

## מבחן סדנה מתקדמת בתכנות-61108

ד"ר מרק קורנבליט וד"ר נעמה קופלמן

סמסטר ב', מועד א', תשע"ח

תאריך: 21.06.18

### הוראות:

- משך המבחן 3 שעות.
- אין להשתמש בחומרי עזר, פרט לדף המצורף לשאלון

### שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת מערך דו-ממדי דינאמי **A** בעל  $n$  שורות וגם מערך **SIZE** המכיל את אורכי השורות של מערך **A** – כך שהאיבר ה- $i$  במערך **SIZE** שווה לגודל השורה ה- $i$  ב-**A**.

על הפונקציה לבצע את היסט מחזורי של השורות ב-**A**, כך ששורה מס' 0 תעבור למקום 1, שורה מס' 1 תעבור למקום 2, ..., שורה מס'  $2n-2$  תעבור למקום  $1n-1$  ושורה מס'  $1n-1$  תעבור למקום 0. בנוסף, הפונקציה תעדכן את איברי **SIZE** בהתאם להיסט המחזורי שבוצע על **A**.

דוגמא (בתחילת כל שורה נמצא מספרה):

#### A אחרי ההיסט

0	2	20	4
1	5	12	6 9
2	0	3	1 29 101
3	13	14	

#### A לפני ההיסט

0	5	12	6 9
1	0	3	1 29 101
2	13	14	
3	2	20	4

**SIZE** לפני ההיסט (משמאל לימין): 4, 5, 2, 3.  
**SIZE** אחרי ההיסט (משמאל לימין): 3, 4, 5, 2.

זמן ריצת הפונקציה יהיה תלוי רק בכמות שורות במערך **A** ולא יהיה תלוי בגדלי השורות!

על הפונקציה להיות אוניברסאלית כלפי טיפוס איברי **A**, ז"א לבצע את ההיסט המחזורי על המערך בעל איברים של טיפוס שרירותי.

אפשר להשתמש במשתני עזר בודדים, אך לא במערכים מכל סוג שהוא.

```

void row_shift (void **arr, int rows, int *SIZE)
{
    void *temp_row = arr[rows-1];
    int temp_size = SIZE[rows-1], i;
    for (i = rows-1; i>0; i--)
    {
        arr[i] = arr[i-1];
        SIZE[i] = SIZE[i-1];
    }
    arr[0] = temp_row;
    SIZE[0] = temp_size;
}
  
```

### שאלה 2 (28 נק')

נתונות הגדרה הבאה של הטיפוס:

```

typedef struct num_pair
{
    int count;
    int sum;
} NumPair;
  
```

כתוב פונקציה אשר מקבלת מערך דו-ממדי סטטי (לא דינאמי)  $M$  של מספרים שלמים בעל  $rows$  שורות ו- $cols$  עמודות. המספר הפיסי של העמודות יינתן ע"י קבוע בשם  $COLS$ . יש להגדיר אותו לפני הפונקציה.

על הפונקציה לבנות מערך דינאמי  $A$  של איברים מסוג  $NumPair$ . שדה ה- $count$  באיבר ה- $i$  של  $A$  יהיה שווה לכמות האיברים הזוגיים בשורה ה- $i$  של  $M$ . שדה ה- $sum$  באיבר ה- $i$  של  $A$  יהיה שווה לסכום האיברים בשורה ה- $i$  של  $M$ .

הפונקציה תעביר by reference את הכתובת של מערך  $A$  ותחזיר את ערך האיבר המקסימלי ב- $M$ .

**דוגמא:**

$A$	
count	sum
4	25
2	43
0	47
3	74

$$M = \begin{bmatrix} 6 & 0 & 4 & 8 & 7 \\ 3 & 10 & 17 & 7 & 6 \\ 9 & 1 & 5 & 31 & 1 \\ 0 & 7 & 48 & 15 & 4 \end{bmatrix}$$

הפונקציה תחזיר 48 בדוגמא.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
#define COLS 5 //example

