

## **מבחן סדנה מתקדמת בתכנות-61108**

**ד"ר מרק קורנבליט, מר חיים שפיר, גב' אסתר אמיתי**

**סמסטר קיץ, מועד א', תשע"ח**

**תאריך: 12.10.18**

### **הוראות:**

- משך המבחן 3 שעות.
- אין להשתמש בחומרי עזר.

### **שאלה 1 (28 נק')**

כתוב פונקציה בוליאנית אשר מקבלת מערך מחרוזות וגודלו.  
על הפונקציה להחזיר 1 אם מחרוזות במערך ממיינות בסדר לקסיקוגרפי עולה (ז"א כל מחרוזת במקום i במערך היא מחרוזת לא קטנה לקסיקוגרפית ממחרוזת שבמקום i-1). אחרת הפונקציה תחזיר 0.

**דוגמה:** למערך המחרוזות הבא

32 \$  
A very nice man!  
Bring me, please.  
Many people  
My house  
Paul 8-th  
Paul 9-th  
a is the first letter

**הפונקציה תחזיר 1.**

```
int ordered_string_array (char **str, int n)
{
    int i;
    for (i=1; i<n; i++)
        if (strcmp(str[i], str[i-1]) < 0)
            return 0;
    return 1;
}
```



מכון טכנולוגי חולון  
Holon Institute of Technology

## שאלה 2 (28 נק')

נתונות ההגדרות הבאות של טיפוסים:

```
typedef struct data_item {  
    int data;  
    struct data_item *next;  
} Dataltem;
```

```
typedef struct ptr_item {  
    Dataltem *ptr;  
    struct ptr_item *next;  
} PtrItem;
```

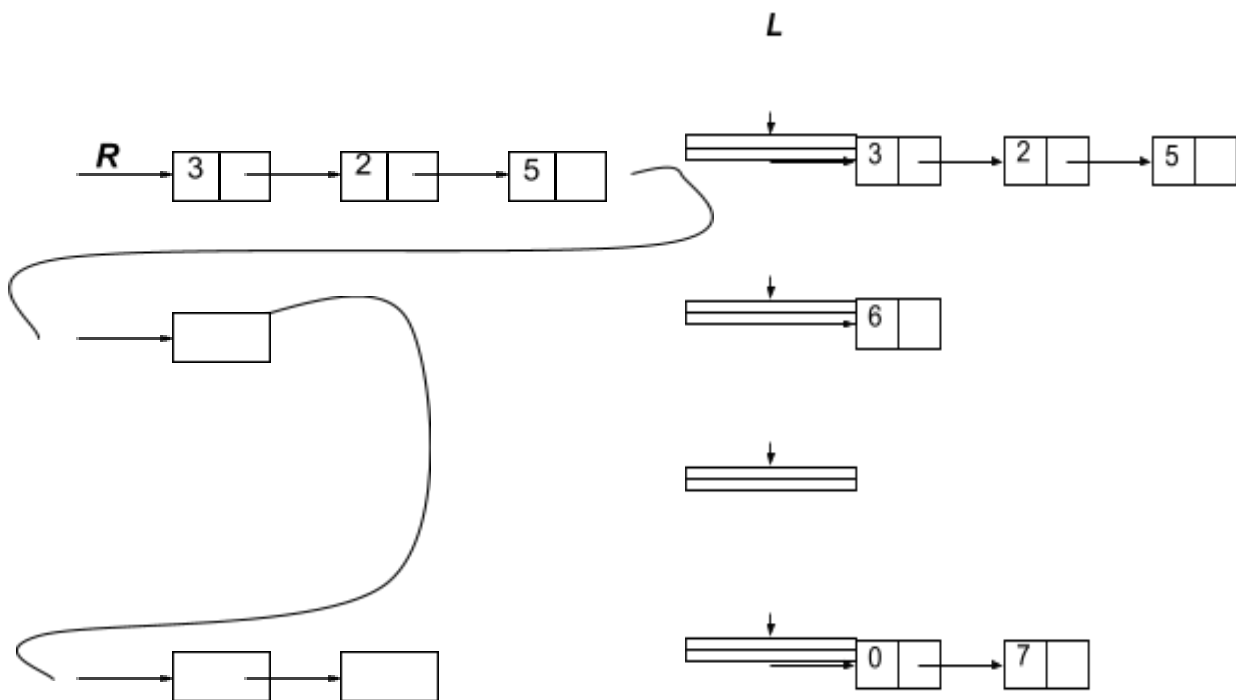
כתוב פונקציה המקבלת **רשימה מקושרת L** של **רשימות מקושרות** אשר מורכבת מאיברים מסוג **PtrItem**. השדה **ptr** של איבר מס' **i** ב- **L** יצביע לראש **רשימת המספרים** מס' **i** ושדה ה- **next** שלו יצביע לאיבר הבא ב- **L**. רשימות מספרים יהיו מורכבות מאיברים מסוג **Dataltem**.

