

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108

סמסטר קיץ תשע"ד
 מועד ב'
 9.11.2014

מרצים: ד"ר מרק קורנבליט ומר לאוניד קוגל

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדפים המצורפים לשאלון

שאלה 1 (28 נק')

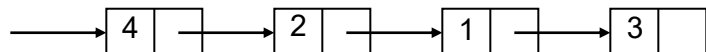
כתוב פונקציה אשר מקבלת רשימה מקושרת בעלת מספרים טבעיים.
 על הפונקציה לבנות מערך דו-ממדי **דינאמי** בעל שורות עם גדלים שונים כך שמספר השורות
 במערך יהיה שווה למספר האיברים ברשימה.
 נסמן גודל שורה מס' i במערך כ- n_i ומספר שנמצא באיבר מס' i מתחילת הרשימה (מניחים
 שאיברי הרשימה ממוספרים כאיברי המערך מ-0) כ- x_i . לכל שורה n_i יהיה שווה ל- x_i+1 .
 איבר מס' j בשורה מס' i ($0 \leq j \leq n_i-2$) יהיה שווה ל- 2^j .
 האיבר האחרון בכל שורה יהיה שווה ל-0.

דוגמא:

מערך חדש

1 2 4 8 0
 1 2 0
 1 0
 1 2 4 0

רשימה מקורית



הפונקציה תחזיר את הכתובת של המערך החדש ותעביר by reference את מספר השורות
 בתוכו.
 בדוגמא הנ"ל הפונקציה תעביר 4.

יש להגדיר את הטיפוס של איברי הרשימה המקורית מעל הפונקציה.

ניתן להניח שיש בזיכרון מספיק מקום להקצאת המערך והרשימה המקורית היא לא ריקה.

```

typedef struct item {
    int data;
    struct item *next;
} Item;

int list_size (Item *lst)
{
    int size = 0;
    while (lst)
    {
        size++;
        lst = lst->next;
    }
    return size;
}

int **list_to_powers_of_two (Item *lst, int *rows_num)
{
    int **arr, i, j, size = list_size(lst);
    arr = (int **)calloc(size, sizeof(int *));
    for (i=0; i<size; i++)
    {
        arr[i] = (int *)calloc(lst->data+1, sizeof(int));
        arr[i][0] = 1;
        for (j=1; j < lst->data; j++)
            arr[i][j] = arr[i][j-1]*2;
        arr[i][j] = 0;
        lst = lst->next;
    }
    *rows_num = size;
    return arr;
}

```

שאלה 2 (28 נק')

קטע סגור $[a, b]$ הוא קבוצת כל הנקודות על ישר המספרים אשר נמצאות בין שתי הנקודות **a** ו-**b** (הנקראות **קצוות הקטע**) כולל **a** ו-**b** עצמם.

כתוב פונקציה המקבלת שני מספרים שלמים (גדולים מ-1) **a** ו-**b** שהם קצוות של הקטע הסגור. על הפונקציה למצוא את **המספרים הראשוניים** (המתחלקים רק בעצמם וב-1) הנמצאים בקטע **$[a, b]$** ולהעתיק אותם למערך דינאמי בשם **primes**. הפונקציה תעביר by reference את הכתובת של **primes** ותחזיר את גודלו. במידה ו-**primes** לא מכיל אף איבר, כתובתו תהיה NULL וגודלו יהיה 0.

לדוגמא, עבור מספרים **a** השווה ל-4 ו-**b** השווה ל-17

הפונקציה תיצור את המערך הבא:

5	7	11	13	17
---	---	----	----	----

ותחזיר 5.

ניתן להניח שיש בזיכרון מספיק מקום להקצאת המערך.

```

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

int primes_interval (int a, int b, int **primes)
{
    int size = 0, num, i=0;
    /* Estimation of array's size */
    for (num=a; num<=b; num++)
        if (is_prime(num))
            size++;
    if (size == 0)
    {
        *primes = NULL;
        return 0;
    }
    /* Allocation of array */
    *primes = (int *)calloc(size, sizeof(int));
    /* Filling of array */
    for (num=a; num<=b; num++)
        if (is_prime(num))
        {
            (*primes)[i] = num;
            i++;
        }
    return size;
}

```

שאלה 3 (28 נק')

כתוב פונקציה (void) שמקבלת מחרוזת **str** ומחרוזת **fileName**. על הפונקציה ליצור קובץ טקסט ששמו נכלל ב- **fileName** ולהעתיק את התוכן של **str** לקובץ כדלקמן:

