקלט מכיל רק אותיות קטנות (באנגלית), רווחים ו-new lines, פרט לתו האחרון בקלט שהינו הספרה 5.

לדוגמה : אם הקלט הוא : 5 to be or not to be  
ידפיס את האות 'o'.

פתרון שאלה 1

### שאלה 3 (25 נקודות)

כתוב פונקציה `unsigned long string2long (const char *s, int n)` המקבלת מחרוזת `s` שכל תויה הם ספרות בלבד (פרט לתו האחרון שהינו `'\0'`) ומספר חיובי `n`. הפונקציה מחזירה את הערך המספרי של `n` התוים הראשונים. אם `n` גדול מאורך המחרוזת הפונקציה תחזיר את הערך המספרי של כל תוי המחרוזת. לדוגמא:

`String2long("5349",2)` תחזיר את המספר 53.  
`String2long("5349",3)` תחזיר את המספר 534.  
`String2 long("0047",3)` תחזיר את המספר 4.  
בסעיף זה אין להשתמש בפונקציות ספריה.

פתרון שאלה 3

**שאלה 4 (25 נקודות)**

- א. (5 נקודות).  
הגדר טיפוס נתונים שמשתנה מסוגו יוכל להכיל תאריך לועזי (יום, חודש ושנה).
- ב. (5 נקודות)  
רשומת אזרח במשרד הפנים כוללת מס' תעודת זהות בן 9 ספרות, שם פרטי, שם משפחה ותאריך לידה.  
הגדר טיפוס נתונים שמשתנה מסוגו יוכל להכיל רשומת אזרח במשרד הפנים.
- ג. (8 נקודות)  
כתוב פונקציה המקבלת שתי רשומות מהסוג שהוגדר בסעיף א' ומחזירה 1 אם התאריך הראשון קודם לשני, אפס אם הם אותו תאריך ו-1 - אם התאריך השני קודם לראשון.
- ד. (7 נקודות)  
כתוב פונקציה המקבלת מערך של רשומות אזרחים (מהסוג שהוגדר בסעיף ב'), ומספר חיובי  $n$  המציין את מספר הרשומות במערך. הפונקציה תחזיר מצביע לרשומת האזרח המבוגר ביותר (הנח כי במערך יש בדיוק אחד כזה).

פתרון שאלה 4

27

מס' קורס : 0509.1821  
סמסטר ב' תשס"ב  
בחינת מעבר מועד א'  
מועד הבחינה : 1.7.02  
משך הבחינה : 3 שעות

**מבחן בקורס "תכנות להנדסה"**  
ד"ר זוהר נאור, עדי אבידור

- מותר השימוש בדף עזר אחד בלבד
- יש לענות על כל השאלות
- התשובות לכל השאלות ב C
- יש לענות על השאלות בגוף הבחינה. טיוטא תכתב במחברת.

בחינה זו מכילה 9 עמודים (כולל עמוד זה).

בהצלחה!

1	2	3	4
23	25	25	25

**שאלה 1 (25 נק')**

כתוב תוכנית המקבלת כקלט סדרה של מספרים שלמים ולא שליליים בתחום 0 עד 100, המתארים ציוני קורס של סטודנטים. אורך הקלט (סדרת המספרים) איננו ידוע מראש, וסוף הקלט מציין על ידי המספר -1, אשר איננו שייך לקלט. על התוכנית למצוא ולהדפיס את:

א. הציון הגבוה ביותר בקורס. (5 נקודות)

ב. הציון הנמוך ביותר בקורס (5 נקודות)

ג. הציון הממוצע בקורס (15 נקודות)

ניתן להניח כי סדרת הקלט איננה ריקה, והיא מכילה לפחות ציון אחד.

```

#include <stdio.h>
void main()
{
    int min,max,i,a,sum;
    float avg;
    printf("\n enter grade :");
    scanf("%d",&a);
    min=a;
    max=a;
    sum=0;
    for (i=0; a!=-1; i++)
    {
        if (min>a)
            min=a;
        if (max<a)
            max=a;
        sum+=a;
        printf("\n enter grade :");
        scanf("%d",&a);
    }
    avg=(sum/i);
    printf("\n the biggest max grade given is:%d",max);
    printf("\n the min grade given is:%d",min);
    printf("\n the average grade is:%f",avg);
}

```

מספר זמני הפסקה ←

**שאלה 2: (25 נק')**

כתוב תוכנית המקבלת כקלט סדרה אחת רציפה באורך לא מוגבל של תווים המסתיימת בתוו `\n` (enter) . הקלט מכיל תווים כלשהם למעט התוו `\n` המציין את סוף הקלט. על התוכנית לחשב ולהדפיס את סכום הספרות המופיעות המופיעות בקלט.  
לדוגמא, אם מחרחת הקלט היא: `xy1ab7?!z83abc\n` אזי התוכנית צריכה להדפיס את המספר 19,  
(  $19 = 3+8+7+1$  ).



```
#include <stdio.h>
void main()
{
    int sum;
    char c; ✓
    sum=0; ✓
    printf("enter kelet");
    while ((c=getchar())!='\n') ✓
    {
        if ((c<='9') && (c>'0')) ✓
            sum+= (c-'0'); ✓
    }
    printf("in the sum of the numbers in the kelet is %d", sum); ✓
}
```

**שאלה 3: (25 נק')**

א. כתוב פונקציה `void strright(const char s[], char dest[], int n)` המקבלת כפרמטרים מחרוזת `dest, s` (`s` ו-`dest` הן מחרוזת חוקיות ב-C), ומספר שלם `n`. הפונקציה מעתיקה את `n` התווים הראשונים במחרוזת `s` אל תוך המחרוזת הריקה `dest`. הפונקציה מניחה כי המחרוזת `dest` ריקה, אך יש בה מספיק מקום להכיל `n` תווים. אם המחרוזת `s` מכילה פחות מ- `n` תווים, היא תועתק בשלמותה אל `dest`.  
(10 נקודות)

ב. כתוב פונקציה `void strleft(const char s[], char dest[], int n)` המקבלת כפרמטרים מחרוזת `s`, `dest` (`s` ו-`dest` הן מחרוזת חוקיות ב-C), ומספר שלם `n`. הפונקציה מעתיקה את `n` התווים האחרונים במחרוזת `s` אל תוך המחרוזת הריקה `dest`. ההעתקה מתבצעת מהסוף להתחלה, כלומר: התו האחרון במחרוזת `s` יהיה התו הראשון במחרוזת `dest`, התו הלפני אחרון במחרוזת `s` יהיה התו השני במחרוזת `dest`, וכך הלאה. הפונקציה מניחה כי המחרוזת `dest` ריקה, אך יש בה מספיק מקום להכיל `n` תווים. אם המחרוזת `s` מכילה פחות מ- `n` תווים, היא תועתק בשלמותה אל `dest`, מהסוף להתחלה, כלומר בסדר תוים הפוך.  
(15 נקודות)

**הערה: בשאלה זאת אסור להשתמש בפונקציות ספרייה!**

```

Void strright (const char s[], char dest[], int n)
{
    int i;
    for (i=0; (i<n) && (s[i]!='\0'); i++)
        dest[i] = s[i];
    dest[i] = '\0';
}
    
```

3

↘ 10

```

Void strleft (const char s[], char dest[], int n)
{
    int x, i, j;
    for (x=0; s[x]!='\0'; x++)
        for (i=x-1, j=0; (i>=0) && (j<n); i--, j++)
            dest[j] = s[i];
    dest[j] = '\0';
}
    
```

3

כמה טיפים קטנים - 5

15

שאלה 4: (25 נק.)

רשומה של סטודנט בקורס כוללת את שם הסטודנט - מחרחת באורך של 20 תוים, ציון בקורס - מספר שלם בתחום 0 עד 100, ומספר זיהוי - מספר שלם.

א. הגדר טיפוס נתונים שמשתנה מסוגו יוכל להכיל רשומה של סטודנט בקורס (5 נקודות).

ב. כתוב פונקציה בשם `print_students` המקבלת שני משתנים מועברים: מערך המכיל את כל רשומות הסטודנטים בקורס, ומספר שלם  $n$  המבטא את מספר הסטודנטים בקורס. על הפונקציה להחזיר את מספר הסטודנטים בקורס אשר ציונם גבוה מאשר הציון הממוצע בקורס, ולהדפיס את שמות כל הסטודנטים אשר ציונם גבוה מאשר הציון הממוצע בקורס, סטודנט אחד בכל שורה (10 נקודות).

ג. כתוב פונקציה בשם `find_student` המקבלת שלושה משתנים מועברים: מערך המכיל את כל רשומות הסטודנטים בקורס, מספר טבעי  $n$  המבטא את מספר הסטודנטים בקורס, ומספר שלם  $x$  המתאר מספר זיהוי של סטודנט. אם ברשימת הסטודנטים בקורס מצוי סטודנט אשר שדה מספר הזיהוי שלו שווה ל- $x$ , על הפונקציה להחזיר את ציונו של הסטודנט בקורס. אם הסטודנט איננו רשום בקורס (כלומר - לא רשום במערך) - על הפונקציה להחזיר את המספר -1 (10 נקודות).

```
struct stu_record
{
    char name[20];
    int grade;
    int t;
};
```

4

```
int Print_students (struct stu_record stu[], int n)
```

4

```
{
    int sum, i, num;
    float avg;
    sum = 0;
    num = 0;
    for (i = 0; i < n; i++)
        sum += stu[i].grade;
    avg = (sum/n);
    printf("\n the students with grades higher than average are :");
    for (i = 0; i < n; i++)
        if (stu[i].grade > avg)
            { printf("\n name : %s", stu[i].name);
              num++; }
    return (num);
}
```

10

if, } }

```
int find_student (struct stu_record stu[], int n, int x)
```

4

```
{
    int i, t = -1;
    for (i = 0; i < n; i++)
        if (stu[i].t == x)
            t = i;
    if (t < 0)
        return (t);
    else return (stu[t].grade);
}
```

15

(t < 0)

75



סמסטר אי תשס"ב  
בחינת מעבר מועד אי  
משך הבחינה: 3 שעות

**מבחן בקורס "תכנות להנדסה"**  
ד"ר זוהר נאור, ד"ר ברק שהם

- אסור השימוש בחומר עזר.
- יש לענות על כל השאלות
- התשובות לכל השאלות ב C
- יש לענות על השאלות בגוף הבחינה. טיוטא תכתב במחברת.

בחינה זו מכילה 9 עמודים (כולל עמוד זה).

בהצלחה!

4	3	2	1
25	25	25	25

**שאלה 1 (25 נק')**

שני מספרים נקראים "חברים" אם סכום המחלקים של המספר הראשון (כולל אחד, לא כולל המספר עצמו) שווה למספר השני ולהפך – סכום המחלקים של המספר השני (כולל אחד, לא כולל המספר עצמו) שווה למספר הראשון.

לדוגמא: 220 ו-284.

220 : מחלקיו הם: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110 – סכום מחלקיו 284.  
284 : מחלקיו הם: 1, 2, 4, 71, 142 – סכום מחלקיו 220.

כתוב תוכנית הקוראת שני מספרים שלמים ומדפיסה האם הם מספרים "חברים".



פתרון שאלה 1

```
#include <stdio.h>
```

```
int check_friends (int a, int b); /* בודק אם שני מספרים הם חברים (מחזיר 1 אם כן, 0 אם לא) */
```

```
int div_sum (int a); /* מחזיר את סכום החלקים של מספר (כאשר מספר זה מתחלק ב-1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100) */
```

```
int main ( )
```

```
{
  int a, b;
```

```
  printf ("Please enter 2 integer numbers : ");
```

```
  scanf ("%d%d", &a, &b);
```

```
  if ((a < 1) || (b < 1))
```

```
  {
    printf ("positive numbers only! \n");
```

```
  }
  return -1;
```

```
  if (check_friends(a,b))
```

```
    printf ("%d and %d are friends! \n", a, b);
```

```
  else printf ("%d and %d are NOT friends. \n", a, b);
```

```
  return 0;
```

```
}
int check_friends (int a, int b)
```

```
{
  if ((div_sum(a) == b) && (div_sum(b) == a))
```

```
    return 1;
```

```
  return 0;
```

```
}
int div_sum (int a)
```

```
{
  int i, sum = 0;
```

```
  for (i = 1; i <= (a/2); ++i) /* מחזיר את סכום חלקי a (כאשר a/2 הוא מספר שלם) */
```

```
  {
    if ((a%i) == 0)
```

```
      sum += i;
```

```
  }
  return sum;
```



שאלה 2: (25 נק')

א. כתוב תוכנית המקבלת כקלט סדרה של לכל היותר 1000 מספרים שלמים וחיוביים. במידה והסדרה מכילה פחות מאשר 1000 מספרים מצויין סוף הקלט על ידי המספר 0, אשר איננו חלק מהסדרה. במידה והסדרה מכילה 1000 מספרים, תסתיים קריאת הקלט לאחר קריאת המספר ה-1000 בסדרה. על התוכנית להדפיס את:

1. המספר הגדול ביותר בסדרה ( 5 נקודות).
2. המספר הקטן ביותר בסדרה ( 5 נקודות).
3. את כל המספרים בסדרה הגדולים מממוצע הסדרה ( 15 נקודות).

