מס' מחברת : מס' ת.ז. :

סמסטר א' תש"ע, מועד מיוחד
משך הבחינה: שלוש שעות
אין להשתמש בחומר עזר

בחינה בקורס תכנות ב- C

מרצים: אפרת משיח, מתי שמרת

מתרגלים: נוגה אורון-לוי, 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
- בסוף המבחן נמצא דף-עזר הכולל רשימה של פונקציות ספרייה.
- נא לכתוב בכתב קריא וברור.
- יש לרשום את מספר תעודת הזהות ואת מספר המחברת בראש כל 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	

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.

מס' מחברת : מס' ת.ז. :

שאלה מס' 1 (33 נקודות)

בשאלה זו נממש פונקציה פשוטה הדומה לפונקציה printf.
שימו לב: בכל סעיף בשאלה זו **חובה** להשתמש בפונקציה מסעיף קודם. תוכלו להשתמש בפונקציה גם אם לא מימשתם אותה. פתרון ללא שימוש בפונקציה מהסעיף הקודם לא יזכה ביותר ממחצית הנקודות של הסעיף.

סעיף א' (5 נקודות):

כתבו פונקציה בשם swapChars המחליפה את הערכים של שני משתנים מטיפוס char.
לדוגמא: אם קיימים בתכנית שני משתנים מטיפוס char בשמות letter1 ו-letter2 שערכיהם 'a' ו-'b' בהתאמה, אזי לאחר ביצוע הפונקציה ערכו של המשתנה letter1 יהיה 'b' בעוד שערכו של letter2 יהיה 'a'.

השלימו את חתימת הפונקציה וממשו את גוף הפונקציה.

```
void swapChars(_________________ c1, _________________ c2)
{
```

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' (8 נקודות):

ממשו את הפונקציה reverseString שהופכת את סדר תווים במחרוזת נתונה כך שהתו הראשון יתחלף עם התו האחרון, השני עם התו הלפני האחרון וכן הלאה. היפוך המחרוזת יתבצע במערך של המחרוזת המקורית (הורסת את המחרוזת המקורית).
לדוגמא הקוד הבא עושה שימוש בפונקציה המבוקשת:

```
char str[] = "abcd";
printf("%s\n", str);
reverseString(str);
printf("%s\n", str);
```

פלט התכנית הוא:

```
abcd
dcba
```

שימו לב: בסעיף זה עליכם להשתמש בפונקציה swapChars מסעיף קודם.

```
void reverseString(char *str)
{
```

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.

מס' מחברת : מס' ת.ז. :

סעיף ג' (8 נקודות):

ממשו פונקציה הממירה מספר שלם (int) למחרוזת. על הפונקציה לטפל במספרים חיוביים ושליליים.
הפונקציה מקבלת מספר שלם (int) ומערך של תווים str וממירה את המספר למחרוזת. במקרה של מספר שלילי הפונקציה תוסיף את התו '-' בתחילת המספר.
הפונקציה מחזירה את אורך המחרוזת שנכתבה לתוך str.

הניחו כי המערך המסופק כקלט לפונקציה גדול מספיק להכיל את התוצאה.

שימו לב: בסעיף זה עליכם להשתמש בפונקציה reverseString מסעיף קודם.
אין להשתמש בפונקצית הספרייה atoi לצורך מימוש הפונקציה.

```
int numToStr(int num, char str[])
{
```

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' (12 נקודות):

ממשו פונקציה בשם printToString הבונה מחרוזת באופן דומה ל printf. הפונקציה מקבלת שלושה פרמטרים כקלט:

str – מערך המיועד להכיל את תוצאת הפונקציה (הניחו כי גדול מספיק)

format– מחרוזת המתארת את פורמט התוצאה

params– מערך של מספרים שלמים שישמשו בעת בניית מחרוזת התוצאה

מחרוזת הפורמט מכילה את הטקסט שיועתק למחרוזת התוצאה. בנוסף היא מכילה תגים שיוחלפו בערכים ממערך הפרמטרים. הפונקציה אותה הינכם נדרשים לממש תתמוך בתג היחיד '%d'. בנוסף כדי להדפיס את התו '%' מחרוזת הפורמט תכיל את התגית '%%'.

בעת מעבר על מחרוזת הפורמט מתחילתה (התו במקום ה-0) יוחלף התג '%d' הראשון בערך הראשון ממערך הפרמטרים, התג '%d' השני יוחלף בערך השני ממערך הפרמטרים וכן הלאה. הניחו כי מערך הפרמטרים מכיל את מספר הערכים הנדרש במחרוזת הפורמט. כמו כן הניחו שמחרוזת הפורמט היא חוקית ואינה מכילה תגים שלא נתמכים ע"י הפונקציה.