- כל אות קטנה פרט לאותיות מ-'a' עד 'f' תהפוך לאות הגדולה התואמת;
- כל אות גדולה פרט לאותיות מ-'A' עד 'F' תהפוך לאות הקטנה התואמת;
- כל ספרה תהפוך לסדרת תווי '1', '2', '3',... '9' (משמאל לימין) כך שאורך הסדרה יהיה שווה לספרה;
- כל אות קטנה מ-'a' עד 'f' תתפרש כספרה הקסדצימלית (מ-10 עד 15 בהתאמה) ותהפוך לסדרת תווי 'a',... '9', '0', '1', '2', '3',... '9' (משמאל לימין) כך שאורך הסדרה יהיה שווה לספרה הקסדצימלית (האותיות בסדרה יהיו קטנות);
- כל אות גדולה מ-'A' עד 'F' תתפרש כספרה הקסדצימלית (מ-10 עד 15 בהתאמה) ותהפוך לסדרת תווי 'A',... '9', '0', '1', '2', '3',... '9' (משמאל לימין) כך שאורך הסדרה יהיה שווה לספרה הקסדצימלית (האותיות בסדרה יהיו גדולות);
- כל תו אחר הפונקציה לא תעתיק לקובץ.

דוגמא:

B\$q0#3Me	המחרוזת המקורית str :
123456789ABQ123m123456789abcde	תוכן הקובץ אחרי קריאה לפונקציה:

```

void transform_string_to_file (char *str, char *fileName)
{
    char c;
    int i;
    FILE *fptr = fopen(fileName, "w");
    if( !fptr )
    {
        printf("File cannot be opened!\n");
        return;
    }
    for (i=0; str[i]!='\0'; i++)
    {
        if (isxdigit(str[i]))
        {
            if (isdigit(str[i]))
                for (c='1'; c<=str[i]; c++)
                    fputc(c, fptr);
            else // temp[i] is between 'a' and 'f' or between
                // 'A' and 'F'
            {
                for (c='1'; c<='9'; c++)
                    fputc(c, fptr);
                if (islower(str[i]))
                    c = 'a';
                else
                    c = 'A';
                for (; c<=str[i]; c++)
                    fputc(c, fptr);
            } // else
        } // if (isxdigit(temp[i]))
        else // not isxdigit(temp[i])
            if (isalpha(str[i]))
                if (islower(str[i]))
                    fputc(toupper(str[i]), fptr);
                else
                    fputc(tolower(str[i]), fptr);
    } // for i
    fclose(fptr);
}

```

שאלה 4 (16 נק')
נתונה התוכנית הבאה:

```

#include <stdio.h>
#include <malloc.h>

#define SIZE 2

typedef struct
{
    int p[SIZE];
} type1;

typedef struct
{
    int *p;
} type2;

```

```

void fun1 ( type1, int *, int,
            void( * )(int *, int *, int) );
void fun2 ( type2, int *, int,
            void( * )(int *, int *, int) );
void fun3 (int *, int *, int);
void fun4 (int *, int *, int);

void main()
{
    int p[] = {10, 20}, i;
    type1 a = {3, 4}; // 1
    type2 b;
    b.p = (int *)calloc(SIZE, sizeof(int));
    b.p[0] = 5;
    b.p[1] = 6;
    fun1(a, p, SIZE, fun3); // 2
    fun2(b, p, SIZE, fun4); // 3
    for (i=0; i<SIZE; i++)
        printf ("%d\n%d\n", a.p[i], b.p[i]);
}

void fun1 ( type1 a, int *p, int n,
            void( *pf )(int *, int *, int) )
{
    pf(a.p, p, n);
}

void fun2 ( type2 b, int *p, int n,
            void( *pf )(int *, int *, int) )
{
    pf(b.p, p, n);
}

void fun3 (int *p1, int *p2, int n)
{
    int i;
    for (i=0; i<n; i++)
        p1[i] = 3*p2[i];
}

void fun4 (int *p1, int *p2, int n)
{
    int i;
    for (i=0; i<n; i++)
        p1[i] = 4*p2[i];
}

```

א. מהו פלט התוכנית?

3
40
4
80

ב. האם יהיה תקין לא לאתחל **a** בשורה המסומנת בהערה כ-1 (// 1)? נמק אם לא, הראה איך ישתנה הפלט אם כן.

לא יהיה תקין. **a** הוא struct ושדה שלו מערך **k** מועבר לפונקציה **by value**. פונקציה **fun1** מעדכנת את העתק של המערך המקורי. לכן אם **a** לא מאותחל אז אין מה להדפיס בסוף.

ג. האם ישתנה הפלט אחרי מחיקת השורה המסומנת בהערה כ-2 (// 2)? נמק אם לא, הראה איך ישתנה הפלט אם כן.

הפלט לא ישתנה. פונקציה **fun1** מעדכנת את העתק של **a** ולא **a** עצמו. לכן ללא קריאתה אותו תוכן מקורי של **a** יודפס.

ד. האם ישתנה הפלט אחרי מחיקת השורה המסומנת בהערה כ-3 (// 3)? נמק אם לא, הראה איך ישתנה הפלט אם כן.

הפלט ישתנה:

3
5
4
6

שאלה 5 (בנוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
    printf( "%d",strcmp("+2&&3","--")*strcmp("-x==8","+234")+
            strcmp("a/str","(4+6)/2")*strcmp("(y+z)*5","apple") );
}
```

-2