ניתן להניח כי סדרת הקלט איננה ריקה, כלומר: הסדרה מכילה לפחות מספר אחד.

עמוד 101 של הספר  
 2 - הממוצע של מספרים - כתיבת קוד

```
#include <stdio.h>
#define MAX_NUMS 1000
float avg(unsigned int *nums, int n_nums); /* מציאת הממוצע (Float) */
int main ()
{
    unsigned int max, min;
    unsigned int nums[MAX_NUMS];
    int i, n_nums;
    float nums_avg;
    printf ("Please enter up to %d numbers (0 to end):\n", MAX_NUMS);
    for (i=0; i<MAX_NUMS; ++i)
    {
        scanf ("%d", &(nums[i]));
        if (nums[i] == 0)
            break;
    }
    n_nums = i;
    max = min = nums[0];
    for (i=0; i<n_nums; ++i)
    {
        if (nums[i] > max)
            max = nums[i];
        if (nums[i] < min)
            min = nums[i];
    }
    printf ("max: %d\n", max);
    printf ("min: %d\n", min);
    nums_avg = avg (nums, n_nums);
    printf ("Average: %f\n", nums_avg);
    for (i=0; i<n_nums; ++i)
        if (nums[i] > nums_avg)
            printf ("%d ", nums[i]);
    return 0;
}
```

```
float avg(unsigned int *nums, int n_nums)
{
    int i;
    float sum = 0.0;
    for (i=0; i<n_nums; ++i)
        sum += nums[i];
    return sum/n_nums;
}
```

```
#include <stdio.h>
```

```
#define MAX_NUMS 1000
```

```
float avg(unsigned int *nums, int n_nums); /*
```

```
int main()
```

```
{
    unsigned int min, max;
    unsigned int nums[MAX_NUMS];
```

```
    int i, j, n_nums;
```

```
    float nums_avg;
```

```
    printf("Please enter numbers (0 to end): \n");
```

```
    for (i=0; i<MAX_NUMS; ++i)
```

```
    {
        scanf("%d", &(nums[i]));
```

```
        if (nums[i] == 0)
```

```
            break;
```

```
    }
```

```
    n_nums = i;
```

```
    max = min = nums[0];
```

```
    for (i=0; i<n_nums; ++i)
```

```
    {
        if (nums[i] > max)
```

```
            max = nums[i];
```

```
        if (nums[i] < min)
```

```
            min = nums[i];
```

```
    }
```

```
    printf("max: %d \n", max);
```

```
    printf("min: %d \n", min);
```

```
    nums_avg = avg(nums, n_nums);
```

```
    printf("Numbers above average: \n");
```

```
    for (i=0; i<n_nums; ++i)
```

```
        if (nums[i] > nums_avg)
```

```
            printf("%d ", nums[i]);
```

```
    return 0;
```

```
float avg(unsigned int *nums, int n_nums)
```

```
{
    int i;
```

```
    float sum = 0.0;
```

```
    for (i=0; i<n_nums; ++i)
```

```
        sum += nums[i];
```

```
    return sum/n_nums;
```

**שאלה 3: (25 נק')**

כתוב פונקציה `int strt(char s[], char t)` המקבלת כפרמטרים מחרוזת `S`, המכילה תווים כלשהם (`S` היא מחרוזת חזוקית ב-`C`), ותו `t`. הפונקציה:  
א. מחזירה כמה פעמים מופיע התו `t` במחרוזת `S` (10 נקודות).  
ב. משנה את המחרוזת `S`, על ידי כך שהיא מוחקת כל מופע של התו `t` ב-`S`. לדוגמא:  
אם `S="abcaa2Ba3,Y"`, `t='a'`, לאחר הפעלת הפונקציה המחרוזת `S` תראה כך: `S="bc2B3,Y"`  
(15 נקודות)

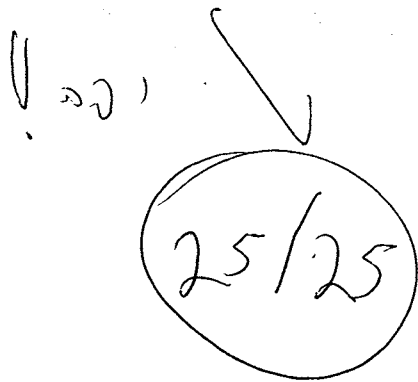
פתרון שאלה 3

```

int strt (char s[], char t)
{
    int n_app, i, j;           /* n_app הוא מספר האותיות שמתחת ל-t */
    j=0;
    n_app=0;
    for (i=0; s[i]!='\0'; ++i)
    {
        if (s[i]==t)
            n_app++;
        else
            s[j++] = s[i];
    }
    s[j] = '\0';
    return n_app;
}

```

/\* שני האותיות באותיות s ו-t  
 הן באותיות שונות מדי,  
 ומאחר שהאותיות שונות... \*/



שאלה 4: (25 נק.)

נקודה במישור מוגדרת בעזרת שני מספרים ממשיים  $x$  ו- $y$ .

א. הגדר טיפוס נתונים שמשתנה מסוגו יוכל להכיל נקודה במישור (5 נקודות).

ב. מרחק בין שתי נקודות  $(x_1, y_1)$ ,  $(x_2, y_2)$  במישור,  $d$ , מוגדר ע"י:

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

כתוב פונקציה המקבלת שני פרמטרים המתארים מערך בן  $n$  נקודות במישור: הפרמטר הראשון הוא מערך של נקודות, והפרמטר השני הוא מספר הנקודות במערך. על הפונקציה:

1. להחזיר את המרחק המינימלי בין 2 נקודות במערך (10 נקודות).
2. להדפיס את הקואורדינטות של שתי הנקודות שהמרחק ביניהן מינימלי. אם יש יותר מאשר זוג נקודות אחד, מספיק להדפיס רק זוג אחד (10 נקודות).

```

#include <math.h>

typedef struct -point {
    float x;
    float y;
} point_t;

float min_dist(point_t *points, int n_points)
{
    int i, j, a, b; /* כאן קצתם עשיתם כתיבות ויקראו a, b
                    כיוון */
    float min, cur_dist;
    a = 0; b = 1;
    min = pow((points[0].y - points[1].y), 2) + pow((points[0].x - points[1].x), 2);
    for (i = 0; i < n_points; ++i)
        for (j = i + 1; j < n_points; ++j) /* i+1 כי אם נשאיר
                                             את j בקטן, אזי לא
                                             אסטמטיק בין אותם נקודות
                                             עדיף */
        {
            cur_dist = pow((points[i].y - points[j].y), 2) +
                pow((points[i].x - points[j].x), 2);
            if (cur_dist < min)
            {
                min = cur_dist;
                a = i;
                b = j;
            }
        }
    printf ("Nearest points are (%.2f, %.2f) and (%.2f, %.2f)\n",
        points[a].x, points[a].y, points[b].x, points[b].y);
    return sqrt(min);
}
    
```

2000/2001 סמסטר ב', C תכנות בשפת C

הוראות: משך המבחן: 3 שעות. חומר עזר מותר: דף פוליו בודד. יש לפתור את 4 השאלות הבאות. הניקוד על כל אחת מן השאלות שווה.

1. כתבו את הפונקציה  $\text{double cos}(\text{double } x, \text{int } n)$  המחשבת קרוב ל  $\cos(x)$  תוך שימוש ב  $n$  האיברים הראשונים בקירוב טיילור. אין להשתמש בפונקציות ספריה של C.

$n$  האיברים הראשונים בקרוב טיילור של  $\cos(x)$ :

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + \frac{(-1)^{n-1} x^{2(n-1)}}{(2(n-1))!}$$

2. כתבו פונקציה רקורסיבית להכפלת מטריצות ריבועיות מממד  $n$ , באשר  $n=2^k$ . הפונקציה תקבל שלוש מטריצות  $A, B, C$  (שתי המטריצות המוכפלות + המכפלה) ואת הממד  $n$ . אם  $n > 1$ , הפונקציה תבצע כפל מטריצות בבלוקים, כלומר,

$$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22} \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22} \end{aligned}$$

באשר  $A = \{A_{ij}\}, B = \{B_{ij}\}, C = \{C_{ij}\}$   $i, j \in \{1, 2\}$  (שימו לב ש  $A_{ij}, B_{ij}$ , הם בלוקים מממד  $n/2$  כל אחד)

אם  $n=1$ , הפונקציה תבצע הכפלה של סקלרים.

3. כתבו תכנית הקולטת מהמקלדת מספר שלם חיובי  $n$ , ולאחריו  $n$  תווים (תו=char). בסיום, על התכנית להדפיס "1" אם רצף התווים שהוקלד מהווה פלינדרום, ו "0" אחרת. (פלינדרום הנו



מחרוזת  $a(1) a(2) \dots a(n)$  אשר מקיימת  $a(k)=a(n-k+1)$  עבור  $k=1, \dots, n$   
(דוגמאות: ABBA, DND, pop)  
בשאלה זו אין להניח ש  $n$  קטן ממספר כלשהו.

4. כתבו את הפונקציה

```
struct rational RationalDecrease(struct rational a)
```

המקבלת מספר רציונלי ומחזירה את המספר כשהוא מצומצם.  
הניחו ש `struct rational` מוגדר באופן הבא:

```
struct rational {  
    int numerator; /* מונה */  
    int denominator; /* מכנה */  
};
```

67



סמסטר א' תש"סא  
בחינת מעבר מועד א'  
משך הבחינה: 3 שעות

הנדסת חשמל  
הנדסת תצורה

מבחן בקורס "תכנות להנדסה" – ברק שהם

- מותר השימוש בדף עזר אחד בלבד
- יש לענות על כל השאלות
- התשובות לכל השאלות ב C++
- אין להשתמש בפונקציות ספרייה, למעט cin ו- cout
- יש לענות על כל שאלה במחברת נפרדת.

1. (36 נקודות)

א. (18 נקודות) קרא סדרה המכילה לכל היותר 100 מספרים שלמים השונים מ-0. המספר 0 (אשר איננו שייך לסידרה) מסמן את סוף הקלט. יש להדפיס את שלושת המספרים הגדולים ביותר בסדרה בסדר יורד, ואת שלושת המספרים הקטנים ביותר בסדרה בסדר עולה. יש להדפיס כך שמשמעות ההדפסה תהיה ברורה.

ב. (18 נקודות)

כתוב תוכנית הקוראת קלט תווי באורך בלתי מוגבל המורכב מספרות, התו ' / ', התו '\$' ורווחים. התכנית תחשב את סכום הספרות בין סימני ' / ' עוקבים. בין התווים יתכן שיופיעו רווחים (לא בהכרח). סוף הקלט מצוין ע"י '\$'.

דוגמא: קלט: \$ / 1 2 5 / 4 1 8 2 / 1 2 5 4 / \$  
פלט: 8 15 12

2. (36 נקודות)

א. (18 נקודות)

כתוב פונקציה my\_sin (double x) המחשבת את sin(x) בעזרת הנוסחה:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

עד (לא כולל) האלמנט הראשון בפיתוח לטור שקטן בערכו מ- 0.001.  
רק חישוב יעיל יקבל את מלוא הנקודות!

ב. (18 נקודות)

(1) (2 נקודות)

כתוב structure בשם complex המייצג מספר מרוכב. complex מכיל שני שדות: real ו-image שניהם מטיפוס float, המתארים את הרכיבים הממשי והמדומה של המספר המרוכב.

(2) (8 נקודות)

כתוב פונקציה void add (complex a, complex b, complex &c) המקבלת שני מספרים מרוכבים a ו-b, ומחשבת את  $c=a+b$ .

(3) (8 נקודות)

כתוב פונקציה sub (complex a, complex b, complex &c) המקבלת שני מספרים מרוכבים a ו-b, ומחשבת את המספר המרוכב  $c=a-b$ .

3. (28 נקודות)

א. (14 נקודות)

נתונה רשימה מקושרת הבנויה מאיברים מהטיפוס הבא:

```
struct NODE {  
    int num;  
    struct NODE* next;  
};
```

כתוב פונקציה המקבלת שני פרמטרים: 1. מצביע לאיבר הראשון ברשימה 2. מספר שלם.

