

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108

סמסטר ק' תשע"ב

מועד א'

3.10.2012

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת מערך **A** בעל **n** מספרים טבעיים.
על הפונקציה לבנות מערך דו-ממדי דינאמי בעל **n** שורות עם גדלים שונים כך שגודל שורה מספר **i** יהי שווה לתוכן איבר מספר **i** של מערך **A**.
איבר מספר **j** בכל שורה יהי שווה ל- 2^j .

דוגמא:

מערך חדש
1 2 4 8
1 2
1
1 2 4
1 2 4 8 16

A=4,2,1,3,5

הפונקציה תחזיר כתובת של המערך החדש.
ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

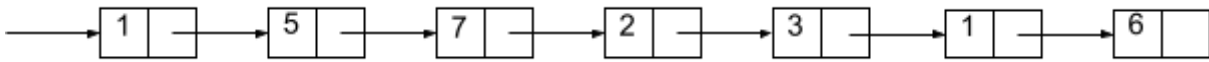
```
int **powers_of_two (int *A, int n)
{
    int **arr, i, j;
    arr = (int **)calloc(n, sizeof(int *));
    for (i=0; i<n; i++)
    {
        arr[i] = (int *)calloc(A[i], sizeof(int));
        arr[i][0] = 1;
        for (j=1; j<A[i]; j++)
            arr[i][j] = arr[i][j-1]*2;
    }
    return arr;
}
```

}

שאלה 2 (28 נק')

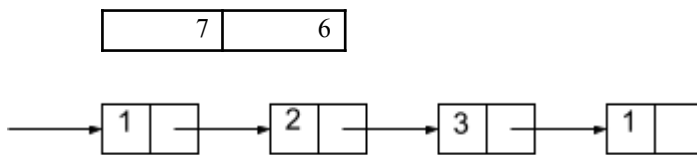
כתוב פונקציה המקבלת רשימה מקושרת של מספרים שלמים ומספר שלם k . על הפונקציה לבנות מערך דינאמי ורשימה מקושרת חדשה כך שהמערך יהיה מורכב מאיברי הרשימה המקורית הגדולים מ- k והרשימה החדשה תהיה מורכבת מאיברי הרשימה המקורית הקטנים מ- k . הפונקציה תעביר את המצביעים לאיברים הראשונים של המערך והרשימה החדשה וגם ואת גודל המערך. הפונקציה תחזיר את מספר האיברים של הרשימה המקורית השווים ל- k .

לדוגמא, עבור הרשימה הבאה:



ומספר k השווה ל-5

הפונקציה תיצור את המערך והרשימה הבאים:



הפונקציה תעביר גם 2 ותחזיר 1.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

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

int big_small_num (element *lst, int k,
                  int **array_big, int *size_big, element **lst_small)
{
    int count_big = 0, equal_num = 0;
    element *ptr = lst, *ptr_small;

    *lst_small = (element *)malloc(sizeof(element)); // allocation
    of dummy element
    ptr_small = *lst_small;
    /* Building of list and estimation of array's size */
    *size_big = 0;
    while (ptr)
    {
        if (ptr->data > k)
            (*size_big)++;
        else
            if (ptr->data < k)
            {
                ptr_small->ptr_next =
                    (element *)malloc(sizeof(element));
                ptr_small = ptr_small->ptr_next;
                ptr_small->data = ptr->data;
            }
            else
                equal_num++;
        ptr = ptr->ptr_next;
    }
    ptr_small->ptr_next = NULL;
```

