

**COMPUTER SCIENCE DEPARTMENT המחלקה למדעי המחשב**

**סדנה מתקדמת בתכנות 61108**

סמסטר קיץ תשע"ז

מועד א'

19.10.2017

**מרצים: ד"ר מרק קורנבליט, מר רועי זימון, גב' אסתר אמיתי**

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

**שאלה 1 (28 נק')**

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

על הפונקציה לבנות מערך דו-ממדי דינאמי  $B$  חדש בעל  $n$  שורות עם גדלים שונים. כל שורה מס'  $i$  של  $B$  תורכב אך ורק מהמספרים הראשוניים (המתחלקים בעצמם וב-1 בלבד) הנמצאים בשורה מס'  $i$  של  $A$ . כל שורה של  $B$  תסתיים באיבר ששווה ל-0.

הפונקציה תחזיר כתובת של המערך החדש.

**דוגמא:**

$\underline{B}$					$\underline{A}$					
2	3	7	11	0	2	3	8	7	4	11
5	17	19	0		5	8	6	17	15	19
0					6	9	4	55	25	10
3	3	29	0		3	3	33	40	44	29

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.



מכון טכנולוגי חולון  
Holon Institute of Technology

```
int is_prime (int num)
{
    int i, lim = (int)sqrt(num);
    if (num == 2)
        return 1;
    if (num < 2 || num%2 == 0)
        return 0;
    for (i=3; i<=lim; i+=2)
        if (num % i == 0)
            return 0;
    return 1;
}

#define COLS 6 //example

int **prime_numbers_array (int A[][COLS], int rows, int cols)
{
    int i, j, count;
    int **B = (int **)malloc(rows*sizeof(int *));
    for (i=0; i<rows; i++)
    {
        B[i] = (int *)malloc((cols+1)*sizeof(int));
        count = 0;
        for(j=0; j<cols; j++)
        {
            if (is_prime(A[i][j]))
            {
                B[i][count] = A[i][j];
                count++;
            }
        }
        B[i][count] = 0;
        B[i] = (int *)realloc(B[i], (count+1)*sizeof(int));
    }
    return B;
}
```

גולומב 52, ת.ד. 305, חולון 5810201  
טלפקס: 03-5026528

52 Golomb St., Holon 5810201 Israel

**www.hit.ac.il** Tel/Fax: 972-3-502-6528

הפקולטה למדעים  
המחלקה למדעי המחשב

Faculty of Sciences

Department of Computer Science

**שאלה 2 (28 נק')**

נתונות הגדרות הבאות של הטיפוסים:

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

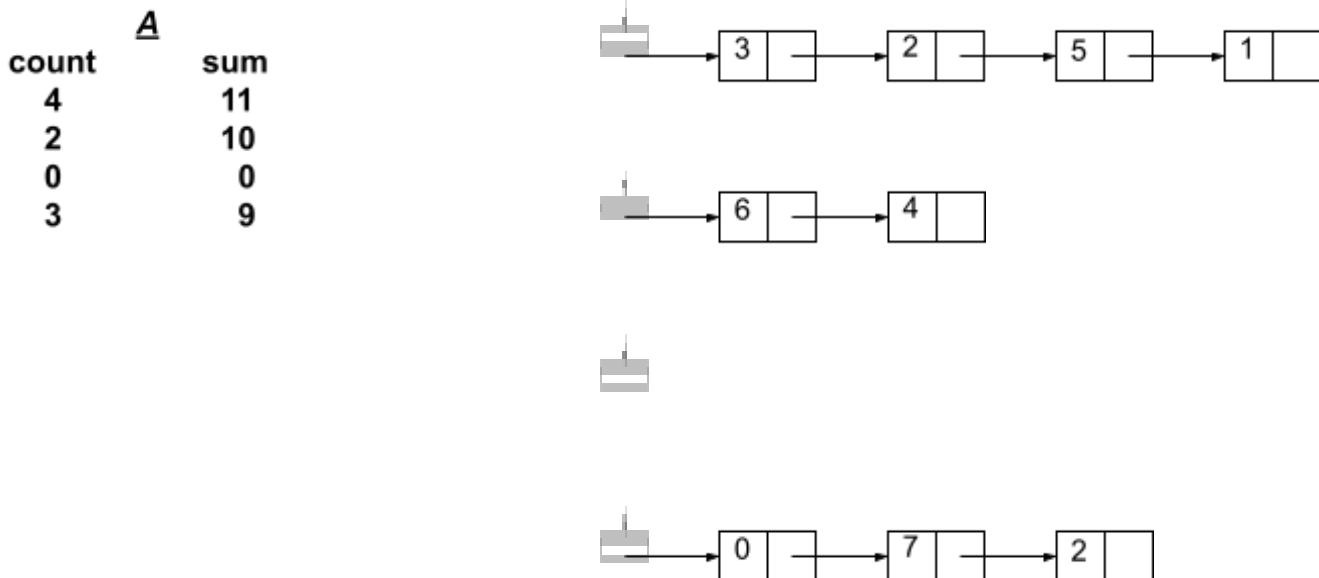
```
typedef struct ptr_item {
    Dataltem *ptr;
    struct ptr_item *next;
} PtrItem;
```