הפונקציה תחזיר את מספר המופעים של המספר ברשימה.

ב. (14 נקודות)

נתונה רשימה מקושרת הבנויה מאיברים מהטיפוס הבא:

```
struct NODE {  
    int num;  
    struct NODE* next;  
};
```

כתוב פונקציה: struct NODE\* MakeEvenList(struct NODE \*head). הפונקציה תיצור רשימה מקושרת חדשה המכילה עותק של האיברים בעלי ערך זוגי בשדה num של הרשימה המתקבלת כפרמטר. הפונקציה תחזיר מצביע לתחילת הרשימה החדשה.

בהצלחה!



```

#include <iostream.h> // (18) cin, cout
const int SIZE = 100; // (19) k מנייה לזכרון
const int STOP = 0; // (20) מנייה לזכרון

void sort(int a[], int size) { // מיון בועות
    int i, j, temp; // מניין מהאיר הקול קטן // גודל המערך size
    for (i=0; i < size-1; i++)
        for (j=0; j < size-i-1; j++)
            if (a[j] > a[j+1]) { // מניין מהאיר הקול קטן
                temp = a[j]; // הנכנסת אל הקול המערך
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}

```

```

void print(int a[], int size) { // מודפס כל המספרים
    int i; // מניין מהאיר הקול קטן
    if (size <= 3) { // כמות האברים קטנה או שניה
        cout << "Minimals:\n"; // מודפס המספרים
        for (i=0; i < size; i++) // מניין מהאיר הקול קטן
            cout << a[i] << " ";
        cout << "Maximals:\n"; // מודפס המספרים
        for (i=size-1; i >= 0; i--) // מניין מהאיר הקול קטן
            cout << a[i] << " ";
    }
    else {
        cout << "Minimals:\n"; // מודפס המספרים
        for (i=0; i < 3; i++) // מניין מהאיר הקול קטן
            cout << a[i] << " ";
        cout << "Maximals:\n"; // מודפס המספרים
        for (i=size-1; i >= size-3; i--) // מניין מהאיר הקול קטן
            cout << a[i] << " ";
    }
}

```

```

main()
// מניין מהאיר הקול קטן

```

```

// קודם להגדרת המספרים
void read() { // קודם להגדרת המספרים
    int num, i, nums[SIZE];
    // מספר המינימום, המספר המקסימום, מספר המספרים
    i = 0; // מספר המינימום
    cin >> num; // קודם להגדרת המספרים ✓

    while (cin && i < SIZE && num != stop) { // קודם להגדרת המספרים
        nums[i] = num; // קודם להגדרת המספרים ✓
        i++; // קודם להגדרת המספרים ✓
        cin >> num; // קודם להגדרת המספרים
    }

    if (i == 0) { // קודם להגדרת המספרים
        cout << "Empty list\n"; // קודם להגדרת המספרים
        return; // קודם להגדרת המספרים
    }

    sort(nums, i); // קודם להגדרת המספרים
    print(nums, i); // קודם להגדרת המספרים
}

void main() { // קודם להגדרת המספרים
    read(); // קודם להגדרת המספרים
}

```

מספר המינימום, המספר המקסימום, מספר המספרים

```
# include <iostream.h>
```

scanf

```
void read() {  
    char c; int count; int flag1, flag2;  
    count = 0;  
    c = cin.get();  
    size = 0; flag1 = 0; flag2 = 0;  
    while (c != EOF && c != '$') {  
        size++;  
        if (c == '1') {  
            if (size == 7)  
                flag1 = 1;  
            if (c != '1') {  
                if (flag1 == 1) {  
                    cout << count << ' ';  
                    count = 0;  
                }  
            }  
            else {  
                flag1 = 1;  
            }  
        }  
        else {  
            if (isdigit(c))  
                count += c - '0';  
        }  
        c = cin.get();  
    }  
    cout << "\n";  
}
```

}

```

#include <iostream.h> // cout cin גורר
int isdigit(char c) { // מתבדק האם האות (char) היא ספרות
    return ('0' <= c && c <= '9');
}

void read() { // בונה קובץ של מילים עם קובץ המילים
    char c;
    int count, flag, size; // flag - מצביע האם הוכנס האות החדש
    count = flag = size = 0; // מציבים את המשתנים
    c = cin.get(); // קולט את האות
    while (c != EOF && c != '$') { // כל עוד לא הסתיים הקלט
        size++; // הגדלת גודל המילה
        if (c == ' ') { // אם יש רווח
            if (size == 1) // אם המילה היא רווח
                flag = 1;
            else if (flag == 1) { // אם הוכנס רווח
                cout << count << " "; // הדפסת המילה
                count = 0; // אפס מחדש
            }
            else // אכן הוכנס האות
                flag = 1;
        }
        else if (isdigit(c)) // אם ספרות
            count += c - '0'; // חישוב המיקום
        else if (c != '$') { // אם לא רווח
            cout << "Error in input\n"; // הודעת שגיאה
            return;
        }
    }
}

void main() { // קובץ המילים
    read();
}

```



```

double my_sin(double x) {
    if (x == 0) return 0;
    int power; int pos = 1;
    long fact;
    double total, organ;

    power = 1;
    organ = x;
    total = organ;

    while (abs(organ) >= 0.001) {
        power += 2;
        organ = (organ * x * x) / ((power) * (power - 1));

        if (pos) {
            organ = -organ;
            pos = 0;
        } else {
            organ = organ;
            pos = 1;
        }
        total += organ;
    }

    return total;
}

```

$\frac{10}{10} \frac{11}{11} \frac{12}{12}$   
 $(10) \frac{10, 2}{11 \cdot 11 \cdot 11}$

	total	power	pos	organ
	0.5	1	1	0.5
	$0.5 - \frac{(0.5)^3}{3 \cdot 2}$	3	0	$\frac{(0.5)^3}{3 \cdot 2}$
0.5	$0.5 - \frac{(0.5)^5}{3 \cdot 2}$	5		$-\frac{(0.5)^3}{3 \cdot 2}$ $+\frac{(0.5)^5}{3 \cdot 2 \cdot 5 \cdot 4}$

מחלקה

20

// פונקציה המחזירה את הערך המוחלט של מספר

```
double abs(double n) {
  if (n < 0) // ? if
    return -n; // החזרה של הערך הנגדי
  else
    return n;
}
```

כדי להימנע מבעיה של overflow, נשתמש ב-atan2

```
double my_sin(double x) {
  int power; // מספר החזקה
  double organ; // האיבר הנוכחי
  double total; // סכום האיברים
  power = 1; // החזקה הראשונה
  organ = x; // האיבר הראשון
  total = organ; // סכום האיברים הראשון
  while (abs(organ) >= 0.001) {
    power += 2; // החזקה הבאה
    organ = (organ * x * x) / ((power) * (power - 1));
    organ = (-1) * organ;
    total += organ;
  }
  return total;
}
```

```
power = 1; // החזקה הראשונה
organ = x; // האיבר הראשון
total = organ; // סכום האיברים הראשון
```

```
while (abs(organ) >= 0.001) {
  power += 2; // החזקה הבאה
```

```
organ = (organ * x * x) / ((power) * (power - 1));
organ = (-1) * organ;
```

```
total += organ;
}
```

```
return total;
}
```

// פונקציה המחזירה את הערך המוחלט של מספר

18

מחלקה

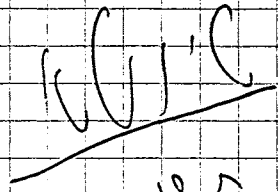
37

```
typedef struct comp {  
    float real;  
    float image;  
} complex;
```

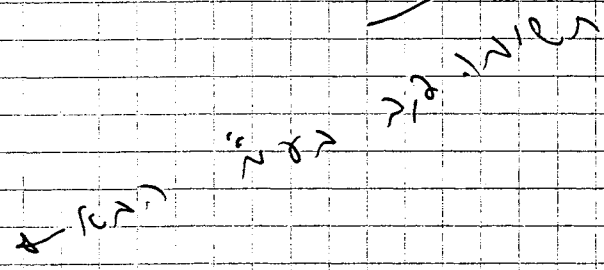
②

①

void



②

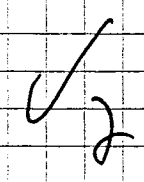


```

typedef struct comp {
    float real; // ריאל
    float image; // אימג'ינר
} complex;

```

1 2 3

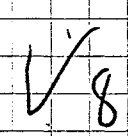


```

void add (complex a, complex b, complex & c) {
    c.real = a.real + b.real; // סכום ריאל
    c.image = a.image + b.image; // סכום אימג'ינר
}

```

2

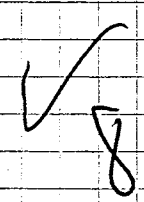


```

void sub (complex a, complex b, complex & c) {
    c.real = a.real - b.real; // הפרק ריאל
    c.image = a.image - b.image; // הפרק אימג'ינר
}

```

3



39

```
int count (struct NODE* head, int n) {
```

```
int occur;
```

```
struct NODE* temp;
```

```
temp = head;
```

```
occur = 0;
```

```
while (temp) {
```

```
    if (temp->num == n)
```

```
        occur++;
```

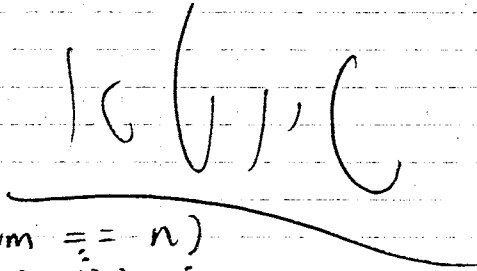
```
    temp = temp->next;
```

```
}
```

```
return occur;
```

```
}
```

3  
C



C

40

ג'לד

3

```

int count(struct NODE* head, int n) {
    int occur = 0; // פועל סתם
    struct NODE* temp = head; // temp - נקודת התחלה
    // מניין - occur
    // מניין - temp

    while (temp) { // (מניין) NULL נגה temp זרז
        if (temp->num == n) // המספר מניין
            occur++; // הוסיף 1
        temp = temp->next; // מעבר למקום הבא
    } // סוף פונקציה

    return occur; // המספר (מניין)
}

```

3 // מניין

// סוף פונקציה 3

```
#include <iostream.h> // cout ג'עבן
#include <stdlib.h> // exit() ג'עבן
```

41

3  
2

```
struct NODE { // אגור ברעיאנע
    int num;
    struct NODE* next;
};
```

```
void assert(struct NODE* ptr) { // אגור פאראן אגור אגור
    if (ptr == NULL) { // אגור אגור אגור
        cout << "Memory Error.\n"; // אגור אגור אגור
        exit(1); // terminate - error code 1. - אגור
    }
}
```

```
struct NODE* MakeEvenList(struct NODE* head) { // אגור אגור
    struct NODE* newhead; // אגור אגור
    struct NODE* index1; // אגור אגור
    struct NODE* index2; // אגור אגור
```

```
    index1 = head; // אגור אגור
    index2 = newhead = NULL; // אגור אגור
```

```
    while (index1) { // אגור אגור
        if (index1->num % 2 == 0) { // אגור אגור
            if (index2 == NULL) { // אגור אגור
                index2 = newhead = new struct NODE; // אגור אגור
                assert(newhead); // אגור אגור
                index2->num = index1->num; // אגור אגור
                index2->next = NULL; // אגור אגור
            }
            else { // אגור אגור
                index2->next = new struct NODE; // אגור אגור
                assert(index2->next); // אגור אגור
                index2->next->num = index1->num; // אגור אגור
                index2->next->next = NULL; // אגור אגור
                index2 = index2->next; // אגור אגור
            }
        }
        else { // אגור אגור
            index2->next = new struct NODE; // אגור אגור
            assert(index2->next); // אגור אגור
            index2->next->num = index1->num; // אגור אגור
            index2->next->next = NULL; // אגור אגור
            index2 = index2->next; // אגור אגור
        }
    } // אגור אגור
```

```
    } // אגור אגור
```

```
    index1 = index1->next; // אגור אגור
} // אגור אגור
```

```
return newhead; // אגור אגור
```

```

int
struct NODE* MakeEvenList(struct NODE* head) { (2)
    struct NODE* newhead,
    struct NODE* index1;
    struct NODE* index2;

    index1 = head;
    index2 = newhead = NULL;

    while (index1) {
        if (index1->num % 2 == 0) {
            if (index2 == NULL) { // insert at the front
                index2 = newhead = new struct NODE;
                assert(newhead);
                index2->num = index1->num;
                index2->next = NULL;
            }
            else {
                index2->next = new struct NODE;
                assert(index2->next);
                index2->next->num = index1->num;
                index2->next->next = NULL;
            }
            index1 = index1->next;
        }
    }
    return newhead;
}

```





**שאלה 1 (50 נק')**

**סעיף א (25 נק')**

כתוב תוכנית הקוראת קלט תווי באורך לא מוגבל ומדפיסה את כל השלשות הפולינדרומיות. כלומר שלשות אשר קריאתן מימין לשמאל ומשמאל לימין זהה. סוף הקלט מסומן ע"י ' .

