



## המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

### סדנה מתקדמת בתכנות 61108

סמסטר ב' תשע"ב

מועד ב'

1.08.2012

### מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

### שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת מערך דו-ממדי סטטי (ז"א לא דינאמי)  $A$  של מספרים שלמים בעל  $n$  שורות ו-  $m$  עמודות. ידוע שכל שורה ב-  $A$  היא מערך ממוין.

על הפונקציה לבנות מערך דו-ממדי דינאמי  $B$  בעל  $n$  שורות עם גדלים שונים כך ששורה מס'  $i$  שלו תהיה בעלת אותו תוכן של תאי שורה מס'  $i$  במערך  $A$ , אך ללא איברים חוזרים.

הפונקציה תיצור גם מערך  $SIZE$  המורכב מגדלי שורות של מערך  $B$  כך שאיבר מס'  $i$  של  $SIZE$  יהיה שווה לגודל שורה מס'  $i$  של  $B$ .

דוגמא:

<u>SIZE</u>	<u>B</u>	<u>A</u>
4	2 3 6 9	2 3 3 6 6 6 6 9
3	0 1 7	0 0 0 0 1 1 7 7
1	5	5 5 5 5 5 5 5 5
8	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7

על הפונקציה להחזיר את הכתובת של מערך  $B$  ולהעביר את הכתובת של מערך  $SIZE$ .

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

#define COLS 8 //example

int **compress_matrix (int A[][COLS], int rows, int cols, int **SIZE)
{
    int i, j, k, **B = (int **)calloc (rows, sizeof(int *));
    *SIZE = (int *)calloc (rows, sizeof(int));
    for (i=0; i<rows; i++)
    {
        /* Estimation of i-th row size */
        (*SIZE)[i] = 1;
        for (j=1; j<cols; j++)
            if (A[i][j] != A[i][j-1])
                (*SIZE)[i]++;
        /* Allocation of i-th row */
        B[i] = (int *)calloc((*SIZE)[i], sizeof(int));
        /* Filling i-th row */
        B[i][0] = A[i][0];
        k = 1;
        for (j=1; j<cols; j++)
            if (A[i][j] != A[i][j-1])
            {
                B[i][k] = A[i][j];
                k++;
            }
    }
    return B;
}

```

## שאלה 2 (28 נק')

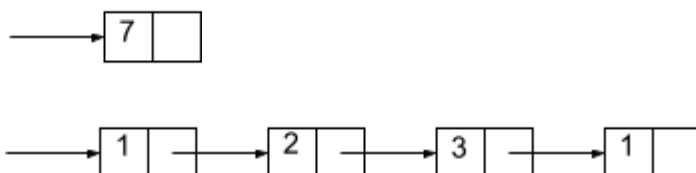
כתוב פונקציה המקבלת מערך  $A$  של מספרים שלמים ומספר שלם  $k$ .  
 על הפונקציה לבנות שתי רשימות מקושרות כך שהרשימה הראשונה תהיה מורכבת מאיברי מערך  $A$  הגדולים מ- $k$  והרשימה השנייה תהיה מורכבת מאיברי מערך  $A$  הקטנים מ- $k$ .  
 הפונקציה תעביר את המצביעים לאיברים הראשונים של שתי הרשימות ותחזיר את מספר האיברים של מערך  $A$  השווים ל- $k$ .

לדוגמא, עבור המערך  $A$  הבא:

1	5	7	2	3	1	5
---	---	---	---	---	---	---

ומספר  $k$  השווה ל-5

הפונקציה תיצור את שתי הרשימות הבאות:



ותחזיר 2.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

typedef struct element
{
    int data;
    struct element *ptr_next;
} element;

int array_to_lists_of_large_small_num (int *a, int size, int k,
                                         element **lst_large, element **lst_small)
{
    element *ptr_large, *ptr_small;
    int equal_num=0, i;
    /* allocation of dummy elements */
    *lst_large = (element *)malloc(sizeof(element));
    *lst_small = (element *)malloc(sizeof(element));
    /* ----- */
    ptr_large = *lst_large;
    ptr_small = *lst_small;
    for (i=0; i<size; i++)
    {
        if (a[i] > k)
        {
            ptr_large->ptr_next =
                (element *)malloc(sizeof(element));
            ptr_large = ptr_large->ptr_next;
            ptr_large->data = a[i];
        }
        else
            if (a[i] < k)
            {
                ptr_small->ptr_next =
                    (element *)malloc(sizeof(element));
                ptr_small = ptr_small->ptr_next;
                ptr_small->data = a[i];
            }
            else
                equal_num++;
    }
    ptr_large->ptr_next = ptr_small->ptr_next = NULL;
    /* deletion of dummy elements */
    ptr_large = *lst_large;
    *lst_large = (*lst_large)->ptr_next;
    free (ptr_large);
    ptr_small = *lst_small;
    *lst_small = (*lst_small)->ptr_next;
    free (ptr_small);
    /* ----- */
    return equal_num;
}

