

**COMPUTER SCIENCE DEPARTMENT המחלקה למדעי המחשב**

**סדנה מתקדמת בתכנות 61108**

סמסטר קיץ תשע"ה

מועד א'

14.10.2015

**מרצים: ד"ר מרק קורנבליט וד"ר לאוניד קוגל**

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר

**שאלה 1 (28 נק')**

כתוב פונקציה המקבלת מערך **דו-ממדי דינאמי A** של מספרים שלמים וכמות שורות בתוכו. איבר **0** בשורה **i** של **A** שווה לגודל (כולל איבר **0** עצמו) של שורה **i**.

על הפונקציה לבנות רשימה מקושרת המורכבת מכל האיברים של **A** פרט לאיברי **0** של שורותיו (משורה **0** עד לשורה אחרונה, מאיבר **1** בשורה עד לאיבר אחרון בתוכה).

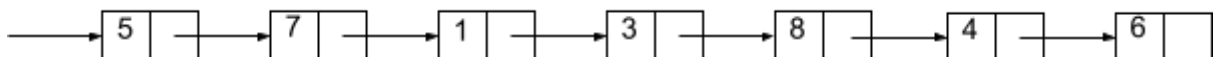
הפונקציה תחזיר את כתובת האיבר הראשון ברשימה.

**דוגמא:**

**A**

EMBED Equation.3

**רשימה:**



יש להגדיר את הטיפוס של איברי הרשימה מעל הפונקציה.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

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

Item *two_dimensional_to_list (int **A, int rows_num)
{
    int i, j;
    Item *curr_ptr, *lst = (Item *)malloc(sizeof(Item)); //dummy
    curr_ptr = lst;
    for (i=0; i<rows_num; i++)
        for (j=1; j<A[i][0]; j++)
        {
            curr_ptr->next = (Item *)malloc(sizeof(Item));
            curr_ptr = curr_ptr->next;
            curr_ptr->data = A[i][j];
        }
    curr_ptr->next = NULL;
    /* deletion of dummy */
    curr_ptr = lst;
    lst = lst->next;
    free (curr_ptr);
    return lst;
}
```

### שאלה 2 (28 נק')

כתוב פונקציה המקבלת מערך **A** של מספרים שלמים ומספר שלם **k**.  
על הפונקציה לבנות שני מערכים חדשים **דינאמיים** כך שהמערך הראשון יהיה מורכב מאיברי מערך **A** הגדולים מ-**k** והמערך השני יהיה מורכב מאיברי מערך **A** הקטנים מ-**k**.  
הפונקציה תעביר by reference את הכתובות של המערכים החדשים ואת הגדלים שלהם ותחזיר את מספר האיברים של מערך **A** השווים ל-**k**.

לדוגמא, עבור המערך **A** הבא:

67	9	88	78	57	88	74	57	6	57
----	---	----	----	----	----	----	----	---	----

ומספר **k** השווה ל-57

הפונקציה תיצור את שני המערכים החדשים הבאים:

67	88	78	88	74
----	----	----	----	----

9	6
---	---

הפונקציה תעביר גם מספרים 5 ו-2 ותחזיר 3.



## מכון טכנולוגי חולון Holon Institute of Technology

במידה ואחד מהמערכים החדשים לא מכיל אף איבר, כתובתו תהיה NULL וגודלו יהיה 0.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
int big_small_num (int *a, int size, int k, int **array_big,
                  int **array_small, int *size_big, int *size_small)
{
    int count_big = 0, count_small = 0, equal_num, i;
    /* Estimation of array sizes */
    *size_big = *size_small = 0;
    for (i=0; i<size; i++)
        if (a[i] > k)
            (*size_big)++;
        else
            if (a[i] < k)
                (*size_small)++;
    equal_num = size - *size_big - *size_small;
    /* Allocation of arrays */
    if (*size_big)
        *array_big = (int *)malloc(*size_big*sizeof(int));
    else
        *array_big = NULL;
    if (*size_small)
        *array_small = (int *)malloc(*size_small*sizeof(int));
    else
        *array_small = NULL;
    if (equal_num == size)
        return equal_num;
    /* Filling arrays */
    for (i=0; i<size; i++)
        if (a[i] > k)
        {
            (*array_big)[count_big] = a[i];
            count_big++;
        }
        else
            if (a[i] < k)
            {
                (*array_small)[count_small] = a[i];
                count_small++;
            }
    return equal_num;
}
```

### שאלה 3 (28 נק')

כתוב פונקציה בוליאנית אשר מקבלת מחרוזת המהווה שם של קובץ טקסט. על הפונקציה להחזיר 1 אם שורות הקובץ ממוינות בסדר לקסיקוגרפי עולה (ז"א כל שורה במקום  $i$  מלמעלה היא מחרוזת לא קטנה לקסיקוגרפית ממחרוזת (שורה) במקום  $(i-1)$ ). אחרת הפונקציה תחזיר 0.

דוגמה:

Alex 80  
Bill 89  
Jim 88  
Jonathan 74  
Mike 93  
Paul 60  
Peter 100

הפונקציה תחזיר 1 כי שורות הקובץ ממוינות בסדר עולה.

במקרה של קובץ ריק על הפונקציה להחזיר 1.  
אפשר (אבל לא חובה) להשתמש בפונקציה סטנדרטית fgets בעלת הפורמט הבא:

(שם מחרוזת, מספר שלם, מצביע לקובץ) fgets

ניתן להניח שאורך השורה במחרוזת אינו עולה על 99.

```
int ordered_file (char *file)
{
    FILE *fptr = fopen(file, "r");
    char str_prev[100]="", str_next[100];
    if( !fptr )
        return 0;
    while (fgets(str_next, 100, fptr))
    {
        if (strcmp(str_prev, str_next) > 0)
        {
            fclose(fptr);
            return 0;
        }
        strcpy(str_prev, str_next);
    }
    fclose(fptr);
    return 1;
}
```

שאלה 4 (16 נק')

נתונה התוכנית הבאה:

```
#include <stdio.h>

struct first
{
    int x;
    double y;
};
```

```
typedef union
{
    int x;
    int *p;
} second;

void main()
{
    // need word struct before first
    first st1, st2 = {5,8.9}, *ptr;
    second un;
    ptr = &st1;
    scanf ("%d", &st1.x);
    scanf ("%lf", ptr->y); // running error: need &
    if (st1 == st2) // '==' : illegal for struct
        printf ("Structures are equal\n");
    un.p = &st1.x;
    (*un.p)++;
    printf ("%d\n", *un.p);
    un.x = 3;
    // running error:
    (*un.p)--; // un.p is destroyed after changing un.x
    printf ("%d\n", *un.p);
}
```

התוכנית שגויה!  
יש לתאר את כל השגיאות (קומפילציה וריצה).

**שאלה 5 (בונוס – 10 נק')**

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
    char c[4][4][4][4]={0};
    printf("%c", "c[0][1][2]"[3]);
}
```



**מכון טכנולוגי חולון**  
Holon Institute of Technology

[