דוגמה:

קלט: `abcdcb*b++++/.`

פלט: `cdc`

`b*b`

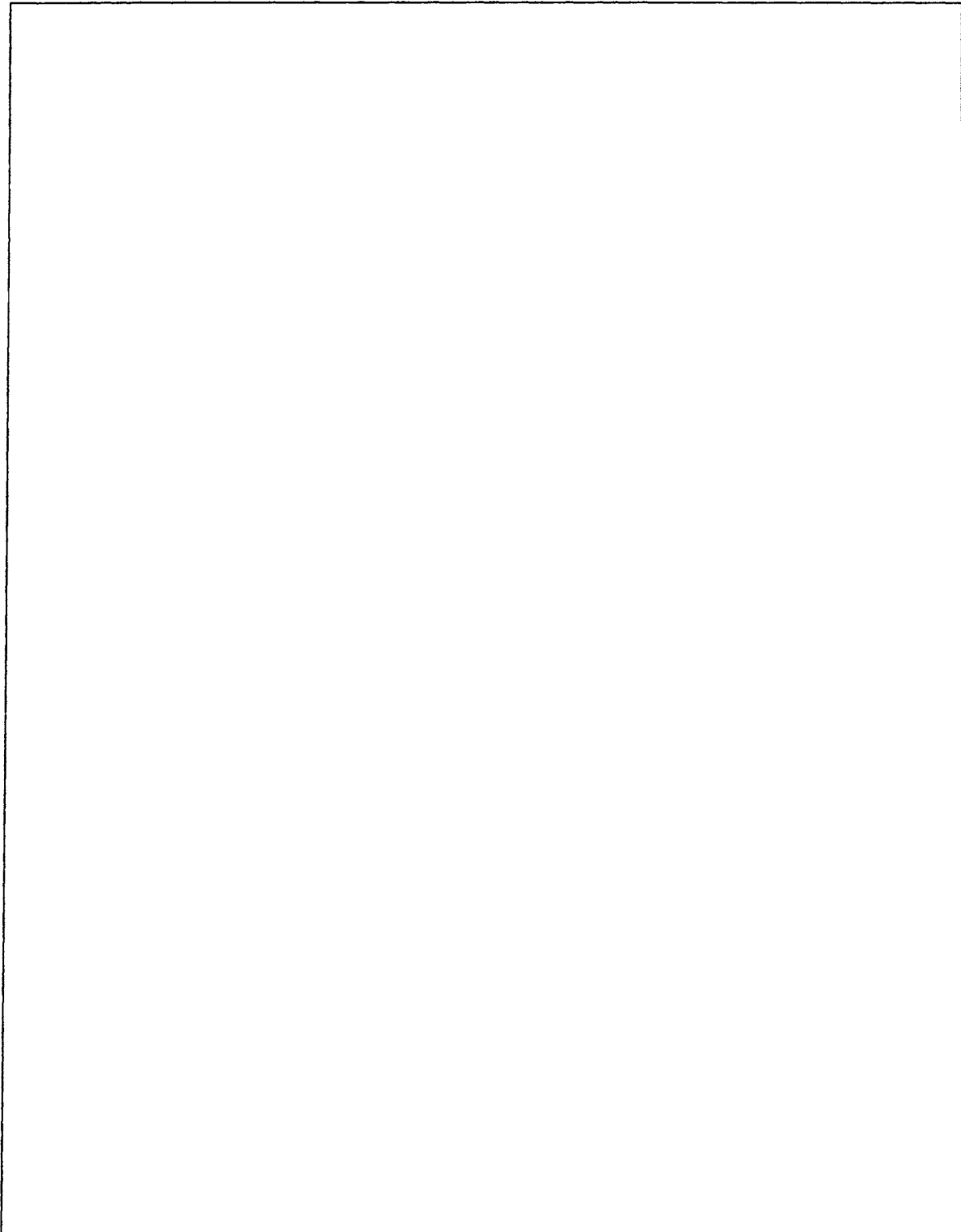
`+++`

`+++`

### סעיף ב (25 נק')

כתוב תוכנית הקוראת מהקלט סידרה של מספרים שלמים (סוף הסדרה מסומן ע"י 1-) ומדפיסה את אורך תת הסדרה הארוכה ביותר המסודרת בסדר עולה ואת אורך תת הסדרה הארוכה ביותר המסודרת בסדר יורד.

### פתרון שאלה 1 סעיף ב



## שאלה 2: (20 נק.)

זוג מחרוזות  $s_1, s_2$  יקראו זהות עד-כדי גודל אות (case-indifference-identical) אם לכל זוג תווים מתאימים  $(s_1[i], s_2[i])$  מתקיים, לפחות אחד מהתנאים הבאים:  
א.  $s_1[i] = s_2[i]$  או  
ב.  $s_1[i]$  היא אות קטנה (a..z) והאות המתאימה לה ב- $s_2$  ( $s_2[i]$ ) היא אותה אות אבל גדולה (A..Z).  
ג.  $s_1[i]$  היא אות גדולה (A..Z) והאות המתאימה לה ב- $s_2$  ( $s_2[i]$ ) היא אותה אות אבל קטנה (a..z).

לדוגמה: 'abC4tm' היא case-indifference-identical לעצמה, ל-'aBC4Tm' ול-'Abc4tm', אבל אינה case-indifference-identical ל-'apc4tm' או ל-'abC4t'.

כתבו פונקציה

```
int caseIndifferenceIdentical(char s1[], char s2[]);
```

המקבלת זוג מחרוזות ומחזירה 1 אם הן case-indifference-identical ו-0 אחרת.

שאלה 3: (30 נק.)

סעיף א (20 נק):

כתוב פונקציה  $\text{int Ndiv}(\text{int } n)$ ; המחשבת את מספר המחלקים של שלם  $n$ . לדוגמה: המחלקים של 12 הם 1, 2, 3, 4, 6, 12, ולכן:  $\text{Ndiv}(12)=6$

פתרון שאלה 3 סעיף א

**סעיף ב (10 נק')::**

כתוב פונקציה `int prime(int n);` המחשבת האם מספר נתון הוא ראשוני. הפונקציה תחזיר ערך 1 אם המספר ראשוני ואפס אחרת. יש להשתמש בפונקציה שבשאלה 4 א' לצורך פתרון.

**פתרון שאלה 3 סעיף ב**

סמסטר א' תשס"א  
מועד: ב' 26/8/2001  
משך הבחינה: 3 שעות  
חומר עזר: שני דפי עזר

בחינה בקורס: תכנות  
מרצה: איריס גבר

**הנחיות כלליות לבחינה:**

- המבחן מורכב משני חלקים:
  - חלק א' כולל שאלות תכנות שמשקלן הכולל 60 נק'.
  - חלק ב' כולל 5 שאלות אמריקאיות. משקל כל שאלה 8 נקודות, סה"כ: 40 נק'.
- חובה לתעד בשאלות התכנות כל פעולה לא ברורה שנעשית.
- בשאלות האמריקאיות יש לסמן תשובה אחת לכל שאלה בטבלה המצורפת. במידה ומספר תשובות נראות נכונות יש לסמן את התשובה הנכונה ביותר.
- בסיום המבחן יש לרשום מס' ת.ז. על גבי טבלת התשובות האמריקאיות ולהגישה בתוך מחברת הבחינה.
- נא לכתוב בכתב קריא ולא מחובר.

**בהצלחה !**

טבלת תשובות לחלק האמריקאי

**! חובה לספק  
הסבר עבור  
תשובה ה'!**

ה	ד	ג	ב	א	
					שאלה 3
					שאלה 4
					שאלה 5
					שאלה 6
					שאלה 7

הערות לתשובות

אמנם רק תשובה נכונה תזכה בניקוד עבור כל שאלה, אולם ניתן לצרף לכל תשובה אמריקאית הסבר קצר. ההסבר לא ייבדק במסגרת הבדיקה הרגילה, אך ניתן יהיה להסתמך עליו במסגרת ערעור, אם יידרש. מומלץ לצרף הסבר לתשובה אמריקאית במיוחד במקרים קיצוניים בהם נראה לך שתשובתך נכונה, אך נראה לך שהיא איננה התשובה שאליה התכוון המרצה. מובן שהסבר שגוי או בלתי סביר לא יועיל בכל מקרה (אך גם לא יזיק, אם ממילא סימנת את התשובה הנכונה).

שאלה 3:	
שאלה 4:	
שאלה 5:	
שאלה 6:	
שאלה 7:	



## שאלה 1 (20 נק')

נתונה פונקציה בשם SquareEq המקבלת 3 מספרים A,B,C המהווים מקדמי משוואה ריבועית:

$$Ax^2 + Bx + C = 0$$

הפונקציה מחזירה את מספר הפתרונות (0,1 או 2) ואת הפתרונות עצמם בארגומנטים הרביעי והחמישי.

אם מספר הפתרונות הינו 0 לארגומנטים אין משמעות. אם מספר הפתרונות הינו 1 הארגומנט הרביעי הינו פתרון המשוואה ואילו לחמישי אין משמעות.

עליך לכתוב:

א. פונקציה בשם Asolution. הפונקציה מקבלת 3 מספרים ומחזירה:  
1 – אם יש למשוואה הריבועית איזשהו פתרון (אחד או שניים).  
0 – אם אין למשוואה הריבועית פתרון כלל.

את הפתרון, אם קיים, תחזיר הפונקציה בארגומנט הרביעי. (אם קיים יותר מאחד, אזי את אחד הפתרונות ואם לא קיים אף פתרון הערך בארגומנט חסר משמעות).

על הפונקציה Asolution להשתמש בפונקציה SquareEq.

ב. תכנית הקולטת שלשות של מספרים A,B,C. על כל שלשה תדפיס התכנית:  
No Solution

או:

A Solution is: x

כאשר x יהיה הפתרון בהנחה שישנו פתרון למשוואה. התכנית תרוץ כל עוד יתקבלו שלושה מספרים שלא כולם 0.

## שאלה 2 (40 נק')

עליך לכתוב תכנית עבור ספריית DVD פשוטה.

סרט DVD יאופיין על ידי:

- מס' יחודי שלם.
- שם הסרט (לא יעלה על 50 תווים).
- האם הסרט מתורגם לעברית: כן/לא.

לקוח בספריה יאופיין על ידי:

- תעודת זהות (מספר יחודי שלם).
- שם הלקוח (לא יעלה על 100 תווים).
- כמות הסרטים שברשותו (לא יותר מ-3).
- רשימת מספרי הסרטים שברשותו.

א. התכנית מתחילה בכך שהיא שואלת כמה סרטים וכמה לקוחות יש, ולאחר מכן, קולטת התכנית את נתוני הסרטים, ואת נתוני הלקוחות (בהתחלה לכל הלקוחות אין כלל סרטים ברשותם).

ב. התכנית תציג תפריט ובו האופציות של:

- קליטת ת.ז. של לקוח – והשכרת סרטים עבורו (לא יותר מ-3). יש לבדוק שסרט שלקוח מעוניין בו אכן נמצא במאגר הסרטים.
- מעבר לבדיקת שגיאות: שגיאה תהיה סרט שנמצא בו-זמנית אצל יותר מלקוח אחד. באם נמצאה שגיאה, התכנית תדפיס הודעת שגיאה ותסיים.
- סיום.

על הקוד שתכתוב להיות בהיר, מובן ומתועד. חשוב תחילה על תכנון נאות וחלוקה לפונקציות.

### שאלה 3

נתונה התכנית הבאה:

```
#include <iostream.h>

float half (int x)
{
    return x/2;
}

float sum (int x)
{
    int N = 3;

    return half(N) + half(x+0.5);
}

void main()
{
    cout << sum(5.5);
}
```

מה תדפיס התכנית?

- א. 3
- ב. 3.5
- ג. 4
- ד. 4.5
- ה. אף תשובה אינה נכונה

#### שאלה 4

נתונה הפונקציה הבאה:

```
int operate (unsigned int a, unsigned int b)
{
    if (a < b)
        return a;
    else
        return operate (a-b, b);
}
```

מהו הערך המוחזר ע"י הפונקציה operate?

- א. הערך של a.
- ב. השארית של חלוקת a ב-b.
- ג. ההפרש של a ו-b.
- ד. המנה של חלוקת a ב-b.
- ה. אף תשובה אינה נכונה.

