מכון טכנולוגי חולון
Holon Institute of Technology

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108
סמסטר ב' תשע"ז
מועד ב'
24.07.2017

מרצים: ד"ר מרק קורנבליט ומר מרדכי חגיז

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף המצורף לשאלון

שאלה 1 (28 נק')

מטריצה דלילה הינה מטריצה שמרבית איבריה בעלי ערך אפס.

כתוב פונקציה שמקבלת מטריצה דלילה ***A*** **כמערך דו-ממדי סטטי** של מספרים שלמים בעל ***n*** שורות ו- ***m*** עמודות. המספר הפיסי של העמודות יינתן ע"י קבוע בשם **COLS**. יש להגדיר אותו לפני הפונקציה. בנוסף הפונקציה מקבלת מחרוזת **fileName**.

על הפונקציה ליצור **קובץ טקסט** ששמו נכלל ב- **fileName** לשמירת מטריצה ***A*** בדרך חסכונית באמצעות אחסון רק איברי המטריצה שאינם בעלי ערך אפס.

שורה מס' ***i*** בקובץ תתאים לשורה מס' ***i*** של ***A***(שורות בקובץ ממוספרות מ-**0**). בתחילת כל שורה יוכנס מספרה עם נקודה ורווח אחריו.

כל איבר של ***A*** ששונה מאפס ייוצג בקובץ כזוג המספרים: ערך של האיבר ומס' עמודה בה הוא נמצא במטריצה. מספרים בכל שורה יהיו מופרדים ברווחים.

דוגמא:

$$A = \begin{bmatrix} 0 & 0 & 4 & 0 & 7 \\ 3 & 0 & 0 & 0 & 0 \\ 9 & 1 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

תוכן הקובץ

0. 4 2 7 4

1. 3 0

2. 9 0 1 1 6 3

3.

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

```c
#define COLS 5 //example

void sparse_matrix_to_file (int A[][COLS], int rows, int cols, char
*filename)
{
	int i, j;
	FILE *fptr = fopen(filename, "w");
	if (!fptr)
		exit(1);
	for (i=0; i<rows; i++)
	{
		fprintf(fptr, "%d.", i);
		for (j=0; j<cols; j++)
			if (A[i][j] != 0)
				fprintf(fptr, " %d %d", A[i][j], j);
		fprintf(fptr, "\n");
	}
	fclose(fptr);
}
```

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;
```

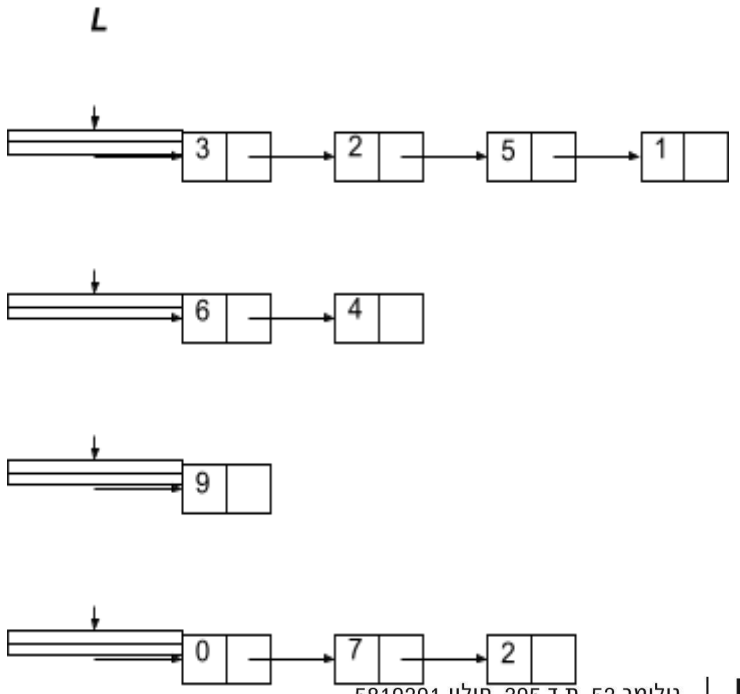
כתוב פונקציה המקבלת כפרמטרים **מערך דו-ממדי דינאמי *A*** של מספרים שלמים ומספר ***n*** המהווה **כמות שורות** במערך. בנוסף הפונקציה מקבלת מערך ***SIZE*** המורכב מגדלי שורות של ***A*** כך שאיבר מס' ***i*** של ***SIZE*** יהיה שווה לגודל שורה מס' ***i*** של ***A***.

על הפונקציה לבנות **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**. גודל רשימה מס' ***i*** יהיה שווה לגודל שורה מס' ***i*** של ***A***. שדה **data** של איבר מס' ***j*** ברשימה מס' ***i*** יהיה שווה לאיבר **[A[i][j]** ושדה **next** שלו יצביע לאיבר הבא ברשימה מס' ***i*** (מניחים שאיברים ברשימה ממוספרים מ-**0**).

הפונקציה תחזיר את **כתובת** ראש הרשימה ***L***.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

דוגמא:



A

3	2	5	1
6	4		
9			
0	7	2	

SIZE

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

4
2
1
3