```

### שאלה 3 (28 נק')

כתוב פונקציה (void) אשר מקבלת מחרוזת ומעבדת אותה כדלקמן:

- אם תו שנמצא במקום  $i$  הוא ספרה ותווים הנמצאים במקומות  $i-1$  ו- $i+1$  הם לא ספרות אז זוג תווי מס'  $i$  ו- $i+1$  יהפוך לסדרת התווים השווים לתו מס'  $i+1$ . אורך הסדרה יהיה שווה למספר המיוצג ע"י הספרה ממקום  $i$ ;
- אם המחרוזת מכילה רצף ספרות הנמצאות במקומות  $i+3, i+2, i+1, i, \dots$  אז הרצף יהפוך לסדרות התווים השווים לתווי מס'  $i+3, i+1, \dots$ . אורכי הסדרות יהיו שווים למספרים המיוצגים ע"י הספרות ממקומות  $i+2, i, \dots$  בהתאמה;
- התו שהוא לא ספרה יישאר במחרוזת ללא שינוי אם התו שלפניו הוא גם לא ספרה;

- הפונקציה תסיר מהמחרוזת את הספרה שנמצאת במקום האחרון במחרוזת אם לפי הכללים לעיל היא תציג אורך סדרה.

**דוגמאות:**

<b>st4f&amp;2*1Q3549</b>	מחרוזת לפני עיבוד:
<b>stffff&amp;**Q5559999</b>	מחרוזת אחרי עיבוד:

<b>2%Kg0P8</b>	מחרוזת לפני עיבוד:
<b>Kg%%</b>	מחרוזת אחרי עיבוד:

<b>H^725</b>	מחרוזת לפני עיבוד:
<b>H^^^^^2222222</b>	מחרוזת אחרי עיבוד:

ניתן להניח שהגודל הפיסי של המחרוזת הנתונה הוא מספיק למחרוזת המעובדת.

אפשר (אבל לא חובה) להשתמש במחרוזת עזר. במקרה זה ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
void string_transform (char *str)
{
    int i=0, j=0, len=strlen(str);
    char c, *temp = (char *)malloc(len+1);
    strcpy (temp, str);
    while (i<len-1)
    {
        if (isdigit(temp[i]))
        {
            for (c='1'; c<=temp[i]; c++, j++)
                str[j] = temp[i+1];
            i+=2;
        }
        else
        {
            str[j] = temp[i];
            j++;
            i++;
        }
    }
    if (i==len-1 && !isdigit(temp[len-1]))
    {
        str[j] = temp[len-1];
        j++;
    }
    str[j] = '\0';
    free (temp);
}
```

**שאלה 4 (16 נק')**  
נתונה התוכנית הבאה:

```
#include <stdio.h>

int fun (int);

void main ()
{
    int i;
    for (i=1; i<=4; i++)
```

```

        printf ("%d ", fun(i));
    }

    int fun (int x)
    {
        static int n=2;  // 1
        n++;
        return (x+n & 1) % 2; // 2
    }

```

1. מהו פלט התוכנית?

**0 0 0 0**

2. מה יהיה פלט התוכנית ללא המילה static בשורה המסומנת בהערה כ-1 (// 1)?

**0 1 0 1**

3. מה יהיה פלט התוכנית ללא המילה static בשורה המסומנת בהערה כ-1 (// 1) ועם הסימן & במקום הסימן & בשורה המסומנת בהערה כ-2 (// 2)?

**1 1 1 1**

4. מה יהיה פלט התוכנית עם המילה static בשורה המסומנת בהערה כ-1 (// 1) ועם הסימן & במקום הסימן & בשורה המסומנת בהערה כ-2 (// 2)?

**1 1 1 1**

**שאלה 5 (בנוס – 10 נק')**  
מהו פלט התוכנית הבאה?

```

#include <stdio.h>

void main()
{
    printf ("%s\n", !"!"?"?":":");
}

```

▪  
▪