## שאלה 5

נתון קטע הקוד הבא:

```
int place[] = {5, 3, 4, 6, 0, 7, 2, 1};  
  
int ind=0;  
  
while (1)  
{  
    int tmp = place[ind];  
  
    place[ind] = 0;  
    cout << tmp;  
    ind = tmp;  
  
    if (place[ind] == 0)  
        break;  
}
```

מה ידפיס קטע הקוד?

- |                              |    |
|------------------------------|----|
| 53460721                     | א. |
| 5713624                      | ב. |
| 534                          | ג. |
| התכנית תיכנס ללואה אינסופית. | ד. |
| אף תשובה אינה נכונה.         | ה. |

שאלה 6  
נתונה הפונקציה הבאה:

```
void func()
{
    int arr[10], i, var;
    int a=0, b=9;

    for (i=0; i<10; i++)
    {
        cin >> var;
        if (var < 0)
            arr[a++] = var;
        else
            arr[b--] = var;
    }
}
```

מה מן המשפטים הבאים נכון:

- א. הפונקציה מאחסנת את כל המספרים שהוכנסו ממיינים מהקטן לגדול.
- ב. הפונקציה מאחסנת את כל השליליים לפני כל האי-שליליים.
- ג. הפונקציה עלולה לבצע דריכת זכרון, זאת אם תקבל מספר לא שווה של שליליים ואי-שליליים.
- ד. הפונקציה מבצעת דריכת זכרון בכל מקרה.
- ה. אף אחד מהמשפטים לעיל אינו נכון.

## שאלה 7

נתונה התכנית הבאה:

```
#include <iostream.h>

void main()
{
    int i;

    i = 11;

    char str1[i] = "abcdefghij";
    int str2[i] = {1,0,1,0,1,0,1,0,1,0};

    for (i=0; i<10; i++)
        str1[i] += str2[i];

    cout << str1;
}
```

מה תדפיס התכנית:

- |                      |     |
|----------------------|-----|
| abcdefghij           | .א. |
| aacceeggii           | .ב. |
| bbddffhhjj           | .ג. |
| AbCdEfGhIj           | .ד. |
| אף תשובה אינה נכונה. | .ה. |

ב ה צ ל ח ה !

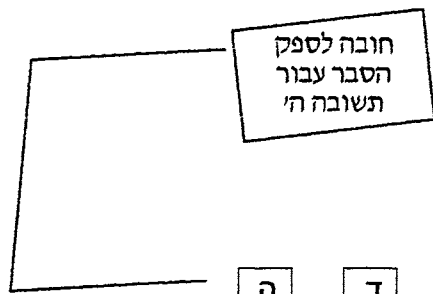
סמסטר א' תשס"א  
מועד: א' 27/2/2000  
משך הבחינה: 3 שעות  
חומר עזר: שני דפי עזר

בחינה בקורס: תכנות  
מרצה: איריס גבר

הנחיות כלליות לבחינה:

- המבחן מורכב משני חלקים:  
חלק א' כולל שאלות תכנות שמשקלן הכולל 65 נק'.
- חלק ב' כולל 5 שאלות אמריקאיות. משקל כל שאלה 8 נקודות, סה"כ: 40 נק'.
- חובה לתעד בשאלות התכנות כל פעולה לא ברורה שנעשית.
- בשאלות האמריקאיות יש לסמן תשובה אחת לכל שאלה בטבלה המצורפת. במידה ומספר תשובות נראות נכונות יש לסמן את התשובה הנכונה ביותר.
- בסיום המבחן יש לרשום מס' ת.ז. על גבי טבלת התשובות האמריקאיות ולהגישה בתוך מחברת הבחינה.
- נא לכתוב בכתב קריא ולא מחובר.

**בהצלחה !**



ת.ז.:

טבלת תשובות לחלק האמריקאי

ה	ד	ג	ב	א	
					שאלה 3
					שאלה 4
					שאלה 5
					שאלה 6
					שאלה 7

הערות לתשובות

אמנם רק תשובה נכונה תזכה בניקוד עבור כל שאלה, אולם ניתן לצרף לכל תשובה אמריקאית הסבר קצר. ההסבר לא יבדק במסגרת הבדיקה הרגילה, אך ניתן יהיה להסתמך עליו במסגרת ערעור, אם ידרש. מומלץ לצרף הסבר לתשובה אמריקאית במיוחד במקרים קיצוניים בהם נראה לך שתשובתך נכונה, אך נראה לך שהיא איננה התשובה שאליה התכוון המרצה. מובן שהסבר שגוי או בלתי סביר לא יועיל בכל מקרה (אך גם לא יזיק, אם ממילא סימנת את התשובה הנכונה).

חובה לצרף הסבר אם בחרת בתשובה ה

שאלה 3: \_\_\_\_\_

שאלה 4: \_\_\_\_\_

שאלה 5: \_\_\_\_\_

שאלה 6: \_\_\_\_\_

שאלה 7: \_\_\_\_\_



שאלה 1 (25 נק')

א. כתוב פונקציה המקבלת שני ארגומנטים מספריים. הפרמטר הראשון יכול להיות מספר חיובי או שלילי. הפונקציה תחזיר 'Y' אם הוא חיובי, ו-'N' אם הוא שלילי. בנוסף, מי שיפעיל את הפונקציה, יוכל לקבל, בארגומנט השני, בכל מקרה, את ערכו המוחלט של הארגומנט הראשון.

ב. הנח שברשותך הפונקציה מסעיף א' ופונקציה נוספת בשם IsPrime. הפונקציה הנ"ל מקבלת מספר שלם ומחזירה 1 אם הוא ראשוני ו-1 אם הוא אינו ראשוני. כתוב פונקציה שנעזרת בשתי הפונקציות הללו, בשם AbsPrime. הפונקציה הזו תקבל מספר, ותחזיר 1 אם ערכו המוחלט של המספר ראשוני, ו-0 אם ערכו המוחלט של המספר אינו ראשוני.

ג. כתוב תכנית שלמה בהיעזר בפונקציות הללו. התכנית תקלוט מספרים, ועבור כל אחד מהם תדפיס "The Absolute value is Prime" - אם ערכו המוחלט של המספר ראשוני, ו-"The Absolute value is not Prime" - אם ערכו המוחלט של המספר אינו ראשוני. התכנית תעשה זאת עד שיתקבל המספר 0 - ואז תפסיק את התהליך.

דוגמא לריצת התכנית

```
13          /* דוגמא לקלט שמכניס המשתמש */
The Absolute value is Prime
10
The Absolute value is not Prime
-11
The Absolute value is Prime
-22
The Absolute value is not Prime
0
```

שאלה 2 (40 נק')

עליך לכתוב תכנית המתחזקת מערך של תלמידים. תלמיד מיוצג ע"י:

- שמו (האורך אינו ידוע, אך לא יעלה על 100 תווים).
- תעודת הזהות שלו (9 ספרות).
- ממוצע ציוניו (מספר לאו דווקא שלם בין 0-100).

התכנית מתחילה בכך שהיא שואלת את המשתמש בכמה תלמידים עליה לטפל. לאחר מכן תציג התכנית תפריט שיכלול את האפשרויות הבאות:

1. קליטת נתוני התלמידים. עבור כל תלמיד יישאל המשתמש מהו אורך שמו ולאחר מכן יתבקש להכניס את השם, תעודת הזהות וממוצע הציונים.
2. הדפסת נתוני התלמידים.
3. הדפסת התלמיד בעל ממוצע הציונים הגבוה ביותר.
4. קבלת שם של תלמיד והדפסת נתוני האישיים (ת.ז. וממוצע ציונים).
5. סיום התכנית.

### הנחיות:

1. עליך לכתוב מספר של פונקציות עזר.
2. פונקציית ה-main אמורה להיות קצרה (לא יותר מ-15 שורות), ולהכיל תפריט למשתמש המפרט את ארבע האופציות העומדות לרשותו, המשתמש יקיש מספר בין 1-5 בהתאמה מול סוג הפעולה שברצונו לבצע.
3. חשוב היטב על מבנה הנתונים שתבחר!
4. ניתן להניח שאין תקלות (למשל, אין סיכוי שבאפשרות 4 בתפריט יתקבל שם של תלמיד שלא נמצא כלל במערכת).

### שאלה 3

נתונה התכנית הבאה:

```
int M(int x, int y)
{
    return x/y;
}

int N(int x)
{
    return (x) + (x/2);
}

void main( )
{
    int n=3;
    int m=5;

    float res=0;

    res = N(M(m+n,m));
}
```

מה ערכו של res בתום התכנית?

- א. 3.0
- ב. 1.5
- ג. 2.4
- ד. 1.0
- ה. אף תשובה אינה נכונה

#### שאלה 4

מה תדפיס התכנית הבאה:

```
# include <iostream.h>

int MyFunc (int &y);

void main()
{
    int x=0;
    int i=0;

    x = MyFunc(i);
    cout << x << ' ' << i;
}

int MyFunc (int &y)
{

    y++;
    return x+y;
}
```

- א. 1 1
- ב. 2 1
- ג. 1 0
- ד. 2 0
- ה. אף תשובה אינה נכונה

#### שאלה 5

מה יכלול המערך merged לאחר קריאה לפונקציה merge עם המערכים sorted1 ו-sorted2 הבאים:

```
int sorted1[] = {1, 1, 8, 17, 0};
int sorted2[] = {3, 7, 8, 12, 55, 0};
int merged[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

הקוד עצמו נמצא בעמוד הבא:

```

// merge two arrays of positive integers
void merge (int sorted1[], int sorted2[], int merged[])
{
    while (*sorted1 > 0 && *sorted2 > 0)
    {
        if (*sorted1 < *sorted2)
            *merged = *sorted1;
        else
            *merged = *sorted2;

        while (*sorted1 == *merged)
            ++sorted1;
        while (*sorted2 == *merged)
            ++sorted2;

        ++merged;
    }

    while (*sorted1 > 0)
    {
        *merged = *sorted1;

        while (*sorted1 == *merged)
            ++sorted1;

        ++merged;
    }

    while (*sorted2 > 0)
    {
        *merged = *sorted2;

        while (*sorted2 == *merged)
            ++sorted2;

        ++merged;
    }

    *merged = 0;

    return;
}

```

א. {1, 1, 3, 7, 8, 8, 12, 17, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
   
 ב. {1, 3, 7, 8, 12, 17, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
   
 ג. {1, 1, 3, 7, 8, 12, 17, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
   
 ד. {1, 1, 8, 17, 3, 7, 8, 12, 55, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
   
 ה. אף תשובה אינה נכונה.

## שאלה 6

מה הערך המוחזר ע"י הפונקציה operation?

```
int operation (int a, int b)
{
    if (b < 0)
        return - operation (a, b);
    else if (b == 0)
        return 0;
    else
        return a + operation (a, b-1);
}
```

- א. הערך של a.
- ב. הערך של b.
- ג. הסכום של a ו-b.
- ד. המכפלה של a ו-b.
- ה. אף תשובה אינה נכונה.

## שאלה 7

מה ידפיס קטע הקוד הבא:

```
char abc[26] = {'a','b','c','d','e','f','g','h','i','j','k',
               'l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
char key[26] = {'a','u','w','u','e','f','t','o','i','j','k','m','l','c',
               'd','y','q','p','h','g','v','o','z','s','r','b'};
char str1[] = "margey";
char str2[20];

int i;

for (i=0; str1[i]; ++i)
{
    str2[i] = key[str1[i] - 'a'];
}

str2[i] = 0;

cout << str2;
```

- א. mangle
- ב. laptop
- ג. gentle
- ד. lapter
- ה. אף תשובה אינה נכונה.

## מבחן בתכנות להנדסה מרצה: ברק שהם, מועד א'

(תעשיה)

משך הבחינה: 3 שעות.

- מותר שימוש בדף עזר בלבד.
- יש לענות על כל השאלות.
- יש לכתוב כל שאלה בעמוד חדש.
- דפי טיוטא יהיו בסוף המחברת ויסומנו בבירור ככאלה.
- התשובות לכל השאלות ב - ++C.
- יש לכתוב עם עימוד מתאים.
- אין להשתמש בפונקציות ספרייה.
- יש לשים לב אם בשאלה נדרש לכתוב פונקציה או תוכנית.

1. (25%) נקודה במישור,  $P$ , מוגדרת ע"י ערכי  $x$  ו- $y$  שלה. ישר,  $L$ , מוגדר ע"י  $L: y = ax + b$ , כאשר  $a, b$  קבועים.

א. (4%) הגדר מבנה מתאים לייצוג נקודה במישור. הגדר מבנה מתאים לייצוג ישר במישור.

ב. (10%) כתוב פונקציה `Intersect` המקבלת שני ישרים כפרמטרים ובודקת האם הם נחתכים. הפונקציה תחזיר ערך 1 אם הישרים נחתכים ו-0 אחרת.

ג. (11%) כתוב פונקציה `Get_Intersection` המחשבת את נקודת החיתוך של שני ישרים. הפונקציה תקבל כפרמטרים שני ישרים ותחזיר את נקודת החיתוך. הנח כי הישרים נחתכים.

2. (30%)

א. (15%) בהתאמת מחרוזות מחפשים מופע של מחרוזת אחת (התבנית) בתוך השניה (הטקסט). לעתים מחפשים התאמה לא מושלמת, שיש בה מספר סתירות - מקומות בהן התבנית שונה מהטקסט. לדוגמא, אם התבנית היא `abac` והטקסט `aabccc` הרי במקום הראשון קימת התאמה עם שתי סתירות (התו השני שונה - `a` בטקסט, `b` בתבנית, והתו השלישי שונה `b` בטקסט, `a` בתבנית). לעומת זאת במקום השני יש התאמה עם סתירה אחת (התו השלישי שונה - `c` בטקסט, `a` בתבנית). זוהי גם ההתאמה הכי "טובה" - יש בה מספר מינימלי של סתירות. כתוב פונקציה `match` המקבלת שתי מחרוזות - תבנית וטקסט, ומחשבת את ההתאמה הטובה ביותר. הפונקציה מחזירה את מספר הסתירות בהתאמה זו.

ב. (15%) כתוב פונקציה  $\text{int contained}(\text{char } s1[], \text{char } s2[])$  המקבלת כפרמטרים שתי מחרוזות  $s1, s2$  ומחזירה 1 אם כל תו שמופיע ב- $s1$  (לא חשוב כמה פעמים) מופיע גם ב- $s2$  (לא חשוב כמה פעמים). הפונקציה תחזיר 0 אם קיים ב- $s1$  לפחות תו אחד שלא מופיע ב- $s2$ .

3. (25%)  
ניתן ליצג פולינום באופן יחיד לפי השיטה הבאה:  
כל אחד מאיברי פולינום עם מקדם שונה מאפס מתואר ע"י זוג מספרים. זוגות המספרים מופיעים לפי סדר עולה של החזקות, לדוגמה:

$$p_1 = 3.4 + 2.1 x^3 - 15.3 x^{25} + 1.2 x^{50}$$

$p_1:$	3.4	2.1	-15.3	1.2	
	0	3	25	50	-1

נתון כי פולינום לא יכול יותר מעשרה איברים.  
סוף הפולינום ייוצג ע"י הערך 1- בחזקה.

א. (5%) הגדר טיפוס נתונים ליצוג איבר מאברי פולינום. הגדר משתנה היכול להכיל פולינום.

ב. (20%) כתוב פונקציה המקבלת שלושה פרמטרים  $p_1, p_2, p_3$  מסוג המשתנה שהוגדר בסעיף א'.  $P_1$  ו- $p_2$  מכילים פולינום בו האיברים מופיעים לפי סדר עולה של החזקות. הפונקציה תכתוב ב- $p_3$  את סכום הפולינומים ב- $p_1$  ו- $p_2$  תוך מעבר אחד על  $p_1$  ו- $p_2$ . שים לב שיצוג סכום הפולינומים ב- $p_3$  חייב גם הוא להיות מסודר לפי סדר עולה של החזקות.

(20%) כתוב פונקציות המחשבות את  $f(n)$  ו- $g(n)$

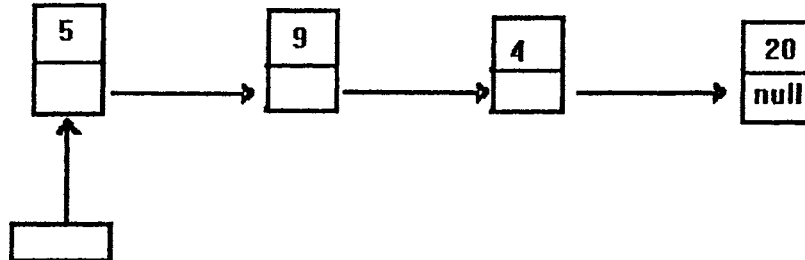
$$f(x) = \begin{cases} f(x-1) + g(x-2) * f(x-2) & x > 5 \\ g(x+5) & 0 < x \leq 5 \\ 7 & x \leq 0 \end{cases}$$

$$g(x) = \begin{cases} g(x-1) + g(x-3) & x \geq 3 \\ 2 & x < 3 \end{cases}$$

**בהצלחה !!**

3

4. (25%) נתונה רשימה מקושרת המכילה מספרים שלמים. כמו כן נתון מצביע לתחילת הרשימה. דוגמא:



- א. (5%) הגדר מבנה נתונים מתאים עבור איבר ברשימה. הגדר משתנה אשר יוכל להצביע לתחילת הרשימה.
- ב. (10%) כתוב פונקציה המקבלת כפרמטר מצביע לתחילת הרשימה. הפונקציה תדפיס את הערך הממוצע והשונות של אברי הרשימה.
- ג. (10%) כתוב פונקציה המקבלת שני ארגומנטים:  
- מצביע לתחילת הרשימה  
- מספר שלם - average - מציין את ממוצע ערכי האברים ברשימה.  
הפונקציה תחזיר מצביע לאיבר הראשון שערכו גדול מהממוצע אם הוא קיים ו null אחרת.

**בהצלחה !!**



## מבחן בתכנות להנדסה מרצה: ברק שהם, מועד א'

(חשמל)

משך הבחינה: 3 שעות.

- מותר שימוש בדף עזר בלבד.
- יש לענות על כל השאלות.
- יש לכתוב כל שאלה בעמוד חדש.
- דפי טיוטא יהיו בסוף המחברת ויסומנו בבירור ככאלה.
- התשובות לכל השאלות ב - ++c.
- יש לכתוב עם עימוד מתאים.
- אין להשתמש בפונקציות ספרייה.
- יש לשים לב אם בשאלה נדרש לכתוב פונקציה או תוכנית.

1. (25%) נקודה במישור,  $P$ , מוגדרת ע"י ערכי  $x$  ו- $y$  שלה. ישר,  $L$ , מוגדר ע"י  $y = ax + b$ , כאשר  $a, b$  קבועים.

- א. (4%) הגדר מבנה מתאים לייצוג נקודה במישור. הגדר מבנה מתאים לייצוג קו במישור.
- ב. (10%) כתוב פונקציה `Intersect` המקבלת שני ישרים כפרמטרים ובודקת האם הם נחתכים. הפונקציה תחזיר ערך 1 אם הישרים נחתכים ו-0 אחרת.
- ג. (11%) כתוב פונקציה `Get_Intersection` המחשבת את נקודת החיתוך של שני ישרים. הפונקציה תקבל כפרמטרים שני ישרים ותחזיר את נקודת החיתוך. הנח כי הישרים נחתכים.

2. (25%)

- א. (15%) בהתאמת מחרוזות מחפשים מופע של מחרוזת אחת (התבנית) בתוך השניה (הטקסט). לעתים מחפשים התאמה לא מושלמת, שיש בה מספר סתירות - מקומות בהן התבנית שונה מהטקסט. לדוגמא, אם התבנית היא `abac` והטקסט `aabccc` הרי במקום הראשון קימת התאמה עם שתי סתירות (התו השני שונה - `a` בטקסט, `b` בתבנית, והתו השלישי שונה `b` בטקסט, `a` בתבנית). לעומת זאת במקום השני יש התאמה עם סתירה אחת (התו השלישי שונה - `c` בטקסט, `a` בתבנית). זוהי גם ההתאמה הכי "טובה" - יש בה מספר מינימלי של סתירות. כתוב פונקציה `match` המקבלת שתי מחרוזות - תבנית וטקסט, ומחשבת את ההתאמה הטובה ביותר. הפונקציה מחזירה את מספר הסתירות בהתאמה זו.

## 5

ב. (10%) כתוב פונקציה  $\text{int contained}(\text{char } s1[], \text{char } s2[]);$  המקבלת כפרמטרים שתי מחרוזות  $s1, s2$  ומחזירה 1 אם כל תו שמופיע ב- $s1$  (לא חשוב כמה פעמים) מופיע גם ב- $s2$  (לא חשוב כמה פעמים). הפונקציה תחזיר 0 אם קיים ב- $s1$  לפחות תו אחד שלא מופיע ב- $s2$ .

3. (25%) ניתן ליצג פולינום באופן יחיד לפי השיטה הבאה:  
כל אחד מאיברי פולינום עם מקדם שונה מאפס מתואר ע"י זוג מספרים. זוגות המספרים מופיעים לפי סדר עולה של החזקות, לדוגמה:

$$p_1 = 3.4 + 2.1 x^3 - 15.3 x^{25} + 1.2 x^{50}$$

$p_1$ :

3.4	2.1	-15.3	1.2	
0	3	25	50	-1

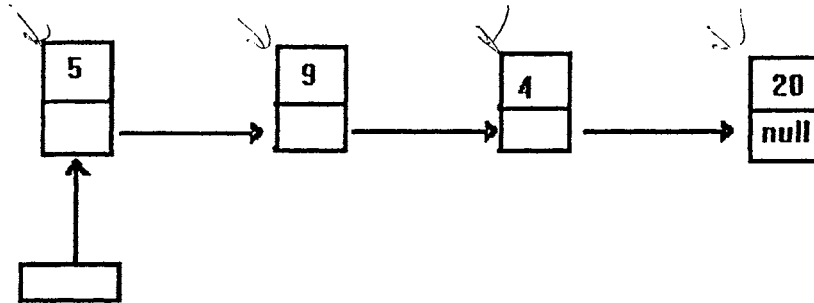
נתון כי פולינום לא יכיל יותר מעשרה איברים.  
סוף הפולינום ייוצג ע"י הערך 1- בחזקה.

א. (5%) הגדר טיפוס נתונים ליצוג איבר מאברי פולינום. הגדר משתנה היכול להכיל פולינום.

ב. (20%) כתוב פונקציה המקבלת שלושה פרמטרים  $p_1, p_2, p_3$  מסוג המשתנה שהוגדר בסעיף א'.  $P_1$  ו- $p_2$  מכילים פולינום בו האיברים מופיעים לפי סדר עולה של החזקות. הפונקציה תכתוב ב- $p_3$  את סכום הפולינומים ב- $p_1$  ו- $p_2$  תוך מעבר אחד על  $p_1$  ו- $p_2$ . שים לב שיצוג סכום הפולינומים ב- $p_3$  חייב גם הוא להיות מסודר לפי סדר עולה של החזקות.

6

4. (25%) נתונה רשימה מקושרת המכילה מספרים שלמים. כמו כן נתון מצביע לתחילת הרשימה. דוגמא:



- א. (5%) הגדר מבנה נתונים מתאים עבור איבר ברשימה. הגדר משתנה אשר יוכל להצביע לתחילת הרשימה.
- ב. (10%) כתוב פונקציה המקבלת כפרמטר מצביע לתחילת הרשימה. הפונקציה תדפיס את הערך הממוצע והשונויות של אברי הרשימה.
- ג. (10%) כתוב פונקציה המקבלת שני ארגומנטים:  
- מצביע לתחילת הרשימה  
- מספר שלם - average - מצין את ממוצע ערכי האברים ברשימה.  
הפונקציה תחזיר מצביע לאיבר הראשון שערכו גדול מהממוצע אם הוא קיים ו null אחרת.

**בהצלחה !!**

## מבחן בתכנות להנדסה

מרצה: ברוק שהם, מועד א'

משך הבחינה: 3 שעות.

- אין להשתמש בחומר עזר.
- יש לענות על כל השאלות.
- יש לכתוב כל שאלה בעמוד חדש.
- דפי טיוטא יהיו בסוף המחברת ויסומנו בבירור ככאלה.
- התשובות לכל השאלות ב - ++c.
- יש לכתוב עם עימוד מתאים.
- אין להשתמש בפונקציות ספריה.
- יש לשים לב אם בשאלה נדרש לכתוב פונקציה או תוכנית.

1. (25%) מספר ארמסטרונג הינו מספר שלם השווה לסכום ספרותיו מועלות בחזקה שלישית. כתוב פונקציה המקבלת מחרוזת המייצגת מספר שלם. הפונקציה תחזיר ערך 1 במידה והמספר המיוצג ע"י המחרוזת הינו מספר ארמסטרונג, 0 אחרת. ניתן להניח כי המחרוזת חוקית וכי מופיעים בה רק תווי ספרות.

דוגמא:

$$153 = 1^3 + 5^3 + 3^3 \quad \text{כי } \text{armstrong}("153") \text{ תחזיר ערך } 1$$

$$17! = 1^3 + 7^3 \quad \text{כי } \text{armstrong}("17") \text{ תחזיר ערך } 0$$

2. (25%) כתוב פונקציות המחשבות את  $f(n) - 1$  ו-  $g(n)$

$$f(x) = \begin{cases} f(x-1) + g(x-2) * f(x-2) & x > 5 \\ g(x+5) & 0 < x \leq 5 \\ 7 & x \leq 0 \end{cases} \quad g(x) = \begin{cases} g(x-1) + g(x-3) & x \geq 3 \\ 2 & x < 3 \end{cases}$$

3.

א. (15%) כתוב פונקציה `void rotate (int array[], int size, int n)` המקבלת מערך `array` בגודל `size` ומסובבת את אברי המערך `n` מקומות שמאלה.

דוגמא:

קלט:

array = 

1	5	2	3	4	7
---	---	---	---	---	---

 size = 6 n = 2

פלט:

array = 

2	3	4	7	1	5
---	---	---	---	---	---

8

קלט:

array = 

3	4	1	7
---	---	---	---

 size = 4 n = 1

פלט:

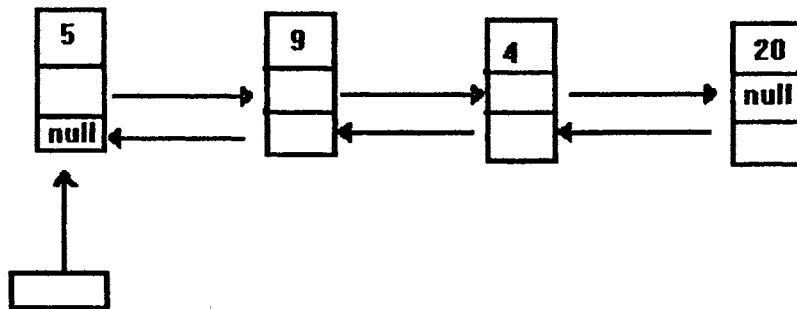
array = 

4	1	7	3
---	---	---	---

ב. (10%) כתוב פונקציה

`void replace (char str1[], char name1[], char name2[], char str2[])`  
המעתיקה את המחרוזת שב- `str1` ל- `str2` תוך החלפה של המופע הראשון של המחרוזת `name1` במחרוזת `name2`. ניתן להניח כי `str2` יוכל להכיל את מחרוזת הפלט.

4. נתונה רשימה דו-מקושרת המכילה מספרים שלמים. כמו כן נתון מצביע לתחילת הרשימה. דוגמא:



א. (5%) הגדר מבנה נתונים מתאים עבור איבר ברשימה. הגדר משתנה אשר יוכל להצביע לתחילת הרשימה.

ב. (10%) כתוב פונקציה המקבלת שני ארגומנטים: מצביע לתחילת הרשימה

- מספר שלם -  $n$

הפונקציה תדפיס את ערכו של האיבר ה- $n$  ברשימה אם הוא קיים והודעה "איבר לא קיים" אחרת.

דוגמא: בהפעלת הפונקציה על הרשימה לעיל עם  $n=3$  יודפס הערך 4.

ג. (10%) כתוב פונקציה המקבלת שני ארגומנטים:

- מצביע לתחילת הרשימה

- מספר שלם -  $p$

הפונקציה מחזירה מצביע לאיבר שערכו  $p$  אם הוא קיים ו-`null` אחרת.

דוגמא: בהפעלת הפונקציה על הרשימה לעיל עם  $p=20$  יוחזר מצביע לאיבר המכיל את הערך 20.

**בהצלחה !!**

**מבחן בתכנות להנדסה**  
**מרצה: ברק שהם, מועד א'**

**משך הבחינה: 3 שעות.**

- אין להשתמש בחומר עזר.
- יש לענות על כל השאלות.
- יש לכתוב כל שאלה בעמוד חדש.
- דפי טיוטא יהיו בסוף המחברת ויסומנו בבירור ככאלה.
- התשובות לכל השאלות ב - ++c.
- יש לכתוב עם עימוד מתאים.
- אין להשתמש בפונקציות ספרייה.
- יש לשים לב אם בשאלה נדרש לכתוב פונקציה או תוכנית.

1. (25%) כתוב תוכנית הקוראת קלט תווי באורך לא מוגבל המורכב מספרות, התו '0' והתו '.'. התוכנית מחשבת את סכום המספרים בין סמני '0'. סוף הקלט מסומן עי'י '0'.

דוגמה:

קלט: 1 3 6 | 4 9 2 3 | 3 3 6 2 | .  
 פלט: 10 18 14

2. (25%) הרקע לחלק זה הוא סיפור מצדה (ולאו דווקא תואם במדויק את האמת ההיסטורית). כמסופר, החליטו מגיני מצדה להתאבד כדי לא ליפול בשבי. על כן, הם עמדו במעגל, והראשון הרג את החייל שלימינו והעביר את החרב לבא אחריו (מימין). התהליך נמשך עד אשר נותר רק חייל בודד, ועליו להתאבד. נניח כי במצדה היו  $N=30$  מגינים.

א. (15%) עליך לכתוב פונקציה `int FindLastLive();`

המבצעת סימולציה של התהליך בעזרת מערך בגודל  $N$  ומחזירה את מספרו של החייל האחרון שנשאר בחיים. החיילים ממוספרים מ-0 (החייל הראשון) ועד  $N-1$ . ערך אחד במקום ה- $j$  במערך יציין כי החייל ה- $j$  חי וערך אפס יציין כי הוא מת. החייל שמספרו אפס הוא החייל הראשון, הבא אחריו מספרו אחד וכן הלאה. החייל הבא אחרי החייל ה- $N$  הוא החייל הראשון. ניתן להשתמש בפונקציה `int FindNext(int live[], int x);` (אין צורך לכתוב אותה בסעיף זה) המקבלת כפרמטר מערך שלמים מאורך  $N$  בו ערך אחד באינדקס  $i$  מסמן כי החייל ה- $i$  חי, ואפס מסמן כי הוא מת. ומספר חייל. הפונקציה תחזיר את מספרו של החייל החי הבא. הפונקציה מודעת למעגליות הבעיה.

ב. (10%) יש לכתוב את פונקציה `int FindNext(int live[], int x);`

3. (20%)

א. (10%) כתוב פונקציה

