

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108

סמסטר ק' תשע"א

מועד א'

27.09.2011

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

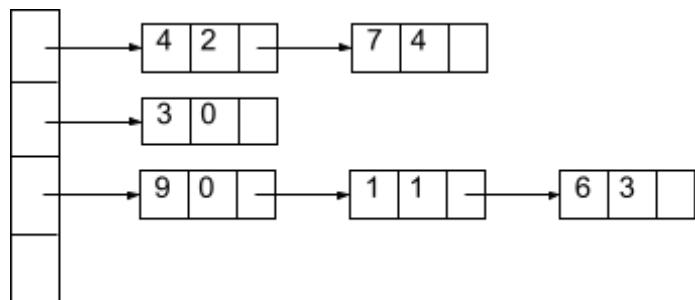
שאלה 1 (28 נק')

מטריצה דלילה הינה מטריצה שמרבית איבריה בעלי ערך אפס.

כתוב פונקציה אשר מקבלת מטריצה דלילה A המיושמת כמערך של רשימות מקושרות (היישום החסכוני באמצעות אחסון רק איברי המטריצה שאינם בעלי ערך אפס). רשימה מס' i של המערך תתאים לשורה מס' i של מטריצה A . איברי הרשימות יכללו בתוכם ערך של איבר המטריצה ומס' עמודה בה הוא נמצא במטריצה. בנוסף לכך הפונקציה תקבל כפרמטר את מספר העמודות במטריצה.

על הפונקציה לשחזר את המטריצה כמערך דו-ממדי **דינאמי**. הפונקציה תחזיר כתובת של המערך הדו-ממדי הדינאמי הבנוי.

דוגמא:

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


בדוגמא הנ"ל הפונקציה תקבל A כמערך הרשימות (יחד עם גודל שלו 4) וגם 5 כמספר העמודות ב- A .

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

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

int **lists_to_sparse_matrix (element **list_row, int rows, int cols)
{
    int **arr, i;
    element *ptr;
    arr = calloc (rows, sizeof(int *));
    for (i=0; i<rows; i++)
        arr[i] = calloc (cols, sizeof(int));
    for (i=0; i<rows; i++)
    {
        ptr = list_row[i];
        while (ptr)
        {
            arr[i][ptr->col] = ptr->data;
            ptr = ptr->ptr_next;
        }
    } // for i
    return arr;
}
```

שאלה 2 (28 נק')

כתוב פונקציה המקבלת מערך A של מספרים שלמים ומספר שלם k .
על הפונקציה לבנות שני מערכים חדשים **דינאמיים** כך שהמערך הראשון יהיה מורכב מאיברי מערך A הגדולים מ- k והמערך השני יהיה מורכב מאיברי מערך A הקטנים מ- k .
הפונקציה תעביר את הכתובות של המערכים החדשים ואת הגדלים שלהם ותחזיר את מספר האיברים של מערך A השווים ל- k .

לדוגמא, עבור המערך A הבא:

67	9	88	78	57	88	74	57	6	57
----	---	----	----	----	----	----	----	---	----

ומספר k השווה ל-57

הפונקציה תיצור את שני המערכים החדשים הבאים:

67	88	78	88	74
----	----	----	----	----

9	6
---	---

הפונקציה תעביר גם מספרים 5 ו-2 ותחזיר 3.

במידה ואחד מהמערכים החדשים לא מכיל אף איבר, כתובתו תהיה NULL וגודלו יהיה 0.
ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

int large_small_num (int *a, int size, int k, int **array_large, int **array_small,
                    int *size_large, int *size_small)
{
    int count_large = 0, count_small = 0, equal_num, i;
    /* Estimation of array sizes */
    *size_large = *size_small = 0;
    for (i=0; i<size; i++)
        if (a[i] > k)
            (*size_large)++;
        else
            if (a[i] < k)
                (*size_small)++;
    equal_num = size - *size_large - *size_small;
    /* Allocation of arrays */
    if (*size_large)
        *array_large = malloc(*size_large*sizeof(int));
    else
        *array_large = NULL;
    if (*size_small)
        *array_small = malloc(*size_small*sizeof(int));
    else
        *array_small = NULL;
    if (equal_num == size)
        return equal_num;
    /* Filling arrays */
    for (i=0; i<size; i++)
        if (a[i] > k)
        {
            (*array_large)[count_large] = a[i];
            count_large++;
        }
        else
            if (a[i] < k)
            {
                (*array_small)[count_small] = a[i];
                count_small++;
            }
    return equal_num;
}

```

שאלה 3 (28 נק')

כתוב פונקציה המקבלת מחרוזת ומעבירה מספר שלם (חיובי או שלילי או אפס) אשר יהיה ערכה המספרי.

המחרוזת אמורה להיות מורכבת מספרות פרט לתו הראשון שלה אשר יכול להיות ספרה או '-' או '+'.

הפונקציה תחזיר 1 אם ניתן להמיר את המחרוזת למספר שלם. אם תוכן המחרוזת יהיה לא תקין הפונקציה תחזיר 0. במקרה האחרון לא משנה מה הפונקציה תעביר.

לדוגמא, עבור המחרוזות "1250", "-569", "+1896", "007", "0" הפונקציה תעביר את המספרים 1250, -569, 1896, 7, 0 בהתאמה ותחזיר 1.

עבור המחרוזות "12-50", "5.69", "98x3", "-+", וגם עבור מחרוזת ריקה הפונקציה תחזיר 0.

לא להשתמש בפונקציה סטנדרטית **atoi** המממשת את האלגוריתם הנ"ל!

```

int string_to_num (char *str, int *num)
{
    int len = strlen (str), i, tens = 1;
    *num = 0;
    if (len==0 || len==1 && !isdigit(str[0]))
        return 0;
    for (i = len-1; i>0; i--)
        if (isdigit(str[i]))
        {
            *num += (str[i]-'0') * tens;
            tens *= 10;
        }
        else
            return 0;
    if (isdigit(str[0]))
        *num += (str[0]-'0') * tens;
    else
        if (str[0] == '-')
            *num *= -1;
        else
            if (str[0] != '+')
                return 0;
    return 1;
}

```

שאלה 4 (16 נק')
נתונה התוכנית הבאה:

```

#include <stdio.h>

struct first
{
    int x;
    double y;
};

typedef union
{
    int x;
    int *p;
} second;

void main()
{
    // need word struct
    first st1, st2 = {5,8.9}, *ptr;
    second un;
    ptr = &st1;
    scanf ("%d", &st1.x);
    scanf ("%lf", ptr->y); // running error: need &

    if (st1 == st2) // '==' : illegal for struct

```

```

        printf ("Structures are equal\n");
un.p = &st1.x;
(*un.p)++;
printf ("%d\n", *un.p);
un.x = 3;
// running error:
(*un.p)--; // un.p is destroyed after changing un.x
printf ("%d\n", *un.p);
}

```

התוכנית שגויה!
'ש לתאר את כל השגיאות (קומפילציה וריצה).

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

```

#include <stdio.h>
#include <string.h>

void main()
{
    char *str = "str";
    printf ("%d", strcmp("strcpy(str,str)","strlen(str)"));
}

```

1-