על הפונקציה ליצור את הרשימה החדשה **R** אשר מורכבת מאיברים מסוג **Dataltem** ומכילה בתוכה את כל האיברים **המקוריים** של הרשימות של **L** (מהרשימה הראשונה ועד לרשימה האחרונה, ומהאיבר הראשון בכל רשימה עד לאיבר האחרון בתוכה).

**שים לב.** הפונקציה לא תקצה זיכרון לרשימה החדשה אלא תעדכן שדות **next** באיברים האחרונים של הרשימות של **L** כך שאיבר אחרון בכל רשימה יהיה קשור עם איבר ראשון של רשימה לא ריקה הבאה. על הפונקציה לשחרר את כל האיברים מסוג **PtrItem** של **L**.

הפונקציה תחזיר את כתובת האיבר הראשון של רשימת **R**.

דוגמא:



שים לב. רשימת  $L$  וכל אחת מהרשימות שלה יכולות להיות ריקות.

```
DataType *update_list_of_lists_to_list (PtrItem *L)
{
    DataType *data_item_list, *last = NULL, *R = NULL;
    PtrItem *temp = L;

    if (!L) return NULL;
    while (temp->next) // scan list of lists
    {
        if (temp->ptr) // if list is not empty
        {
            data_item_list = temp->ptr;
            if (last)
                //last element of previous non-empty list points
to first element of current list
                last->next = data_item_list;
            else
                //R points to first element of first non-empty
list
                R = data_item_list;
            while (data_item_list->next) // scan internal list
                data_item_list = data_item_list->next;
            last = data_item_list;
        }
        temp = temp->next;
    }
    if (last)
        //last element of previous non-empty list points to beginning
of last list
        last->next = temp->ptr;

    /* Release of list of lists */
    temp = L;
    while(temp)
    {
        L = L->next;
        free(temp);
        temp = L;
    }

    return R;
}
```

### שאלה 3 (28 נק')

כתוב פונקציה אשר מקבלת מערך דו-ממדי סטטי (לא דינאמי) של מספרים שלמים בעל **rows** שורות ו-**cols** עמודות, וגם מחרוזת **fileName**. המספר הפיסי של העמודות יינתן ע"י קבוע בשם **COLS**. יש להגדיר אותו לפני הפונקציה. ידוע שכל שורה ב-**A** היא מערך ממין בסדר עולה.

על הפונקציה ליצור קובץ טקסט ששמו נכלל ב- **fileName** לשמירת מערך **A** בדרך חסכונית וללא איברים חוזרים בשורות באופן הבא:

שורה מס' **i** בקובץ תתאים לשורה מס' **i** של **A** (השורות בקובץ ממוספרות מ-0). בתחילת כל שורה בקובץ יופיע מספר השורה עם נקודה ורווח אחריו. כל סדרת איברים זהים בשורה מס' **i** של **A** תיוצג בקובץ כזוג מספרים: (1) ערך האיבר החוזר ו-(2) מספר העמודה שבה מסתיימת הסדרה. מספרים בזוג יהיו מופרדים במקף. הזוגות בכל שורה של הקובץ יהיו ממוינים בסדר עולה על פי המספר הראשון בזוג (שהוא גם המספר שהופיע במקור במערך). הזוגות בכל שורה יהיו מופרדים ברווחים.

