

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

מעבדה למחשבים אישיים 61105

סמסטר ק' תשס"ט

מועד ב'

3.11.2009

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

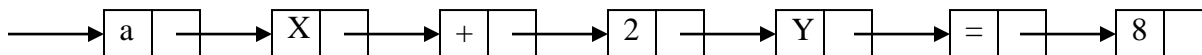
אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

שאלה 1 (28 נק')

כתוב פונקציה אשר מקבלת רשימה מקושרת של תווים ובונה שני מערכים דינאמיים מהטיפוס `int`. המערך הראשון יהיה מורכב מקודי ASCII של אותיות של הרשימה המקורית. המערך השני תכלול בתוכו מספרים חד-ספרתיים התואמים לתווים של הרשימה אשר מציגים ספרות ('0' ברשימה יופיע כ-0 במערך, '1' ברשימה יופיע כ-1 במערך וכו'). על הפונקציה להעביר כתובות המערכים וגדלים שלהם ולהחזיר הכמות של האיברים ברשימה המקורית.

דוגמא:

רשימה מקורית:



מערך ראשון:

97 88 89

מערך שני:

2 8

הפונקציה תחזיר 7 בדוגמא.

במידה ואחד ממערכים החדשים לא מכיל אף איבר, כתובתו תהיה `NULL`. ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

typedef struct element
{
    char data;
    struct element *ptr_next;
} element;

int two_arrays (element *list, int **let_array, int **dig_array, int *let_size, int *dig_size)
{
    element *curr_ptr = list;
    int list_size = 0, let_count = 0, dig_count = 0;

    *let_size = *dig_size = 0;
    while (curr_ptr)
    {
        if (isalpha(curr_ptr->data))
            (*let_size)++;
        else
            if (isdigit(curr_ptr->data))
                (*dig_size)++;
        curr_ptr = curr_ptr->ptr_next;
        list_size++;
    }

    if (*let_size)
        *let_array = malloc(*let_size);
    else
        *let_array = NULL;
    if (*dig_size)
        *dig_array = malloc(*dig_size*sizeof(int));
    else
        *dig_array = NULL;
    if (*let_size==0 && *dig_size==0)
        return list_size;

    curr_ptr = list;
    while (curr_ptr)
    {
        if (isalpha(curr_ptr->data))
        {
            (*let_array)[let_count] = curr_ptr->data;
            let_count++;
        }
        else
            if (isdigit(curr_ptr->data))
            {
                (*dig_array)[dig_count] = curr_ptr->data - '0';
                dig_count++;
            }
        curr_ptr = curr_ptr->ptr_next;
    }
    return list_size;
}

```

שאלה 2 (28 נק')

כתוב פונקציה (void) שמימשת הכפלת מטריצות. על הפונקציה לקבל שני מערכים דו-ממדיים כך שאחד מהם יהי סטטי והשני יהי דינאמי, להקצות מקום למערך החדש, לחשב מכפלת מטריצות, ולהעביר כתובת של המערך החדש באמצעות שיטת call by reference. אם $A=(a_{ij})$ היא מטריצה חצומ ו- $B=(b_{jk})$ היא מטריצה קאח אז $C=AB$ היא מטריצה קאח שלכל איבר c_{ik} שלה

$$c_{ik} = \sum_{j=1}^n a_{ij} b_{jk}$$

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
#define N 5 //example
```

```
void matr_product (int a[][N], int **b, int m, int n, int p, int ***c)
{
    int i,j,k;

    *c = calloc(m, sizeof(int *));
    for (i=0; i<m; i++)
        (*c)[i]=calloc(p, sizeof(int));
    for (i=0; i<m; i++)
        for (k=0; k<p; k++)
        {
            (*c)[i][k]=0;
            for (j=0; j<n; j++)
                (*c)[i][k]+=a[i][j]*b[j][k];
        }
}
```

שאלה 3 (28 נק')

כתוב פונקציה אשר מקבלת מחרוזת ומצמצמת כמות רווחים בין מילים בתוכה עד רווח אחד.

דוגמא:

I	live in	Israel	מחרוזת לפני עיבוד:
I	live in	Israel	המחרוזת אחרי עיבוד:

ניתן להניח שהתו הראשון והתו האחרון של המחרוזת הם לא רווחים.
אפשר (אבל לא חובה) להשתמש במחרוזת עזר.

```
void delete_space (char *str)
{
    char *temp;
    int len = strlen (str), i, j;

    temp = malloc(len+1);
    temp[0] = str[0];
    for (i=1, j=0; i<=len; i++)
        if ((str[i]!=' ') || (str[i-1]!=' '))
        {
            j++;
            temp[j] = str[i];
        }
    strcpy (str, temp);
    free (temp);
}
```

```
#include <stdio.h>

#define SIZE 3
#define A 5

int sum_arr (int *a, int SIZE); error: SIZE - constant
void fun1 (int b);
void fun2 (int x);

int a=0;

void main()
{
    int a[] = {1, 2, 3}, sum, x;

    sum = sum_arr (a, 3);
    a++; error: '++' needs l-value
    A++; error: '++' needs l-value
    fun1 (A);
    scanf ("%d", &x);

    if (x > 0)
    {
        int x;
        x = 1;
        printf ("%d\n", x);
        int y = 2; error: declaration in the middle of the block
        printf ("%d\n", y);
    }
    fun2 (x);
    printf ("%d\n", x);
}

int sum_arr (int *a, int SIZE) error: SIZE - constant
{
    int i, sum=0;
    for (i=0; i<SIZE; i++)
        sum += a[i];
} warning: sum_arr must return a value

void fun1 (int b)
{
    b++;
```

```

    a++;
    A++; error: '++' needs l-value
    printf ("%d %d %d\n", b, a, A);
}

void fun2 (int x)
{
    int x = 3; error: redefinition of formal parameter x
    x++;
    printf ("%d\n", x);
}

```

התוכנית שגויה!
 'ש לתאר את כל השגיאות.

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

```

#include <stdio.h>
void main()
{
    printf ("%d", sizeof"+"sizeof"+"");
}

```