

**המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT**

**סדנה מתקדמת בתכנות 61108**

סמסטר ב' תשע"ה

מועד א'

30.06.2015

**מרצה: ד"ר מרק קורנבליט**

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדפים המצורפים לשאלון

**שאלה 1 (28 נק')**

כתוב פונקציה המקבלת מערך **A** של מספרים שלמים לא שליליים. המערך מורכב מסדרות של מספרים חיוביים המובדלות באפסים (הסדרה יכולה להיות ריקה, ז"א אפסים ב- **A** יכולים להופיע גם ברציפות). גם האיבר הראשון של **A** יכול להיות אפס (אם הסדרה הראשונה היא ריקה). האיבר האחרון של **A** תמיד יהיה אפס (סוף הסדרה האחרונה).

על הפונקציה לבנות מערך **דו-ממדי דינאמי** בעל שורות עם גדלים שונים כדלקמן:

- כמות שורות במערך הדו-ממדי תהיה שווה לכמות סדרות ב- **A**;
- גודל שורה מס' **i** במערך הדו-ממדי יהיה אחד גדול מגודל סדרה מס' **i** (משמאל לימין) ב- **A** (מניחים שסדרות ב- **A** ממוספרות מ-0);
- איבר מס' **0** בשורה מס' **i** במערך הדו-ממדי יהיה שווה לגודל סדרה מס' **i** ב- **A**;
- שאר האיברים בשורה מס' **i** במערך הדו-ממדי יהיו שווים לאיברים של סדרה מס' **i** ב- **A** (סדר האיברים נשמר).

הפונקציה תחזיר כתובת של המערך הדו-ממדי הדינאמי הבנוי ותעביר by reference את מספר השורות בתוכו.

**דוגמא:**

מערך חדש

**A** = 5, 7, 9, 0, 12, 11, 0, 1, 3, 10, 87, 0, 6, 5, 0, 0, 31, 0

EMBED Equation.3

הפונקציה תעביר 6 בדוגמא.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

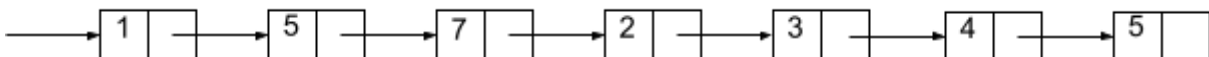
```
int **one_to_two_dimensional (int *A, int A_size, int *rows_num)
{
    // temp. array of row sizes (initially - zeros)
    int *row_size = (int *)calloc (A_size, sizeof(int)),
    **arr, offset=0, i=0, j;
    for (j=0; j<A_size; j++)
        if (A[j] != 0)
            row_size[i]++;
        else
            i++; // new sequence
    *rows_num = i;
    arr = (int **)calloc (*rows_num, sizeof(int*));
    for (i=0; i < *rows_num; i++)
    {
        // allocation of row i
        arr[i] = (int *)calloc (row_size[i]+1, sizeof(int));
        arr[i][0] = row_size[i];
        /* copying sequence i into row i */
        for (j=0; j < row_size[i]; j++)
            arr[i][j+1] = A[offset+j];
        offset += row_size[i] + 1;
    }
    free (row_size);
    return arr;
}
```

### שאלה 2 (28 נק')

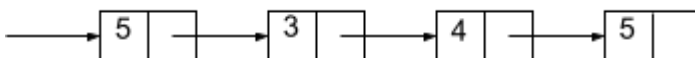
קטע סגור  $[a, b]$  הוא קבוצת כל הנקודות על ישר המספרים אשר נמצאות בין שתי הנקודות  $a$  ו- $b$  (הנקראות קצוות הקטע) כולל  $a$  ו- $b$  עצמם.

כתוב פונקציה המקבלת רשימה מקושרת של מספרים שלמים וגם שני מספרים שלמים  $a$  ו- $b$  שהם קצוות של הקטע הסגור. על הפונקציה לבנות את הרשימה החדשה המורכבת מאיברי הרשימה המקורית הנמצאים בתוך קטע  $[a, b]$ . הפונקציה תחזיר את כתובת האיבר הראשון ברשימה החדשה.

