מכון טכנולוגי חולון
Holon Institute of Technology

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108
סמסטר קיץ תשע"ז
מועד ב'
19.11.2017

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 מטריצה כ**מערך דו-ממדי סטטי** (לא דינאמי) של מספרים שלמים **לא שליליים** 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** המורכב מאיברי המטריצה שבכל אחד מהם הספרה הימנית (ספרת אחדות) שווה למספר העמודה של האיבר והספרה השנייה מימינה (ספרת עשרות) שווה למספר השורה של האיבר. סדר האיברים במערך החדש לא משנה.
שים לב. למספר חד-ספרתי מספר עשרות שווה ל-0. למספר בעל יותר מ-2 ספרות יש לנתח רק שתי הספרות הימניות ולא משנה מהן הספרות האחרות.

הפונקציה תעביר by reference את הכתובת של המערך החדש ותחזיר את גודלו.
במידה והמערך החדש לא מכיל אף איבר, כתובתו תהיה NULL וגודלו יהיה 0.

לדוגמא, עבור המטריצה הבאה:

	0	**1**	**2**	**3**	**4**
0	0	67	2	18	55
1	14	11	80	98	14
2	20	45	21	623	24
3	99	15	32	20	34

הפונקציה תיצור את המערך הבא:

0, 2, 11, 14, 20, 623, 24, 32, 34

ותחזיר **9**.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

הפקולטה למדעים
המחלקה למדעי המחשב
Faculty of Sciences
Department of Computer Science
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

```
#define COLS 5 //example

int two_index_array (int A[][COLS], int rows, int cols, int **p_arr)
{
	int i, j, count=0;
	*p_arr = (int *)malloc(rows*cols*sizeof(int));
	for (i=0; i<rows; i++)
		for(j=0; j<cols; j++)
			if (A[i][j]%10 == j && A[i][j]/10%10 == i)
				(*p_arr)[count++] = A[i][j];
	*p_arr = (int *)realloc(*p_arr, count*sizeof(int));
	return count;
}
```

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;
} DataItem;

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

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 **DataItem**.

על הפונקציה לבנות **מערך דו-ממדי דינאמי *A*** של מספרים שלמים.
מספר השורות ב- ***A***יהיה שווה למספר הרשימות המקושרות ב- ***L***.
איבר **A[i][0]** בשורה מס' ***i*** של המערך יהיה שווה לגודל רשימה מס' ***i*** ב-***L*** (מניחים שרשימות ב-***L*** ממוספריות מ-**0**). גודל שורה מס' ***i*** יהיה שווה ל- **A[i][0]+1**. איבר **A[i][j]** (***j*>0**) יהיה שווה לשדה **data** של איבר מס' ***j*-1** ברשימה מס' ***i*** (מניחים שאיברים ברשימה ממוספרים מ-**0**).

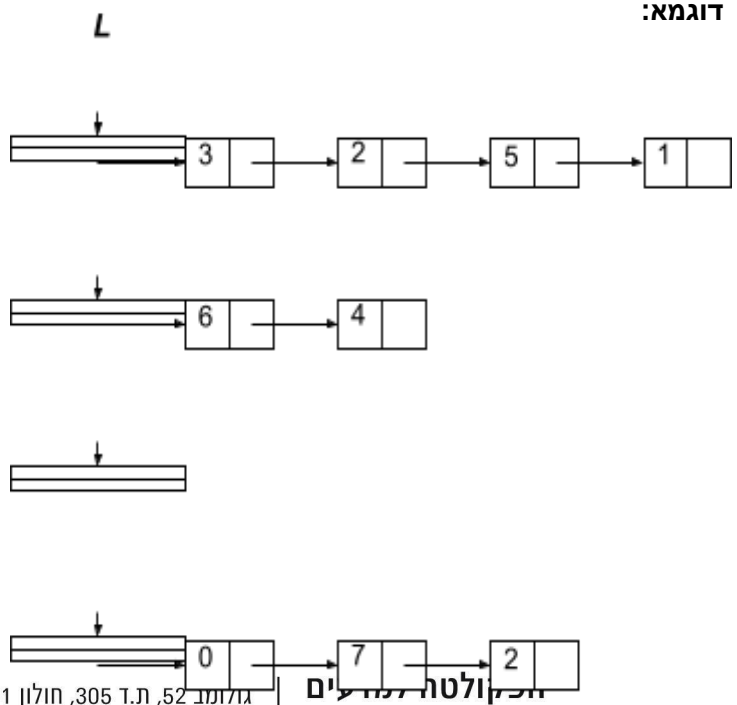
הפונקציה תחזיר את הכתובת של מערך ***A*** ותעביר by reference את מספר השורות בתוכו.
במידה ו- ***L***היא רשימה ריקה, כתובת של ***A*** תהיה NULL ומספר השורות יהיה 0.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

דוגמא:

A

4 3 2 5 1
2 6 4
0
3 0 7 2



הפונקציה תעביר **4** בדוגמא.

```
int **list_of_lists_to_two_dimensional (PtrItem *L, int *rows)
{
	PtrItem *temp = L;
	int **A, i=0, j, count;
	DataItem *data_item_list;

	*rows = 0;
	if (!L)
		return NULL;

	/* Estimation of the list's length - the number of rows */
	while (temp)
	{
		temp = temp->next;
		(*rows)++;
	}

	A = (int **)calloc(*rows, sizeof(int *)); // Allocation of array of
pointers

	temp = L;
	while (temp)
	{
		/* Estimation of the internal i-th list's length - the size
of i-th row */
		data_item_list = temp->ptr;
		count = 0;
		while (data_item_list)
		{
			data_item_list = data_item_list->next;
			count++;
		}

		A[i] = (int *)calloc(count+1, sizeof(int )); // Allocation of
i-th row

		/* Copy i-th list to i-th row */
		A[i][0] = count;
		data_item_list = temp->ptr;
		j=1;
		while (data_item_list)
		{
			A[i][j] = data_item_list->data;
			data_item_list = data_item_list->next;
			j++;
		}

		temp = temp->next;
		i++;
```

```
        }
        return A;
}
```

שאלה 3 (28 נק')

כתוב פונקציה אשר מקבלת מחרוזות **Outfile**, **Folder** ו-**Extension**. המחרוזות **Folder** ו-**Outfile** יהוו שמות של שני קבצי הטקסט. הקובץ ששמו נכלל במחרוזות **Folder** יהיה תיקית קבצים אחרים בה כל שורה היא שם קובץ (שיכול לכלול גם סיומת).

על הפונקציה להעתיק את תכני כל הקבצים בעלי סיומת הנכללת במחרוזת **Extension** שהשמות שלהם מופיעים בתיקית **Folder**, לקובץ טקסט יחיד חדש. שם של הקובץ החדש יהיה תוכן המחרוזת **Outfile**. במידע ושם הקובץ בתיקיה לא תקין (ז"א מכיל תווים אחרים חוץ מ**אותיות**, **ספרות**, **קווי תחתון**, **נקודות** ו**רווחים**) לא לנסות לפתוח את הקובץ ולעבור לשם קובץ הבא. ניתן להניח שאורך כל שם הקובץ בתיקיה אינו עולה על **30** (כולל סיומת ונקודה).

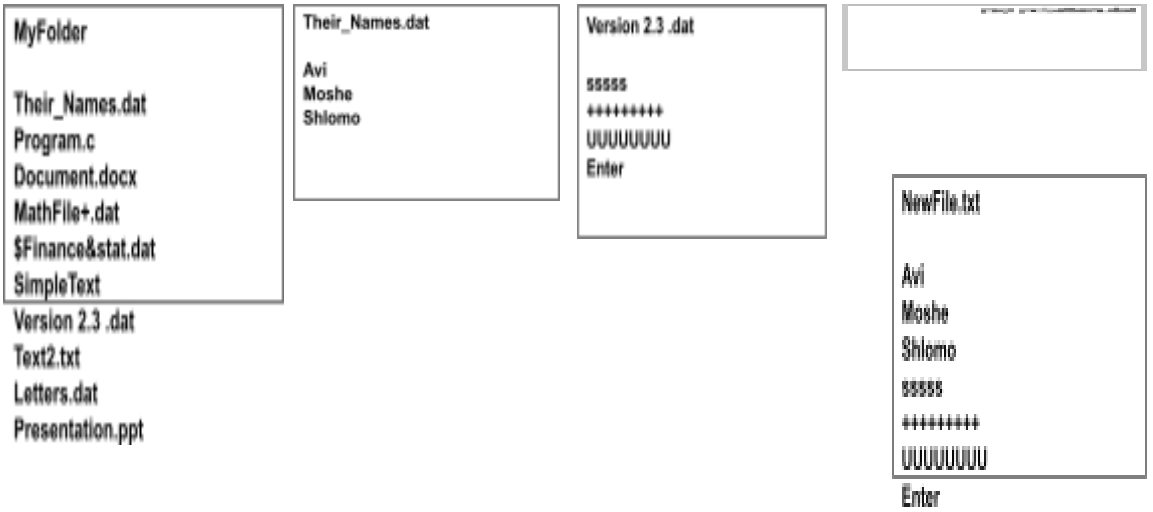
יש לבצע העתקת תכני הקבצים לפי סדר השמות שלהם בתיקיה.
יש להתחיל העתקה של כל קובץ משורה חדשה בקובץ החדש.

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**.