דוגמא:

תוכן הקובץ	A
0. 2-0 3-2 6-6 9-7	2 3 3 6 6 6 6 9
1. 0-3 1-5 7-7	0 0 0 0 1 1 7 7
2. 5-7	5 5 5 5 5 5 5 5
3. 3-2 4-7	3 3 3 4 4 4 4 4

```
void compress_matrix_to_file (int A[][COLS], int rows, int cols, char
*fileName)
{
    int i, j;
    FILE *fptr = fopen(fileName, "w");
    if (!fptr)
        exit(1);
    for (i=0; i<rows; i++)
    {
        fprintf(fptr, "%d.", i);
        for (j=1; j<cols; j++)
            if (A[i][j] != A[i][j-1]) //new series
                fprintf(fptr, " %d-%d", A[i][j-1], j-1);
        /* last series in row i */
        fprintf(fptr, " %d-%d\n", A[i][j-1], cols-1);
    }
    fclose(fptr);
}
```

**שאלה 4 (16 נק')**

נתונה התוכנית הבאה:

```
#include <stdio.h>

int fun (int);
int fun1 (int);
int fun2 (int);
int fun3 (int);

void main()
{
    int i;
    for(i=1; i<=3; i++)
        printf("%d \n", fun (i));
    printf("\n");
}

int fun (int x)
{
    static int i=0;
    int ( *pf[] )(int)={fun1, fun2, fun3},
        y = (*pf[i])(x);
    i++;
    return y;
}

int fun1 (int x)
{
    static int i=0;
    int a[]={10,20,30};
    i++;
    a[i] += x;
    return a[i];
}

int fun2 (int x)
{
    return x*x;
}
```

```
int fun3 (int x)
{
    return 3*x;
}
```

1. מהו פלט התוכנית? נמק.

21  
4  
9

פונקציה fun נקראת 3 פעמים ל-  $i=1,2,3$ .  
ל- $x=1$   $i=1$  בתוך `static int i=0` (fun) ולכן נקראת פונקציה fun1 אשר נמצאת במקום 0 במערך pf של מצביעים לפונקציות. ב-fun1 אשר מקבלת  $x=1$ ,  $a[1]=20$  מתעדכן ל- 21 ומוחזר לתוך y ב-fun אשר מוחזר ומודפס ב-main.  
ל- $i=2$  נקראת fun(2) ו- `static int i=1` בתוכה. לכן נקראת פונקציה fun2(2) שנמצאת במקום 1 ב-pf. הפונקציה מחזירה  $2*2=4$ .  
ל- $i=3$  נקראת fun(3) ו- `static int i=2` בתוכה. לכן נקראת פונקציה fun3(3) שנמצאת במקום 2 ב-pf. הפונקציה מחזירה  $3*3=9$ .

2. האם יהיה תקין למחוק את המילה `static` בפונקציה fun? נמק תשובתך. אם כן, הסבר כיצד ישתנה הפלט.

לא תקין עקב שגיאת ריצה: בקריאה השלישית ל-fun ניסיון לעדכן איבר מסי 3 שלא קיים במערך a של פונקציה fun1.

3. האם יהיה תקין למחוק את המילה `static` גם בפונקציה fun וגם בפונקציה fun1? נמק תשובתך. אם כן, הסבר כיצד ישתנה הפלט.

תקין.  
פלט:  
21  
22  
23

בכל 3 קריאת ל-fun קוראים ל-fun1 עם  $x=1,2,3$ . מחברים x ל- $a[1]=20$  ו- מחזירים 21, 22, 23 בהתאמה.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
void main()
{
    printf("%d",printf("%d",printf("%d",1)));
}
```

יש לנמק את התשובה.

**111**

פונקציה `printf` פנימית מדפיסה 1 ומחזירה מספר בתים כתובים שווה ל-1. לכן `printf` שנייה מדפיסה 1 וגם מחזירה 1 המודפס ע"י `printf` חיצונית.