לדוגמא, עבור מספרים  $a$  השווה ל-3 ו- $b$  השווה ל-6 והרשימה המקורית הבאה:



הפונקציה תיצור את הרשימה החדשה הבאה:



יש להגדיר את הטיפוס של איברי הרשימות מעל הפונקציה.

ניתן להניח שיש בדיכרון מספיק מקום להקצאה.

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

Item *closed_interval (Item *lst, int a, int b)
{
    Item *curr_ptr, *new_lst = (Item *)malloc(sizeof(Item)); //dummy
    curr_ptr = new_lst;
    while (lst)
    {
        if (lst->data >= a && lst->data <= b)
        {
            curr_ptr->next = (Item *)malloc(sizeof(Item));
            curr_ptr = curr_ptr->next;
            curr_ptr->data = lst->data;
        }
        lst = lst->next;
    }
    curr_ptr->next = NULL;
    /* deletion of dummy */
    curr_ptr = new_lst;
    new_lst = new_lst->next;
    free (curr_ptr);
    return new_lst;
}
```

### שאלה 3 (28 נק')

- כתוב פונקציה (void) שמקבלת מחרוזת ומעבדת אותה כדלקמן:
- כל אות קטנה תהפוך לאות הגדולה התואמת;
  - כל אות גדולה תהפוך לאות הקטנה התואמת;
  - כל ספרה זוגית תהפוך לספרה העוקבת וכל ספרה אי-זוגית תהפוך לספרה הקודמת;
  - כל תו אחר בעל קוד ASCII זוגי תהפוך לתו בעל קוד ה-ASCII הקודם;
  - כל תו אחר בעל קוד ASCII אי-זוגי תהפוך לתו בעל קוד ה-ASCII העוקב.

ניתן להניח שתווי המחרוזת לא נמצאים בקצוות טבלת ה-ASCII.

#### דוגמא:

B9\$q0?3Me  
b8#Q1@2mE

מחרוזת לפני עיבוד:  
המחרוזת אחרי עיבוד:

קודי ה-ASCII של תווי המחרוזת הנ"ל הם: \$ - 36, # - 35, ? - 63, @ - 64.

```
void string_transform (char *str)
```

```
{
    int i;
    for (i=0; str[i]!='\0'; i++)
        if (islower(str[i]))
            str[i] = toupper(str[i]);
        else
            if (isupper(str[i]))
                str[i] = tolower(str[i]);
            else
                if (isdigit(str[i]))
                    if ((str[i]-'0')%2 == 0)
                        str[i]++;
                    else
                        str[i]--;
                else
                    if (str[i]%2 == 0)
                        str[i]--;
                    else
                        str[i]++;
}
```

## שאלה 4 (16 נק')

נתונה התוכנית הבאה:

```
#include <stdio.h>

void main()
{
    int i, j;
    //it isn't checked whether open fails
    FILE *f=fopen("file.dat", "w");
    for (i=1; i<=10; i++)
    {
        fprintf("file.dat", "%d", 2*i);
        //pointer to file shouldn't change for scanning file
        f++;
    }
    for (i=10; i>=1; i--)
    {
        //file was opened for writing only
        fscanf("file.dat", "%d", &j);
        printf ("%d ", j);
        //pointer to file shouldn't change for scanning file
        f--;
    }
    //file isn't closed
}
```

Here is file's end. No value is entered and j is not initialized

Here shouldn't be pointer to file and not file's name

התוכנית שגויה!  
יש לתאר את כל השגיאות.

**שאלה 5 (בונוס – 10 נק')**

מהו הפלט של התוכנית הבאה?

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

void main()
{
    char str[]="strcmp";
    str[!strcmp("strcmp","strcmp")] = strcmp("strcmp","strcmp");
    puts(str);
}
```

**S**