int matrix_statistics (int M[][COLS], int rows, int cols, NumPair **A)
{
    int max = M[0][0], i, j;
    *A = (NumPair *)calloc(rows, sizeof(NumPair)); // Array's
    allocation: calloc sets all data to zeros
    for (i=0; i<rows; i++)
        for (j=0; j<cols; j++)
        {
            (*A)[i].sum += M[i][j];
            if (M[i][j]%2 == 0)
                (*A)[i].count++;
            if (M[i][j] > max)
                max = M[i][j];
        }
    return max;
}
```

### שאלה 3 (28 נק')

נתונות הגדרה הבאה של הטיפוס:

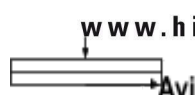
```
typedef struct item
{
    char *string;
    struct item *next;
} Item;
```

כתוב פונקציה אשר מקבלת מחרוזת המהווה שם של קובץ טקסט, ובונה רשימה מקושרת L של מחרוזות המורכבת מאיברים מסוג Item.

שדה ה-string של האיבר (חוליה) ה-i ברשימה L (מתחילתה) יצביע למחרוזת דינאמית המכילה את המילה הראשונה בשורה ה-i מתחילת קובץ הטקסט (מילה היא רצף תווים שאינם רווח המופרדים ממילים אחרות ע"י רווחים). נתון כי אף שורות הקובץ אינן מתחילות ברווח. תתכן שורה המורכבת ממילה יחידה בלבד.

הפונקציה תחזיר את כתובת ראש הרשימה L. במקרה של קובץ ריק או אם הקובץ לא נפתח על הפונקציה להחזיר NULL.

דוגמא:



### תוכן הקובץ

Moshe lives in Tel Aviv  
Irit was the first  
Avi has two children

גולומב 52, ת.ד. 305, חולון 10201  
טלפקס: 03-5026528

52 Golomb St., Holon 5810201 Israel  
Tel/Fax: 972-3-502-6528

[www.hit.ac.il](http://www.hit.ac.il)

המחלקה למדעי המחשב  
Faculty of Sciences  
Department of Computer Science

אפשר (אך לא חובה) להשתמש בפונקציה הסטנדרטית fgets בעלת הפורמט הבא:

(שם מחרוזת, מספר שלם, מצביע לקובץ) fgets

ניתן להניח שאורך כל שורה בקובץ אינו עולה על 99 תווים, וכי יש מספיק מקום בזיכרון לכל ההקצאות.

```
Item *file_to_list (char *file_name)
{
    char str[100];
    FILE *fptr = fopen(file_name, "r");
    Item *lst, *curr_ptr;
    int i;
    if (!fptr)
        return NULL;
    lst = (Item *)malloc(sizeof(Item)); //dummy
    curr_ptr = lst;
    while (fgets(str, 100, fptr))
    {
        curr_ptr->next = (Item *)malloc(sizeof(Item));
        curr_ptr = curr_ptr->next;

        curr_ptr->string = (char *)malloc(100);
        for (i=0; str[i]!='\0' && str[i]!=' '; i++)
            curr_ptr->string[i] = str[i];
        curr_ptr->string[i] = '\0';
        curr_ptr->string = (char *)realloc(curr_ptr->string, i+1);
    }
    curr_ptr->next = NULL;
    /* deletion of dummy */
    curr_ptr = lst;
    lst = lst->next;
    free (curr_ptr);
    fclose(fptr);
    return lst;
}
```

שאלה 4 (16 נק')

נתונה התוכנית הבאה:

```
#include <stdio.h>

int fun(const int *, int *const);

void main()
{
    int a[]={1,2,3}, b[]={1,2,3}, *const p = a;
    printf("%d",fun(p, b));
    printf("%d",fun(a, ++b));
    שגיאת קומפילציה: אסור לשנות b כי שם המערך הוא קבוע
    printf("%d",fun(p+1, b));
}

int fun(const int *a, int *const b)
{
    a++;
    a[0] = b[1];
    שגיאת קומפילציה: אסור לשנות כי a מצביע ל-const
    b[0] = a[1];
    return *(a+b[1]);
    שגיאת ריצה: אחרי פעולת a++, a פורמלי מצביע ל-a[1] אמיתי.
    b[1] שווה ל-2.
    לכן [a+b[1] זה כתובת של a[3] שנמצא מעבר לגבול המערך.
}

```

האם קוד התכנית הוא תקין? **לא**

אם כן, מהו פלט התכנית?  
אם לא, יש לתאר את כל השגיאות (קומפילציה וריצה).

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
printf("%d\n", (int)pow((int)"pow", 'p'^'o'^'w'^(' ^ 'p'^'o'^'w'^('));
}
```

יש לנמק את התשובה.

1

(int)"pow" זה ייצוג מספרי של כתובת של מחרוזת "pow".  
פעולת  $\wedge$  (XOR) היא חלופית ונותנת 0 לאופרנדים שווים.  
כל אחד מהתווים 'w', 's', 'p', 'o' מופיע פעמיים.  
לכן הביטוי  $'p'^{'o'^{'w'^{'( ^ {'p'^{'o'^{'w'^{'($  שווה ל-0.  
pow היא פונקציית חזקה ו- (int)pow זו המרת החזקה למספר שלם.  
כך מקבלים מספר שלם בחזקת 0.  
לכן הפלט יהיה 1.