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

כתוב פונקציה המקבלת רשימה מקושרת  $L$  של רשימות מקושרות אשר מורכבת מאיברים מסוג **PtrItem**. שדה **ptr** של איבר מס'  $i$  ב- $L$  יצביע לראש רשימת מספרים מס'  $i$  ושדה **next** שלו יצביע לאיבר הבא ב- $L$ . רשימות מספרים יהיו מורכבות מאיברים מסוג **Dataltem**.  
על הפונקציה לבנות מערך דינאמי **A** של איברים מסוג **NumPair**. שדה **count** באיבר מס'  $i$  של **A** יהיה שווה לכמות האיברים ברשימה מס'  $i$  של  $L$ . שדה **sum** באיבר מס'  $i$  של **A** יהיה שווה לסכום שדות **data** של איברי הרשימה מס'  $i$  של  $L$ .  
הפונקציה תעביר by reference את הכתובת של מערך **A** ותחזיר את גודלו.

$L$

דוגמא:



הפונקציה תחזיר 4 בדוגמא.



## מכון טכנולוגי חולון Holon Institute of Technology

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
int list_of_lists_statistics (PtrItem *list, NumPair **A)
{
    PtrItem *temp = list;
    int len=0, i=0;
    DataItem *data_item_list;

    if (!list)
    {
        *A = 0;
        return 0;
    }
    /* Estimation of the list's length */
    while (temp)
    {
        temp = temp->next;
        len++;
    }
    *A = (NumPair *)calloc(len, sizeof(NumPair)); // Array's allocation:
    calloc sets all data to zeros
    temp = list;
    while (temp)
    {
        data_item_list = temp->ptr;
        while (data_item_list) // processing of internal list
        {
            (*A)[i].count++;
            (*A)[i].sum += data_item_list->data;
            data_item_list = data_item_list->next;
        }
        temp = temp->next;
        i++;
    }
    return len;
}
```

### שאלה 3 (28 נק')

כתוב פונקציה שמקבלת מערך מחרוזות בשם **Infiles** עם גודלו ומחרוזת נוספת **Outfile**. כל אחת מהמחרוזות מהווה שם של קובץ טקסט ללא סיומת.

על הפונקציה להעתיק את תכני כל הקבצים שהשמות שלהם הם מחרוזות מהמערך **Infiles** בתוספת הסיומת **dat** לקובץ טקסט יחיד חדש. שם של הקובץ החדש יהיה תוכן המחרוזת **Outfile** בתוספת הסיומת **.txt**.

יש לבצע העתקת תכני הקבצים לפי סדר השמות שלהם במערך.  
יש להתחיל העתקה של כל קובץ משורה חדשה בקובץ החדש.

במידה ואחד מהקבצים המקוריים לא נפתח יש לעבור לקובץ הבא. הפונקציה תחזיר את מספר הקבצים שתוכנם הועתק בהצלחה. אם הקובץ החדש לא נפתח, על הפונקציה להחזיר 0.

דוגמא:

#### Infiles

MyFile  
MathFile  
Letters  
Students

#### Outfile

NewFile

MyFile.dat  
Name: Moshe  
Date: 24.05.2012

MathFile.dat  
 $\sin(x) + \cos(x)$   
 $2\log(x)$   
 $\cos(x) +$

Students.dat  
Amir Shai  
Doron Levi  
Tomer Cohen

NewFile.txt  
Name: Moshe  
Date: 24.05.2012  
 $\sin(x) + \cos(x)$   
 $2\log(x)$   
Amir Shai  
Doron Levi  
Tomer Cohen

הפונקציה תחזיר 3 בדוגמא.

במידה ויהיה צורך בזיכרון דינאמי זמני ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
char *add_extension (char *name, char* extension)
{
    char *file_name = (char*)malloc(strlen(name)+strlen(extension)+2);
    //+2 chars - for '.' and '\0'
    strcpy(file_name, name);
    strcat(strcat(file_name, "."), extension);
    return file_name;
}

int copy_files_to_file (char **Infiles, int n, char *Outfile)
{
    int i, count=0; // count - number of opened input files
    char ch, *file_name;
    FILE *fout, *fin;
    file_name = add_extension (Outfile, "txt"); //creation of name for
    output file
    fout = fopen(file_name, "w");
    free(file_name);
    if (!fout)
        return 0;
    for (i=0; i<n; i++)
    {
        file_name = add_extension (Infiles[i], "dat"); //creation of
        name for input file
        fin = fopen(file_name, "r");
        free(file_name);
        if (!fin)
            continue;
        count++;
        while ((ch=fgetc(fin)) != EOF)
            fputc(ch, fout);
        fputc('\n', fout);
        fclose(fin);
    }
    fclose(fout);
    return count;
}
```