```
//Auxiliary function
PtrItem *build_framework_list(int n)
{
	PtrItem *lst = NULL, *temp;
	int i;
	for (i=0; i<n; i++)
	{
		temp = (PtrItem *)malloc(sizeof(PtrItem));
		temp->next = lst;
		lst = temp;
	}
	return lst;
}

//Auxiliary function
DataItem *array_to_list(int *arr, int n)
{
	DataItem *lst = NULL, *temp;
	int i;
	for (i=n-1; i>=0; i--)
	{
		temp = (DataItem *)malloc(sizeof(DataItem));
		temp->data = arr[i];
		temp->next = lst;
		lst = temp;
	}
	return lst;
}

PtrItem *two_dimensional_to_list_of_lists (int **A, int *SIZE, int
rows_num)
{
	int i;
	PtrItem *L = build_framework_list(rows_num), *curr_ptr = L;
	for (i=0; i<rows_num; i++)
	{
		curr_ptr->ptr = array_to_list(A[i], SIZE[i]);
		curr_ptr = curr_ptr->next;
	}
	return L;
}
```

שאלה 3 (28 נק')

כתוב פונקציה שמקבלת **מחרוזת** ובונה **מערך מספרים שלמים** באופן הבא:

- כל ספרה של המחרוזת תיוצג כמספר חד-ספרתי התואם במערך (**'0'** במחרוזת יופיע כ-**0** במערך, **'1'** במחרוזת יופיע כ-**1** במערך ... **'9'** במחרוזת יופיע כ-**9** במערך);
- כל אחת מ-26 האותיות הקטנות של המחרוזת תיוצג כמספר במערך כדלקמן: **'a'** במחרוזת יופיע כ-**10** במערך, **'b'** במחרוזת יופיע כ-**11** במערך ... **'z'** במחרוזת יופיע כ-**35** במערך;
- כל אחת מ-26 האותיות הגדולות של המחרוזת תיוצג כמספר במערך כדלקמן: **'A'** במחרוזת יופיע כ-**36** במערך, **'B'** במחרוזת יופיע כ-**37** במערך ... **'Z'** במחרוזת יופיע כ-**61** במערך;
- כל תו אחר של המחרוזת לא ייוצג במערך;
- סדר האיברים במערך יהיה זהה לסדר התווים התואמים במחרוזת.

הפונקציה תחזיר **כתובת** של המערך החדש ותעביר by reference את **גודלו**.

דוגמא:

המחרוזת (משמאל לימין): **A5b$z04Y#c**

המערך (משמאל לימין): **36, 5, 11, 35, 0, 4, 60, 12**

הפונקציה תעביר **8** בדוגמא.

במידה והמערך החדש לא מכיל אף איבר, כתובתו תהיה **NULL** וגודלו יהיה **0**.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
int *string_to_array (char *str, int *size)
{
	int *arr = (int *)malloc(strlen(str)*sizeof(int)), i, j=0;
	for (i=0; str[i]!='\0'; i++)
	{
		if (isalnum(str[i]))
		{
			if (isdigit(str[i]))
				arr[j] = str[i]-'0';
			else
				if (islower(str[i]))
					arr[j] = str[i]-'a'+10;
				else // isupper(str[i])
					arr[j] = str[i]-'A'+36;
			j++;
		}
	}
	*size = j;
```

```
	arr = (int *)realloc(arr, *size*sizeof(int));
	return arr;
}
```

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;
	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?

שתי הפונקציות מבצעות אותה משימה בדרכים שונות.

2. 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**?

הפונקציות מבצעות הזזת סיביות של מספר value שמאלה ב-n סיביות. מצד ימין המספר מתמלא ב-1-ים.

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

fun2 קודם מזיזה סיביות של value שמאלה ב-n סיביות כך שמצד ימין 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 נמצא בהתחלה מצד ימין

של המספר וזז בכל איטרציה בשלב אחד שמאלה. לכן כל אחד מ- 0n-ים ימניים של value הופך ל-1 ושאר הסיביות לא משתנות.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
     int i;
     for (i=1; *("i<=i"+i); i++)
          printf("%d ", i);
}
```

יש לנמק את התשובה.

3 2 1

<u>הסבר:</u>
צריך להתייחס למחרוזת קבועה במשפט כמו לכתובת התו הראשון שלה. לכן "i<=i" + "i" זה כתובת של תו מס' i במחרוזת "i<=i" . יחד עם סימן * מצד שמאל מקבלים כבר תו מס' i עצמו. המחרוזת מורכבת מ-4 תווים שנמצאים במקומות מ-0 עד 3. פירוש לוגי של התווים הוא "אמת". במקום 4 נמצא '\0'. פירוש לוגי שלו הוא "שקר". משתנה i בלולאה מתחיל מערך 1. לכן הלולאה מתבצעת 3 פעמים ו-3 ערכים של i מודפסים. לכן הפלט יהיה

1 2 3

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