

קורס תכנות

תרגיל מספר 13

מועד הגשה: בשבוע הבא, כמפורט באתר הקורס

הנחיות כלליות: קראו בעיון את השאלות והקפידו שהתכניות שלכם יפעלו בהתאם לנדרש. הקפידו על כללי ההגשה המפורסמים באתר.

שימו לב: את התרגיל יש לפתור לבד! הגשת תרגיל זה הינה חובה על מנת לגשת למבחן בקורס!

שאלה 1

בשאלה זו ננהל רשימת קניות. רשימת הקניות מכילה מוצרים שונים. לצורך השאלה נגדיר את המבנה מוצר (product) באופן הבא:

```
typedef struct product{
    char name[31];
    double quantity;
    double pricePerUnit;
    struct product * next;
} Product;
```

מוצר מכיל שם (name), כמות (ביחידות או בק"ג quantity) ומחיר ליחידה/ק"ג. כל מוצר הוא איבר בתוך רשימה מקושרת ממויינת (בסדר מילוני/לקסיקוגרפי עפ"י name). הרשימה המקושרת היא למעשה רשימת הקניות.

• כתבו פונקציה בעלת המפרט (prototype) הבא:
`Product* findProduct (Product * head, char * name, double pricePerUnit)`

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מוצרים, שממוינת לפי name (בסדר מילוני עולה), שם של מוצר ומחיר ליחידה. הפונקציה צריכה לבדוק אם המוצר קיים ברשימה (כלומר אם קיים מוצר עם שם זה ואותו מחיר ליחידה). אם כן, הפונקציה תחזיר מצביע אליו, אחרת הפונקציה תחזיר NULL. החיפוש יכול להיות סידרתי (מעבר רציף על הרשימה).

• כתבו פונקציה בעלת המפרט (prototype) הבא:
`Product* addProduct (Product * head, char* productName, double quantity, double pricePerUnit)`

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מוצרים, שממוינת לפי name (בסדר מילוני עולה), ופרטים של מוצר חדש. הפונקציה תחפש אם המוצר קיים ברשימה. אם כן, יש לעדכן את הכמות (להוסיף את הכמות החדשה שהתקבלה מהשתמש, למוצר שמצאנו ברשימה). אחרת, יש ליצור את המבנה המתאים למוצר החדש ולהוסיף אותו לרשימה תוך שמירה על סדר המיון. על הפונקציה להחזיר מצביע לתחילת הרשימה המעודכנת או NULL אם הפעולה נכשלה (במקרה זה יש לשחרר את הרשימה שכבר נוצרה). (במקרה שבו ישנו יותר ממוצר אחד בעל אותו שם עם מחירים שונים, אין חשיבות לסדר הפנימי ביניהם).

- כתבו פונקציה בעלת המפרט (prototype) הבא:

`double getTotalPrice (Product * head)`

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מוצרים. עליה להחזיר את המחיר הכולל של סל הקניות (שהוא סכום המחירים של כל המוצרים ברשימה. מחיר מוצר הינו מחיר ליחידה כפול מספר יחידות. אם הכמות היא בק"ג אזי מחיר לק"ג כפול כמות קילוגרמים). אם הרשימה ריקה, הפונקציה תחזיר 0.

- כתבו פונקציה בעלת המפרט (prototype) הבא:

`void freeList (Product * head)`

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת של מוצרים. עליה לשחרר את הזיכרון הדינמי שהוקצה עבור הרשימה.

- כתבו תוכנית (main) שמשמשת בפונקציות שכתבתם.

- התוכנית תקלוט בלולאה מוצרים מהמשתמש (ניתן להניח ששם מוצר אינו מכיל רווחים וכי מספר התווים המכסימלי בשם הוא 30).
- התוכנית תוסיף את המוצרים לרשימה (בהתאם להוקיות המפורטת ב-addProduct)
- תנאי העצירה של הלולאה הוא מוצר עם שדה כמות שלילי או מחיר שלילי (או שניהם) מוצר זה לא יכנס לרשימה.
- בסיום, התוכנית צריכה להדפיס את עלות סל הקניות.
- להזכירכם, יש לשחרר כל זיכרון שהוקצה באופן דינמי לאחר סיום השימוש בו או כאשר חלה שגיאה.