דוגמאות:

- עבור מחרוזת הפורמט **"%d %% %d = %d\n"** ומערך פרמטרים המכיל את הערכים **{10,3,1}** תוחזר המחרוזת (במערך התוצאה str) - **"10 % 3 = 1\n"**
- עבור מחרוזת הפורמט **"%%%%"** תוחזר המחרוזת (במערך התוצאה str) - **"%%"**
- עבור מחרוזת הפורמט **"x = %d"** ומערך פרמטרים המכיל את הערך **{100}** תוחזר המחרוזת (במערך התוצאה str) - **"x = 100"**

שימו לב – בפתרון סעיף זה הינכם נדרשים להשתמש בפונקציה numToStr מהסעיף הקודם.
בנוסף, אין להשתמש בפונקציה sprintf או כל פונקציה אחרת ממשפחת printf בפתרון סעיף זה.

את הפתרון יש לכתוב במקום המיועד לכך בעמוד הבא.

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.

```
void printToStr(char *str, const char* format, int params[])
{
```

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.

שאלה 2 (33 נקודות)

בשאלה זו נממש מערכת לניהול ציוני סטודנטים.
נתון מבנה של סטודנט (Student) המכיל מספר זהות (id) וציון (grade).
כמו כן, נתון מבנה (GradesList) המחזיק רשימה של הסטודנטים ממויינים עפ"י הציון שלהם (בסדר יורד, מהציון הגבוה לנמוך). הרשימה מוחזקת במערך בגודל 1000 (students), מספר הסטודנטים ברשימה מאוכסן בשדה size והאינפורמציה על הסטודנטים האלה מאוכסנת בתאים 0 עד (size-1) במערך.

```
#define SIZE 1000

typedef struct student{
      int id;
      int grade;
} Student;

typedef struct gradeslist{
      int size;
      Student students[SIZE];
} GradesList;
```

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.

מס' מחברת : מס' ת.ז. :

סעיף א' (10 נקודות):

כתבו פונקציה שמקבלת כתובת של רשימת ציונים (grades_list), ציון (new_grade) ומספר זהות (new_id) של סטודנט חדש, ומוסיפה את הפרטים של הסטודנט החדש לרשימה, כך שהרשימה תשאר ממויינת עפ"י ציוני הסטודנטים (בסדר יורד, מהציון הגבוה לנמוך).

```
void addGrade(GradesList* grades_list, int new_grade, new id){
```

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 addBonus שמקבלת כתובת לרשימת ציונים (grades_list), אינדקס של סטודנט (index), ובונוס שיש להוסיף לציון הסטודנט (bonus). הפונקציה תוסיף את הבונוס לסטודנט שנמצא בתא students[index] ואם יש צורך, תתקן את סדר הסטודנטים במערך כך שהמערך ישאר ממויין על פי ציוני הסטודנטים.

```
void addBonus(GradesList* grades_list, int index, int bonus){
```

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' (5 נקודות):

הפונקציה findGrade מקבלת כתובת לרשימת ציונים (grades_list) ערך של ציון שברצוננו לחפש במערך (val) וטווח אינדקסים שבו נרצה לחפש את הציון (from_index, to_index). הפונקציה תחזיר אינדקס של תא במערך שמכיל סטודנט עם הציון המבוקש. אם לא קיים כזה סטודנט הפונקציה תחזיר 1- ואם קיים יותר מסטודנט אחד, הפונקציה תחזיר את האינדקס של אחד מהסטודנטים עם הציון המבוקש (לא חשוב איזה).
החיפוש במערך יעשה בשיטה של חיפוש בינארי במערך ממויין: אם לסטודנט שבאמצע הרשימה יש ציון גבוה מהציון המבוקש, נחפש במחצית הרשימה שאחריו, ואם ציונו נמוך מהציון המבוקש, נחפש במחצית הרשימה שלפניו. אם ציונו שווה לציון המבוקש, הרי שמצאנו את מבוקשנו.
השלימו את הפונקציה הרקורסיבית findGrade:

```
int findGrade(GradesList* grades_list, int val,
              int from_index, int to_index)
{
     int middle_index = [                    ]

     if ( [                                  ]){
          return middle_index;
     }
     if ( [                                  ]){
          return -1;
     }
     if ( [                                  ]){

          return findGrade(grades_list, val,
                           [          ], [          ]);
     }
     return findGrade(grades_list, val,
                      [          ], [          ]);
}
```

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.

מס' מחברת : מס' ת.ז. :

סעיף ד' (8 נקודות):

כתבו תוכנית שקולטת מהמשתמש רשימה של זוגות מספרים שמשמעותם: מספר זהות של סטודנט וציונו עד אשר יקלט זוג המספרים 1-,1-. התוכנית תכניס את כל הסטודנטים למבנה הנתונים `GradesList`. לאחר מכן התוכנית תחפש בזה אחר זה את כל הסטודנטים עם ציון 80 ותוסיף להם בונוס של 5 נקודות.
לבסוף, תדפיס התוכנית את מספרי הזהות והציונים של הסטודנטים, ממויינים עפ"י ציוניהם.
בסעיף זה חובה להשתמש בשלוש הפונקציות שכתבתם בסעיפים א'-ג'.