```
void replace (char str1[], char name1[], char name2[], char str2[])
```

המעתיקה את המחרוזת שב- str1 ל str2 תוך החלפה של המופע הראשון של המחרוזת name1 במחרוזת name2. ניתן להניח כי str2 יוכל להכיל את מחרוזת הפלט.

ב. (10%) כתוב פונקציה

```
int contained (char s1[], char s2[]);
```

המקבלת כפרמטרים שתי מחרוזות s1, s2 ומחזירה 1 אם כל תו שמופיע ב-s1 (לא חשוב כמה פעמים) מופיע גם ב-s2 (לא חשוב כמה פעמים). הפונקציה תחזיר 0 אם קיים ב-s1 לפחות תו אחד שלא מופיע ב-s2.

דוגמא:

אם s1=aabba\*ca ו-s2=cab\*b יוחזר 1.  
ואם s1=aabbaa\*ca ו-s2=bab%aba&ba יוחזר 0.

4. (30%)

א. (15%) רשימה מעגלית היא רשימה בה כל האיברים מקושרים וכל איבר מצביע לאיבר אחד ומוצבע ע"י איבר אחד. נתונה רשימה מקושרת הבנויה מאיברים מהטיפוס הבא:

```
struct NODE {
  int num;
  struct NODE* next;
};
```

כתוב פונקציה המקבלת כפרמטר מצביע לאיבר ברשימה ומחזירה 1 אם הרשימה מעגלית ואפס אחרת.

ב. (15%) נתונה רשימה מקושרת הבנויה מאיברים מהטיפוס הבא:

```
struct NODE {
  int num;
  struct NODE* next;
};
```

כתוב פונקציה:

```
struct NODE* MakeList(int n, int data)
```

שיוצרת רשימה משורשרת המכילה n איברים כאשר בכל איבר ערכו של השדה num הוא data. הפונקציה תחזיר מצביע לתחילת הרשימה. ניתן להניח כי n חיובי.

#include <iostream.h>

#include <stdlib.h>

⇒ exit(0) בסוף

11

1

void calc (int n, int isfirst) ⇒

פונקציה קוראת בודקת

↓ סכום הספרות      ↓ האם קראו מספר

```

{
  char tmp;           ⇒ קראו מספר נוכחי
  do
  {
    cin.get(tmp);    ⇒ קראו מספר. הולאמר מיוחס ה- do
  }
  while (tmp == ' ') ⇒ בעיקר מקרים
    if (tmp == '0')
    {
      if (isfirst)   ⇒ האם הקראו מספר ה- '0'
        calc(0,0);  ⇒ הורגת הקראו
      else {
        cout << n << " ";
        calc(0, isfirst);
      }
    }
  else
  if (tmp == '.')
    cout << "\n";
  else
  if ((tmp <= '9') && (tmp >= '0')) { ⇒ האם קראו ספרות
    n = n + (tmp - '0'); ⇒ מתווסף הספרות
    calc(n, 0);          ⇒ אם הפעם
  }
  else {
    cout << "illegal input";
    exit(0);
  }
}
}

```

void main ()

```

{
  calc(0,1);
}

```

↓ קראו מספר      ⇒ קראו מספר, חסאב והקנסה



```

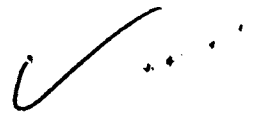
** const int N = 30;
void init (int arr[])
{
  int ind;
  for (ind=0; ind<N; ++ind)
    arr[ind] = 1;
}

```

12

פרוטוטיפ

(10)



```

int findnext (int live[], int x);
int findlastlive ()

```

⇒ prototype

```

{
  int ind=0, cnt, live[N];
  init (live);

```

⇒ הכנסת משתנים

```

  for (cnt=1; cnt<N; ++cnt)

```

⇒ איתחול משתן

⇒ צגן ש"היות" (N-1) אסמ

```

  {

```

```

    live [findnext (live, ind)] = 0;

```

⇒ הניבא

```

    ind = findnext (live, ind);

```

⇒ מציאת הנתון הבא

```

  }

```

```

  return ind;
}

```

מאתחל את ind

```

int findnext (int live[], int x)

```

```

{

```

```

  int ind;

```

```

  ind = (x+1) * N;

```

```

  while (live[ind] == 0)

```

```

  {
    ind = (ind+1) * N;

```

→

מקבלים מציאת כל המספרים  
 (29+1) = 30 → 0  
 (1+1) = 2 → 2

```

  }
  return ind;
}

```

(2)



③ 28KB

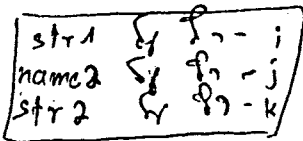
.k

```

int str_eq(char *s1, char *s2)
{
    for (; *s1 == *s2 && *s1 != '\0' && *s2 != '\0'; s1++ && s2++);
    if (*s2 == '\0') return 1;
    return 0;
}
    
```

```

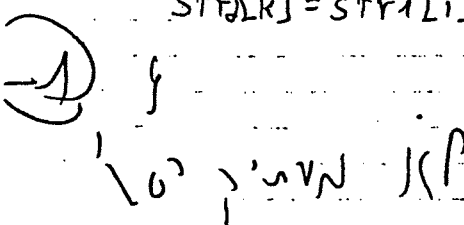
int str_inside(char *s, char *sub_s)
{
    for(int i=0; *s != '\0'; s++ && i++)
        if (str_eq(s, sub_s)) return (i);
    return (-1);
}
    
```



(i=j) - 0 of s1 - 0 s1 ...  
 (counter j = 0 (k)) str2 ← name2 ...  
 if name2 ...

```

void replace(char str1[], char name1[], char name2[], char str2[])
{
    int position = str_inside(str1, name1); i=0, j=0, k=0;
    for (; str1[i] != '\0'; i++ && k++)
        if (position == i) {
            for (; name2[j] != '\0'; j++ && k++)
                str2[k] = name2[j];
            for (j=0; name2[j] != '\0'; j++ && i++);
            str2[k] = str1[i];
        }
}
    
```



28KB

17

אם האות x קיימת ב-s אז תחזיר 1.  
 אחרת תחזיר 0.  
 אם האות x קיימת ב-s אז תחזיר 1.  
 אחרת תחזיר 0.