```

/* deletion of dummy element */
ptr_small = *lst_small;
*lst_small = (*lst_small)->ptr_next;
free (ptr_small);
/* Allocation of array */
if (*size_big)
    *array_big = (int*)malloc(*size_big*sizeof(int));
else
{
    *array_big = NULL;
    return equal_num;
}
/* Filling array */
while (lst)
{
    if (lst->data > k)
    {
        (*array_big)[count_big] = lst->data;
        count_big++;
    }
    lst = lst->ptr_next;
}
/* ----- */
return equal_num;
}

```

שאלה 3 (28 נק')

כתוב פונקציה (void) שמקבלת מחרוזת ומעבדת אותה כדלקמן:

- כל אות קטנה פרט לאותיות מ-'a' עד 'f' תהפוך לאות הגדלה התואמת;
- כל אות גדלה פרט לאותיות מ-'A' עד 'F' תהפוך לאות הקטנה התואמת;
- כל ספרה תהפוך לסדרת תווי '1', '2', '3'... (משמאל לימין) כך שאורך הסדרה יהי שווה לספרה;
- כל אות קטנה מ-'a' עד 'f' תתפרש כספרה הקסדצימלית (מ-10 עד 15 בהתאמה) ותהפוך לסדרת תווי '1', '2', '3'...'9', 'a'... (משמאל לימין) כך שאורך הסדרה יהי שווה לספרה הקסדצימלית (האותיות בסדרה יהיו קטנות);
- כל אות גדלה מ-'A' עד 'F' תתפרש כספרה הקסדצימלית (מ-10 עד 15 בהתאמה) ותהפוך לסדרת תווי '1', '2', '3'...'9', 'A'... (משמאל לימין) כך שאורך הסדרה יהי שווה לספרה ההקסדצימלית (האותיות בסדרה יהיו גדולות);
- כל תו אחר הפונקציה תסיר מהמחרוזת.

דוגמא:

B\$Q0#3Me	מחרוזת לפני עיבוד:
123456789ABQ123m123456789abcde	המחרוזת אחרי עיבוד:

ניתן להניח שהגודל הפיסי של המחרוזת הנתונה הוא מספיק למחרוזת המעובדת.

אפשר (אבל לא חובה) להשתמש במחרוזת עזר. במקרה זה ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

void string_transform (char *str)
{
    char *temp, c;
    int i, count=0;

    temp = (char *)malloc(strlen(str) + 1);
    strcpy (temp, str);
    for (i=0; temp[i]!='\0'; i++)
    {
        if (isxdigit(temp[i]))
        {
            if (isdigit(temp[i]))
                for (c='1'; c<=temp[i]; c++)
                {
                    str[count] = c;
                    count++;
                } // for c
            else // temp[i] is between 'a' and 'f' or 'A' and 'F'
            {
                for (c='1'; c<='9'; c++)
                {
                    str[count] = c;
                    count++;
                } // for c
                if (islower(temp[i]))
                    c = 'a';
                else
                    c = 'A';
                for (; c<=temp[i]; c++)
                {
                    str[count] = c;
                    count++;
                } // for c
            } // else
        } // if (isxdigit(temp[i]))
        else // not isxdigit(temp[i])
            if (isalpha(temp[i]))
            {
                if (islower(temp[i]))
                    str[count] = toupper(temp[i]);
                else
                    str[count] = tolower(temp[i]);
                count++;
            }
    } // for i
    str[count] = '\0';
    free (temp);
}

```

שאלה 4 (16 נק')
נתונה התוכנית הבאה:

```

#include <stdio.h>

int *fun1 ();
int fun2 ();
int fun3 ();

```

```

void main()
{
    int (*fun4) (), x;

    x = *fun1();
    printf ("%d\n", x);
    x = *fun1();
    printf ("%d\n", x);
    fun4 = fun2;
    x = (*fun4)();
    printf ("%d\n", x);
    x = fun4();
    printf ("%d\n", x);
    fun4 = fun3;
    x = (*fun4)();
    printf ("%d\n", x);
    x = fun4();
    printf ("%d\n", x);
}

int *fun1 ()
{
    static int x;
    x++;
    return &x;
}

int fun2 ()
{
    static int x=1;
    x++;
    return x;
}

int fun3 ()
{
    int x=1;
    x++;
    return x;
}

```

מהו הפלט של התוכנית?

1
2
2
3
2
2

האם יהיה תקין למחוק את המילה `static` בפונקציה `fun1`? נמק אם לא (יש לסמן את כל הנימוקים), הראה איך ישתנה הפלט אם כן.

לא תקין כי (א) הפונקציה תחזיר כתובת של משתנה לוקלי שלא קיים מעבר לפונקציה ו-(ב) משתנה x יהיה לא מאותחל

האם יהיה תקין למחוק את המילה `static` בפונקציה `fun2`? נמק אם לא (יש לסמן את כל הנימוקים), הראה איך ישתנה הפלט אם כן.

תקין.

פלט:

1
2
2
2
2
2
2

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

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

void main()
{
    puts(strcmp("puts", "strcmp()")?"puts":"strcmp()");
}
```

puts