דוגמת הרצה:

קלט:	פלט:
1 90	1 90
2 80	4 85
3 82	5 85
4 85	2 85
5 80	3 82
6 70	6 70
-1 -1	

```
void main(){
```

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.

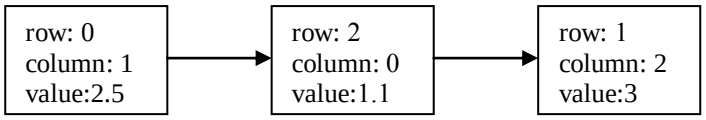
מס' מחברת : מס' ת.ז. :

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.

שאלה 3 (34 נקודות)

בשאלה זו הינכם נדרשים לממש מטריצה כרשימה מקושרת של התאים במטריצה. ברשימת התאים נחזיק **רק** את התאים במטריצה בהם קיים ערך שונה מ-0. כל אבר ברשימה ייצג תא במטריצה ויכיל את מיקומו (שורה ועמודה) וכן את ערכו.
לדוגמא, את המטריצה הבאה ניתן לייצג בעזרת רשימה מקושרת המורכבת מאברים המייצגים את תאים <0,1>, <1,2>,<2,0>.
מכיוון שתא מחזיק את מיקומו במטריצה אין משמעות לסדר התאים ברשימה.

	0	1	2
0	0	2.5	0
1	0	0	3
2	1.1	0	0



סעיף א' (5 נקודות):

הגדירו את המבנה Cell המייצג תא במטריצה, המבנה יכיל את מיקום התא במטריצה (שורה ועמודה) וכן את הערך השמור בתא (מספר ממשי)

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' (8 נקודות):

נתון המבנה Matrix המגדיר מטריצה.

```
typedef struct matrix
{
    int rows, columns;
    Cell* head;
} Matrix;
```

המבנה מכיל את מספר השורות והעמודות במטריצה וכן מצביע לראש רשימת התאים במטריצה.
שימו לב, רק תאים בעלי ערך יופיעו ברשימה.

a) (6 נקודות) ממשו את הפונקציה getCell המחזירה את התא שמיקומו במטריצה הוא <i, j>.
הפונקציה תחזיר NULL אם תא כזה אינו קיים ברשימה.

```
Cell* getCell(Matrix* m, int i, int j)
{
```

b) (2 נקודות) ממשו את הפונקציה getValue המחזירה את הערך של התא ה <i,j> או 0 עם התא לא קיים.

```
double getValue(Matrix* m, int i, int j)
{
```

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' (10 נקודות):

השלימו את מימוש הפונקציה setValue המשנה את ערכו של התא <i,j> במטריצה m כך שערכו החדש יהיה newValue. אם התא קיים כבר ברשימה הפונקציה תשנה את ערכו לערך החדש אחרת תוסיף תא חדש לרשימה. יש להקפיד כי לא יוכנסו תאים עם הערך 0 לרשימה.
הפונקציה `removeCell(Matrix* m, Cell* cell)` המופיעה בקוד מסירה את התא cell מהמטריצה m.

```
void setValue(Matrix* m, int i, int j, double newValue)
{
    Cell* new_cell = NULL;
    Cell* cell = getCell(m, i, j);

    if (newValue == 0) {     /* remove cell if new value is 0 */
        if (cell != NULL)    /* and a cell at i,j exists      */
            removeCell(m, cell);
        return;
    }

    if (cell == NULL)        /* a cell at i,j doesn't exist */
    {



    } else {     /* a cell at i,j exists */



    }
}
```

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' (10 נקודות):

ממשו את הפונקציה addMatrixes המחברת שתי מטריצות ומחזירה את תוצאת החיבור.
פעולת החיבור על מטריצות מוגדרת על מטריצות בעלות אותם מימדים. התוצאה היא מטריצה חדשה מאותם מימדים כמו מטריצות הקלט והערך בתא ה <i,j> במטריצת התוצאה הוא החיבור של ערכי התאים <i,j> במטריצות הקלט. כלומר $C_{i,j} = A_{i,j} + B_{i,j}$ כאשר A ו-B הן מטריצות הקלט ו-C מטריצת התוצאה.

שימו לב: עליכם לוודא את תקינות הקלט לפונקציה.

תוכלו לעשות שימוש בפונקציות מסעיפים קודמים וכן בפונקציה
`Matrix* newMatrix(int rows, int columns);`
העומדת לרשותכם. פונקציה זו מחזירה מטריצה חדשה ממימד rows × columns.

```
Matrix* addMatrices(Matrix* m1, Matrix* m2)
{
```

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.

מס' מחברת : מס' ת.ז. :

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 מהספריה 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 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 מהספריה math.h:

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

פונקציות נבחרות מהספריה ctype.h:

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

פונקציות נבחרות מהספריה stdlib.h:

1) הקצאת זיכרון דינאמית
```
void* malloc(unsigned int size)
```
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 מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.