

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

מעבדה למחשבים אישיים 61105

סמסטר ק' תשס"ח

מועד א'

19.10.2008

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר פרט לדף מצורף לשאלון

שאלה 1 (30 נק')

כתוב פונקציה שמקבלת מטריצה ריבועית A ובונה מטריצה משוחלפת שלה A^T (transpose) אשר מתקבלת על ידי החלפת השורות בעמודות של המטריצה המקורית A .

דוגמא:

$$A = \begin{bmatrix} 1 & 3 & -2 \\ 2 & 8 & 1 \\ 4 & 7 & -7 \end{bmatrix} \rightarrow A^T = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 7 \\ -2 & 1 & -7 \end{bmatrix}$$

על הפונקציה לקבל מטריצה A כמערך דו-ממדי סטטי, להקצות מקום למטריצה החדשה A^T (דינאמית!), למלא A^T , ולהחזיר כתובת של המערך החדש. ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
#define MAX 10
int **transpose (int a[][MAX], int size)
{
    int **at, i, j;

    at = calloc (size, sizeof(int*));
    for (i=0; i<size; i++)
        at[i] = calloc (size, sizeof(int));
    for (i=0; i<size; i++)
        for (j=0; j<size; j++)
            at[i][j] = a[j][i];

    return at;
}
```

שאלה 2 (25 נק')

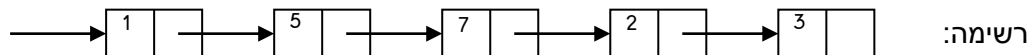
כתוב פונקציה **Find_Last_Vowel** המקבלת מחזורות ומחזירה את **הכתובת** של ה - Vowel האחרון שהיא נתקלת בו בסריקה משמאל לימין. קבוצת ה - Vowels מכילה את האותיות הבאות: A,a;E,e;I,i;O,o;U,u. הפונקציה תחזיר NULL אם אין במחזורות אף Vowel.

```
char *FindLastVowel (char *str)
{
    char *vowel = NULL;
    for (; *str != '\0'; str++)
        if (*str == 'A' || *str == 'a' ||
            *str == 'E' || *str == 'e' ||
            *str == 'I' || *str == 'i' ||
            *str == 'O' || *str == 'o' ||
            *str == 'U' || *str == 'u')
            vowel = str;
    return vowel;
}
```

שאלה 3 (30 נק')

כתוב פונקציה שמקבלת רשימה מקושרת חד-כיוונית. על הפונקציה לבנות מערך דינאמי כך שערכי איברי המערך יהיו תכני איברי הרשימה המופיעים בסדר הפוך ביחס להופעתם ברשימה.

דוגמא:



מערך: 3 2 7 5 1

על הפונקציה להחזיר כתובת של המערך ולהעביר את גודלו לתוכנית הראשית.

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

list_type *list_to_array (element *list, int *size)
{
    list_type *array;
    element *curr_point=list;
    int i;

    *size = 0;
    while (curr_point)
    {
        curr_point = curr_point -> ptr_next;
        (*size)++;
    }
}
```

```

array = calloc (*size, sizeof(list_type));
curr_point = list;
for (i=*size-1; i>=0; i--)
{
    array[i] = curr_point -> data;
    curr_point = curr_point -> ptr_next;
}
return array;
}

```

שאלה 4 (15 נק')
מה עושה התוכנית הבאה?

```

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

int sod (int, char *);

void main()
{
    char str[50];

    gets (str);
    while ( !sod(5, str) )
        gets (str);
}

int sod (int m, char *str)
{
    static int n;

    if (strlen(str) % 2)
    {
        n++;
        printf ("%d.  %s\n", n, str);
    }
    if (n == m)
        return 1;
    return 0;
}

```

התוכנית קולטת מחרוזות ומדפיסה רק מחרוזות בעלות אורך אי-זוגי. התוכנית מסתיימת כאשר מספר מחרוזות בעלות אורך אי-זוגי מגיע לחמש.

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

```
#include <stdio.h>

void main()
{
    printf("%d", '-''-''-');
}
```

0

בהצלחה!