```
int check (char x, char s[])
{
  int in=0;
  for (int i=0; s[i]!='\0'; i++)
    if (x==s[i]) in=1;
  return in;
}
```

```
int contained (char s1[], char s2[])
{
  for (i=0; s1[i]!='\0'; i++)
    if (!check(s1[i], s2)) return 0;
  return 1;
}
```

```
void checkerror (struct NODE *tmp)
{
    if (tmp == NULL)
    {
        cout << "error: not enough memory";
        exit(0);
    }
}
```

פונקציה ש  
בדוקת האם הקצויה פשוט new - ה  
יש לה מפתח

```
struct NODE * makeList (int n, int data)
{
    int ind;
    struct NODE * head;
    struct NODE * tmp;
    head = new struct NODE;
    checkerror(head);
    head -> num = data;
    tmp = head;
    for (ind = 2; ind <= n; ++ind)
    {
        tmp -> next = new struct NODE;
        tmp = tmp -> next;
        checkerror(tmp);
        tmp -> num = data;
    }
    tmp -> next = NULL;
    return head;
}
```

הגדרת המשתנים  
זהו כאן  
לשינוי המיקום  
הוא יחזור  
המשתנה הוא יחזור



לפי זה

```
int isCirc (struct NODE * head)
{
    struct NODE * tmp;
    if (head != NULL)
    {
        tmp = head -> next;
        while (tmp != NULL)
        {
            if (tmp == head)
                return 1;
        }
        return 0;
    }
    else
        return 0;
}
```

הוא קיים למטה

הוא זה tmp שגודל האתר הוא head - נקבע  
1/8  
-> head <= 1/8  
-> head <= 1/8



מבחן בתכנות להנדסה

מרצה: ברק שהם. מועד ב'.

משך הבחינה: 3 שעות.

- אין להשתמש בחומר עזר.
- יש לענות על כל השאלות.
- ניקוד כל שאלה - 25 נקודות.
- יש לכתוב כל שאלה בעמוד חדש.
- דפי טיוטא יהיו בסוף המחברת ויסומנו בבירור ככאלה.
- התשובות לכל השאלות ב - ++c
- יש לכתוב עם עימוד מתאים.
- אין להשתמש בפונקציות ספרייה.
- יש לשים לב אם בשאלה נדרש לכתוב פונקציה או תוכנית.

1. כתוב תוכנית הקוראת קלט תווי באורך לא מוגבל ומדפיסה את כל השלשות הפולינדרומיות. כלומר שלשות אשר קריאתן מימין לשמאל ומשמאל לימין זהה. סוף הקלט מסומן ע"י '!'.

דוגמה:

קלט: abcdcb\*b+++/.

פלט: cdc

b\*b

+++

2. כתוב תוכנית הקוראת סדרה של מספרים שלמים שאורכה קטן מאלף ומדפיסה את המספרים הגדולים מהמוצע ואת ממוצע חמשת המספרים הקטנים בסדרה.

3. עבור  $n$  טבעי (שלם לא שלילי) נגדיר את הפונקציות  $f(n) = 2^n$  ו- $g(n) = \lceil \log_2 n \rceil$  כאשר  $x$  אינו התקרה של  $x$ , כלומר המספר השלם המקיים  $x \leq y < x+1$ . לדוגמא  $5.1 = 6, 8 = 8$ .

- א. ממש את  $f$  באופן איטרטיבי.
- ב. ממש את  $g$  באופן איטרטיבי.
- ג. ממש את  $f$  באופן רקורסיבי.
- ד. ממש את  $g$  באופן רקורסיבי.

$$\log_2 8 = 3$$

$$\log_2 16 = 4$$

$$\log_2 \text{Page 238} = 8$$

$$2^n = a$$

$$\log_2 a = n$$

א. כתוב פונקציה `int squeeze (char s[ ], char t)`. המחזירה את מספר המופעים של התו `t` במחרוזת `s`. הפונקציה תמחק את כל מופעי התו `t` מהמחרוזת `s`.  
 דוגמה: `s = "abcdcb*txcy"`  
`t = 'c'`

הפונקציה תחזיר ערך 3. לאחר ביצוע הפונקציה: `s = "abdb*txy"`

ב. כתוב פונקציה `intersection` המקבלת שלושה פרמטרים מסוג מחרוזת `s1`, `s2`, `s3`. הפונקציה מחשבת את החיתוך של מחרוזות `s1` ו-`s2`, כלומר התווים המשותפים לשניהם. תווי החיתוך מוחזרים במחרוזת `s3` (פרמטר זה הינו "ריק" בכניסה לפונקציה). ניתן להניח שתו מסוים מופיע לכל היותר פעם אחת בכל מחרוזת. כמו כן, ניתן להניח שמחרוזת `s3` הינה בגודל מספיק להכיל את החיתוך.

**בהצלחה !!**



סמסטר א' תשנ"ח  
בחינת מעבר מועד א'  
מועד הבחינה: 16.2.98  
משך הבחינה: 3 שעות

מבחן בקורס "תכנות להנדסה מכנית"

זהר נאור

- אין להשתמש בחומר עזר
- יש לענות על כל השאלות
- יש לענות על כל שאלה בעמוד חדש.
- התשובות לכל השאלות ב C++
- אין להשתמש בפונקציות ספרייה
- בהצלחה!

שאלה 1 (35 נקודות)

כתוב פונקציה `Void merge (int n, int a[], int b[], int m[])` המקבלת שני מערכים ממויינים a ו-b, אשר כל אחד מהם מכיל n מספרים ממויינים בסדר עולה, (כלומר - a[0] הוא האיבר הקטן ביותר במערך a, ו-b[0] הוא האיבר הקטן ביותר במערך b), מספר טבעי n, ומערך m.

הפונקציה merge ממזגת את איברי המערכים a ו-b לתוך המערך m, כך ש-m יכול את כל  $2n$  האיברים כשהם ממויינים בסדר עולה. הפונקציה מניחה כי המערך m הוא ריק ויש בו  $2n$  מקומות. הפתרון חייב להיות יעיל, כלומר: לעשות שימוש בעובדה שהמערכים a ו-b הם ממויינים.

שאלה 2 (35 נקודות)

כתוב פונקציה `int prefix (char *what, char *of)` הבודקת האם המחרוזת what הינה תחילית (רישא) של המחרוזת of. דוגמאות:

"" (המחרוזת הריקה) היא תחילית של כל מחרוזת.

"fat" היא תחילית של "father".

"mother fu" היא תחילית של "mother function".

"father" היא תחילית של "father".

"enough" איננה תחילית של "enum".

הפונקציה Prefix צריכה להחזיר את המספר 1 אם המחרוזת what היא תחילית של המחרוזת of ואת המספר 0 אם היא איננה תחילית של of.

א. (20 נקודות). ממש את הפונקציה Prefix באופן איטרטיבי.

ב. (15 נקודות). ממש את הפונקציה Prefix באופן רקורסיבי.

### שאלה 3 (30 נקודות)

עבור  $n$  טבעי (שלם לא שלילי) נגדיר את הפונקציות  $f(n) = 2^n$ ,  $g(n) =$  התקרה של  $\log_2 n$ . כלומר  $g(n)$  היא המספר השלם  $y$  המקיים:  $y - 1 < \log_2 n \leq y$ . לדוגמא התקרה של 8 היא 8, התקרה של 5.1 היא 6.

א. (5 נקודות). ממש את  $f$  באופן איטרטיבי.

ב. (10 נקודות). ממש את  $f$  באופן רקורסיבי.

ג. (5 נקודות). ממש את  $g$  באופן איטרטיבי.

ד. (10 נקודות). ממש את  $g$  באופן רקורסיבי.





מס' ת.ג. 15799999

סמסטר א' תשנ"ז  
 בחינת מעבר מועד א'  
 מועד הבחינה: 30.1.97  
 משך הבחינה: 3 שעות

**בחינה בקורס "תכנות להנדסה מכנית"**

זהר טאור

- אין להשתמש בחומר עזר
- יש לענות על כל השאלות
- יש לענות על כל שאלה בעמוד חדש
- אין לענות על שאלה ביותר מעמוד אחד, רק העמוד הראשון של כל שאלה יבדק.
- התשובות לכל השאלות ב ++C
- אין להשתמש בפונקציות ספריה
- בהצלחה!

**שאלה 1 (33 נקודות)**

כתוב תוכנית הקוראת סדרה בת לכל היותר 100 מספרים שלמים וחיוביים, מספר אחד בכל פעם. המספר 1- (אשר איננו שייך לסדרה) מסמן את סוף הקלט. על התוכנית להדפיס את 5 המספרים הגדולים ביותר בסדרה.

**שאלה 2 (33 נקודות)**

כתוב תוכנית הקוראת מספר בלתי מוגבל של מספרים שלמים השונים מ-0, מספר אחד בכל פעם. המספר 0 (אשר איננו חלק מהסדרה) מסמן את סוף הקלט. הסדרה מכילה לפחות 3 מספרים. על התוכנית להדפיס את כל המקסימומים המקומיים (מקסימום מקומי הוא מספר הגדול גם מהקודם לו וגם מהבא אחריו). לדוגמא:

קלט: 1 2 3 1 7 5 4 8 3 0  
 פלט: 3 7 8

**שאלה 3 (34 נקודות)**

א. (17 נקודות). כתוב פונקציה רקורסיבית `strcpy` המקבלת כמשתנים מועברים שתי מחרוזות: `s` ו-`t`, ומעתיקה את המחרוזת `s` לתוך המחרוזת `t`. אורך המחרוזת `s` איננו ידוע מראש, אולם הפונקציה מניחה כי במחרוזת `t` יש מספיק מקום להכיל את `s`.

ב. (17 נקודות). כתוב פונקציה רקורסיבית `strlen` המקבלת מחרוזת ומחשבת את אורכה. הערה: אם אינך מצליח לכתוב פונקציות רקורסיביות, ניתן לכתוב פתרון איטרטיבי, אולם רק פתרונות רקורסיביים יקבלו את מלוא הנקודות!

בהצלחה!

```
#include <iostream.h>
```

```
void sort (int array[], int n);
```

```
void main ()
```

```
{
```

```
int number, i, j;
```

```
int array[100];
```

```
cout << "Enter up to 100 integer numbers ending with -1\n";
```

```
for (i=0; i<100; i++)
```

```
{
```

```
cin >> number;
```

```
if (number < 0) break;
```

```
array[i] = number;
```

```
}
```

```
sort (array, i);
```

```
cout << array[i] << array[i-1] << array[i-2] << array[i-3] << array[i-4];
```

```
}
```

```
void sort (int array[], int n)
```

```
{ int i, j, temp;
```

```
for (i=0; i<=n; i++)
```

```
{ for (j=n; j>i; j--)
```

```
if (array[j] < array[j-1])
```

```
{
```

```
temp = array[j];
```

```
array[j] = array[j-1];
```

```
array[j-1] = temp;
```

```
}
```

33  
33

27

22

<u>main</u>															
a=11	<u>max</u>	<u>max</u>													
b=2	a=1	max		<u>max</u>											
	b=2	a=2	<u>max</u>												
	c=3	b=3	a=3	<u>max</u>											
		c=1	b=1	a=1	<u>max</u>										
			c=2	b=2	a=2	<u>max</u>									
				c=3	b=3	a=3	<u>max</u>								
					c=4	b=4	a=4	<u>max</u>							
						c=8	b=8	a=8	<u>max</u>						
							c=3	b=3	a=3	<u>max</u>					
								c=3	b=3	a=3	<u>max</u>				
									c=3	b=3	a=3	<u>max</u>			
										c=3	b=3	a=3	<u>max</u>		
											c=3	b=3	a=3	<u>max</u>	

```
#include <iostream.h>
```

```
void main ()
```

```
{
```

```
int a, b, c,
```

```
cin >> a >> b >> c,
```

```
while (c != 0)
```

```
{
```

```
if (b > a && b > c)
```

```
cout << b;
```

```
c = b;
```

```
b = c;
```

```
cin >> c;
```

```
}
```

```
}
```

! 0 0 ; ✓

33/33

89

void strcpy (char \*s, char \*t) <sup>24</sup>

{

if (\*s)

{ \*t = \*s;

~~return (strcpy (s++, t++))~~

}

else

\*t = '\0';

}

17/77

int strlen (char \*string) 25

27

```
{
    if (*string)
        return (strlen(string++) + 1);
    else
        return (0);
}
```

Handwritten notes in Arabic:   
- Above the first 'string' parameter: "نوع الدالة هو string" (The type of the function is string)   
- Above the second 'string' parameter: "< string, char >"   
- Next to the recursive call: "نوع الدالة هو strlen" (The type of the function is strlen)   
- Next to the return statement: "نوع الدالة هو int" (The type of the function is int)

17/17