**שאלה 4 (16 נק')**

נתונות שתי פונקציות הבאות:

```
unsigned fun1(unsigned value, unsigned n)
{
    value = ~value;
    value >>= n;
    value = ~value;
    return value;
}
```

```
unsigned fun2(unsigned value, unsigned n)
{
    unsigned c, displayMask = 1 << 31;
    value >>= n;
    for (c=1; c<=n; c++)
    {
        value |= displayMask;
        displayMask >>= 1;
    }
    return value;
}
```

1. האם שתי הפונקציות מבצעות אותה משימה בדרכים שונות או הן מבצעות משימות שונות?  
אם הן מבצעות אותה משימה, מה הן מבצעות ומה ההבדל בין הדרכים שבהן הן משתמשות?  
אם הן מבצעות משימות שונות, מה מבצעת `fun1` ומה מבצעת `fun2`?

**שתי הפונקציות מבצעות אותה משימה בדרכים שונות.**

הפונקציות מבצעות הזזת סיביות של מספר `value` ימינה ב-`n` סיביות. מצד שמאל המספר מתמלא ב-1-ים.

`fun1` הופכת כל סיביות של `value` ומזיזה סיביות של המספר החדש ימינה ב-`n` סיביות כך שמצד שמאל המספר מתמלא ב-0-ים. לאחר מכן, הפונקציה שוב הופכת כל סיביות של המספר. כתוצאה 0-ים שמאליים הופכים ל-1-ים וכל שאר הסיביות הופכות למצבים התחלתיים.

`fun2` קודם מזיזה סיביות של `value` ימינה ב-`n` סיביות כך שמצד שמאל המספר מתמלא ב-0-ים. לאחר מכן, מתבצעת `n` פעמים פעולת OR בין סיביות של `value` ל-`displayMask` אשר מורכב מ-0-ים וסיבית אחת 1. הסיבית 1 נמצאת בהתחלה מצד שמאל של המספר וזזה בכל איטרציה בשלב אחד ימינה. לכן כל אחד מ-`n` 0-ים שמאליים של `value` הופכת ל-1 ושאר הסיביות לא משתנות.

2. נחליף בכותרות של שתי הפונקציות unsigned value ב- int value .

(1) האם תשובות לשאלות של סעיף א' ישתנו אם ידוע ש- value תמיד יהיה חיובי? נמק.

**כן, התשובות ישתנו במקרה הכללי.**

במספר חיובי מצד שמאל נמצאת סיבית 0 אשר תהפוך ל-1 אחרי הפיכת הסיביות ב- fun1. אז value יהיה שלילי. שיטת מילוי מספר int שלילי מצד שמאל בהזזה ימינה תלוי במערכת. במיוחד ב- Visual C מספר int שלילי מצד שמאל מתמלא ב-1-ים אשר הופכים ל-0-ים אחרי ההפיכה השנייה. לכן מצד שמאל המספר יתמלא ב-0-ים.

מצד שני fun2 תיתן אותה תוצאה שבסעיף א'.  
אכן, בכל שלב שוב מתבצעת אותה פעולת OR בה משתתף 1 ולכן מצד שמאל המספר מתמלא ב-1-ים כמו בסעיף א'.

(2) האם תשובות לשאלות של סעיף א' ישתנו אם ידוע ש- value תמיד יהיה שלילי? נמק.

**לא, התשובות לא ישתנו.**

במספר שלילי מצד שמאל נמצאת סיבית 1 אשר תהפוך ל-0 אחרי הפיכת הסיביות ב- fun1. אז value יהיה חיובי ולכן יתמלא ב-0-ים מצד שמאל אשר אחרי ההפיכה השנייה יהפכו ל-1-ים כמו בסעיף א'.  
מצד שני fun2 לא תלויה בסימן של value ותיתן אותה תוצאה.

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
#define SIZEOF *

void main()
{
    printf("%d\n", sizeof EOF SIZEOF sizeof "EOF");
}
```

יש לנמק את התשובה.

16

הסבר:





**מכון טכנולוגי חולון**  
Holon Institute of Technology

ערך מספרי של EOF הוא 1. לכן טיפוסו int וגודלו 4. מחרוזת "EOF" כוללת בתוכה 4 תווים כולל '0' ולכן גודלה גם 4. SIZEOF מוחלף ב- \* וכך מקבלים  $4 * 4$  ששווה ל- 16.