אפשר (אבל לא חובה) להשתמש בפונקציה סטנדרטית fgets בעלת הפורמט הבא:
fgets(מצביע ל**קובץ**, שם **מחרוזת**, **מספר** שלם)

דוגמא:

Folder = "MyFolder" , **Outfile** = "NewFile.txt" , **Extension** = "dat"



הפונקציה תחזיר **2** בדוגמא.

```
int has_extension (char *name, char* extension) {/*Auxiliary function */
	int name_len=strlen(name), ext_len=strlen(extension), i, j;
	if (name_len - ext_len <= 2)
		return 0; //must be at least 2 chars before extension: dot
and one char of name
	for (i=name_len-1, j=ext_len-1; j>=0; i--, j--)
		if (extension[j] != name[i])
			return 0;
	if (name[i] != '.')
		return 0;
	return 1;
}

int has_forbidden_chars(char *name) { /* Auxiliary function */
	int i;
	for (i=0; name[i] != '\0'; i++)
		if (ispunct(name[i]) &&  name[i] != '.' && name[i] != '_')
			return 1;
	return 0;
}

#define MAX 32 //max size of file name including '\n' and '\0'

int copy_files_to_file (char *Folder, char *Outfile, char *Extension) {
	int count=0; // count - number of opened input files
	char ch, file_name[MAX];
	FILE *fout, *fin_folder, *fin_file;
	fout = fopen(Outfile, "w");
	if (!fout) return 0;
	fin_folder = fopen(Folder, "r");
	if (!fin_folder) return 0;
	while (fgets(file_name, MAX, fin_folder)) {
		int name_len=strlen(file_name);
		if (file_name[name_len-1] == '\n')
			file_name[name_len-1] = '\0'; //removing '\n' that may
be added by fgets
		if (!has_extension(file_name, Extension) ||
has_forbidden_chars(file_name))
			continue;
		fin_file = fopen(file_name, "r");
		if (!fin_file ) continue;
		count++;
		while ((ch=fgetc(fin_file)) != EOF)
			fputc(ch, fout);
		fputc('\n', fout);
		fclose(fin_file);
	}
	fclose(fin_folder);
```

```
    fclose(fout);
    return count;
}
```

שאלה 4 (16 נק')

נתונות פונקציות הבאות:

```
#define BITS_NUM 8

char isActive(unsigned char state)
{
    return state != 0;
}

unsigned char fun(unsigned char value)
{
    unsigned char c, mask1 = 1 << (BITS_NUM-1), mask2 = 1,
                  flag = 1;
    for (c=1; c < BITS_NUM/2 && flag; c++)
        if (isActive(value & mask1) !=
                                  isActive(value & mask2))
            flag = 0;
        else
        {
            mask1 >>= 1;
            mask2 <<= 1;
        }
    return flag;
}
```

1. מה מבצעת הפונקציה **fun**? נמק.

הפונקציה מקבלת מספר בגודל ביית אחד ובודקת האם הייצוג הבינארי שלו הוא פלינדרום (מחזירה 1 אם כן, 0 אם לא).
באמצעות mask1 ו- mask2 מסורקים את המספר משמאל לימין ומימין לשמאל בהתאמה ל-4 סיביות. באמצעות פונקציית עזר isActive מבדילים את הסיביות המתאימות מצד שמאל ומצד ימין ומשווים אותן. במקרה של אי-שוויון מחזירים 0. אם מגיעים עד לאמצע הבית מחזירים 1.

2. מה תחזיר הפונקציה **fun** עבור ערכים הבאים של **value**: **1**, **126**, **129**, **254**, **255**?
נמק את כל התשובה.

1 – תחזיר 0. הייצוג הבינארי של 1 הוא 00000001 – לא פלינדרום.
126 – תחזיר 1. 2^7-1=127 ולכן הייצוג הבינארי של 127 הוא 01111111. לכן הייצוג של 126 הוא 01111110 – פלינדרום.
129 – תחזיר 1. 2^7=128 ולכן הייצוג הבינארי של 128 הוא 1000000. לכן הייצוג של 129 הוא 10000001 – פלינדרום.
254 – תחזיר 0. 2^8-1=255 ולכן הייצוג הבינארי של 255 הוא 11111111. לכן הייצוג של 254 הוא 11111110 – לא פלינדרום.

הפקולטה למדעים
המחלקה למדעי המחשב
Faculty of Sciences
Department of Computer Science
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

255 – תחזיר 1. הייצוג הבינארי של 255 הוא 11111111 – פלינדרום.

שאלה 5 (בונוס – 10 נק')
מהו הפלט למסך של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
     FILE *_fprintf = stdout;
     fprintf(_fprintf,"%.1f",1.f**"fprintf"-*"eof");
}
```

יש לנמק את התשובה.

1.0

<u>הסבר:</u>
_fprintf הוא מצביע ששווה ל- stdout ולכן fprintf ל- _fprintf שקול ל- printf (פלט למסך).
f.1 (קבוע 1. מסוג float) מכפילים (* ראשונה) בתו שנמצא בכתובת (* שנייה) של המחרוזת "fprintf" ז"א בתו הראשון שלה – 'f'. כך מקבלים קוד ASCII של 'f' כ-float. מחסירים ממנו קוד ASCII של 'e' (התו הראשון של המחרוזת "eof"). אות f מופיע בא"ב אחרי אות e ולכן ההפרש הוא 1. מסוג float. מדפיסים את התוצאה בפורמט "%.1f" – ספרה אחת אחרי הנקודה. לכן הפלט יהיה 1.0 .

הפקולטה למדעים
המחלקה למדעי המחשב
Faculty of Sciences
Department of Computer Science
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