דוגמת הרצה (באדום פלט ובשחור קלט):

```

Enter product (in fmt : productName quantity pricePerUnit)
apple 3 1.2
Enter product (in fmt : productName quantity pricePerUnit)
tommatto 4 1.5
Enter product (in fmt : productName quantity pricePerUnit)
milk 2 5
Enter product (in fmt : productName quantity pricePerUnit)
water 2 4
Enter product (in fmt : productName quantity pricePerUnit)
milk 1 5
Enter product (in fmt : productName quantity pricePerUnit)
milk 1.5 6
Enter product (in fmt : productName quantity pricePerUnit)
a -1 7
The total price of our basket is 41.6

```

שאלה 2

בשאלה זו נקלוט רצף של תווים מהמשתמש ונייצג אותם באמצעות רשימה מקושרת (כאשר כל תו הוא איבר ברשימה המקושרת).

נגדיר על הרשימה פעולת מחיקה **בדילוגים בגודל k** : בפעולה זו נמחק **לאחר** כל k תווים את התו הבא מהרשימה. לדוג':

1. עבור המילה abcd , k=1 נקבל : ac
2. עבור המילה abcde , k=2 נקבל : abde
3. עבור המילה abcde , k=3 נקבל : abce

לצורך השאלה נגדיר חוליה ברשימה באופן הבא :

```
typedef struct node{
    char letter;
    struct node * next;
} Node;
```

- כתבו פונקציה בעלת המפרט (prototype) הבא:

Node* buildString()

הפונקציה קולטת רצף של תווים מהמשתמש (עד אשר מתקבל n), רצף התווים יכול להכיל כל תו אפשרי, כולל רווחים) ובונה עבורם רשימה מקושרת, כך שכל תו מתורגם לחוליה ברשימה. סדר החוליות ברשימה צריך להיות תואם לסדר התווים שהתקבלו. הפונקציה מחזירה מצביע לראש הרשימה או NULL אם אירעה שגיאה כלשהי (במקרה זה יש לשחרר את הצמתים שהוקצו עד כה).

- כתבו פונקציה בעלת המפרט (prototype) הבא:

Node* removeInJumps (Node * head, int k)

הפונקציה מקבלת כפרמטר מצביע לראש הרשימה וגודל קפיצה תקין. הפונקציה תחזיר מצביע לרשימה מעודכנת שמכילה מחרוזת המתקבלת לאחר מחיקה בדילוגים בגודל k. (k יכול להיות גדול מגודל הרשימה, במקרה זה יש להחזיר את הרשימה בשלמותה).

- כתבו פונקציה בעלת המפרט (prototype) הבא:

void printList (Node * head)

הפונקציה מקבלת כפרמטר מצביע לראש הרשימה ומדפיסה את תוכן הרשימה כמחרוזת ובסיומה תו ירידת שורה (\n).

- כתבו תוכנית (main) שמשתמשת בפונקציות שכתבתם.

- התוכנית תקלוט רצף של תווים מהמשתמש ותבנה את הרשימה.
- לאחר מכן התוכנית תקלוט מהמשתמש גודל k ותדפיס את המחרוזת המתקבלת לאחר מחיקה בדילוגים בגודל k.
- התוכנית תבדוק שגודל הדילוג שלם וחיובי ממש. אחרת, תדפיס הודעת שגיאה מתאימה ותצא מהתוכנית.
- להזכירכם, יש לשחרר כל זיכרון שהוקצה באופן דינמי לאחר סיום השימוש בו או כאשר חלה שגיאה.

דוגמת הרצה (באדום פלט ובשחור קלט):

Enter a string:

abcd

Enter jump: 1

The result string is: ac

Press any key to continue . . .

Enter a string:

abcde

Enter jump: 3

The result string is: abce

Press any key to continue . . .

Enter a string:

abcd

Enter jump: 0

The input is not valid!

Press any key to continue . . .