



מכון טכנולוגי חולון
Holon Institute of Technology

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108

סמסטר ב' תשע"א

מועד א'

10.06.2011

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

שאלה 1 (28 נק')

א. (20 נק')

כתוב פונקציה שמקבלת מטריצה סימטרית A ($a_{ij}=a_{ji}$) כמערך דו-ממדי סטטי. על הפונקציה לבנות מערך דו-ממדי דינאמי בעל שורות עם גדלים שונים אשר יישם מטריצה A בדרך חסכונית באמצעות אחסון רק משולש אחד של מטריצה A .

דוגמא:

מערך חדש

1
2 6
3 4 5

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 6 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

הפונקציה תחזיר כתובת של המערך החדש.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
#define MAX 5 //example
```

```
int **stat_to_dyn_sym_matrix (int a[][MAX], int size)
{
    int **a_dyn, i, j;
    a_dyn = calloc (size, sizeof(int *));
    for (i=0; i<size; i++)
        a_dyn[i] = calloc (i+1, sizeof(int));
    for (i=0; i<size; i++)
        for (j=0; j<=i; j++)
            a_dyn[i][j] = a[i][j];
    return a_dyn;
}
```

ב. (8 נק')

כתוב פונקציה המממשת פנייה לאיבר a_{ij} של מטריצה סימטרית A המיושמת באמצעות מערך דינאמי בסעיף (א). הפונקציה תקבל כתובת של המערך הדינאמי וקואורדינטות האיבר.

```
int sym_matr_element (int **a_dyn, int i, int j)
{
    if (i>=j)
        return a_dyn[i][j];
    else
        return a_dyn[j][i];
}
```

שאלה 2 (28 נק')

כתוב פונקציה המקבלת מערך של מספרים שלמים. על הפונקציה לבנות רשימה מקושרת המורכבת ממספרים שמופיעים רק פעם אחת במערך המקורי.

לדוגמא, עבור המערך המקורי הבא:

57	6	57	74	88	57	74	88	9	57
----	---	----	----	----	----	----	----	---	----

הפונקציה תיצור את הרשימה הבאה:



הפונקציה תחזיר כתובת של האיבר הראשון ברשימה.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

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

```

element *alone_elements (int *a, int n)
{
    element *lst, *curr_point;
    int i, j, found_equal;

    lst = malloc(sizeof(element)); // allocation of dummy element
    curr_point = lst;
    for (i=0; i<n; i++)
    {
        found_equal = 0;
        for (j=0; j<i; j++)
            if (a[i] == a[j])
            {
                found_equal = 1;
                break; // for j
            }
        if (found_equal)
            continue; // for i
        for (j=i+1; j<n; j++)
            if (a[i] == a[j])
            {
                found_equal = 1;
                break; // for j
            }
        if (!found_equal)
        {
            // processing of new element
            curr_point->ptr_next = malloc(sizeof(element));
            curr_point = curr_point->ptr_next;
            curr_point->data = a[i];
        }
    } // for i
    curr_point->ptr_next=NULL;
    /* deletion of dummy element */
    curr_point = lst;
    lst = lst->ptr_next;
    free (curr_point);

    return lst;
}

```

מקובל גם פתרון יותר יעיל ($O(n \log n)$) המבוסס על מיון מוקדם של המערך. במקרה זה הפונקציה שמממשת את אלגוריתם המיון צריכה להיות מוצגת במחברת וצריך למיין מערך עזר דינאמי כך שלא לדרוס את המערך המקורי.

שאלה 3 (28 נק')

כתוב פונקציה אשר מקבלת מחרוזת ובונה שני מערכים דינאמיים מהטיפוס int. המערך הראשון יהיה מורכב מקודי ASCII של אותיות של המחרוזת המקורית. המערך השני תכלול בתוכו מספרים חד-ספרתיים התואמים לתווים של המחרוזת אשר מציגים ספרות ('0' במחרוזת יופיע כ-0 במערך, '1' במחרוזת יופיע כ-1 במערך וכו'). על הפונקציה להעביר כתובות המערכים וגדלים שלהם ולהחזיר הכמות של תוים אחרים (לא אותיות ולא ספרות) במחרוזת המקורית.

דוגמא:

מחרוזת מקורית:

$aX+2Y=8$

מערך ראשון:

89 88 97

(89, 88, 97 הם קודי ASCII של 'X', 'a', ו-'Y' בהתאמה).

מערך שני:

8 2

הפונקציה תחזיר 2 בדוגמא.

במידה ואחד ממערכים החדשים לא מכיל אף איבר, כתובתו תהיה NULL.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```
int two_arrays (char *str, int **let_array, int **dig_array,
                int *let_size, int *dig_size)
{
    int let_count = 0, dig_count = 0, other_char_num, i;
    /* Estimation of array sizes */
    *let_size = *dig_size = 0;
    for (i=0; str[i]!='\0'; i++)
    {
        if (isalpha(str[i]))
            (*let_size)++;
        else
            if (isdigit(str[i]))
                (*dig_size)++;
    }
    other_char_num = i - *let_size - *dig_size;
    /* Allocation of arrays */
    if (*let_size)
        *let_array = malloc(*let_size*sizeof(int));
    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 other_char_num;
    /* Filling arrays */
    for (i=0; str[i]!='\0'; i++)
    {
        if (isalpha(str[i]))
        {
            (*let_array)[let_count] = str[i];
            let_count++;
        }
    }
}
```

```

        else
            if (isdigit(str[i]))
            {
                (*dig_array)[dig_count] = str[i] - '0';
                dig_count++;
            }
    } // for i
    return other_char_num;
}

```

שאלה 4 (16 נק')
נתונה התוכנית הבאה:

```

#include <stdio.h>

int fun (int);
int fun1 (int);
int fun2 (int);
int fun3 (int);

void main()
{
    int i;
    for(i=1; i<=3; i++)
        printf("%d \n", fun (i));
    printf("\n");
}

int fun (int x)
{
    static int i=0;
    int ( *pf[] )(int)={fun1, fun2, fun3},
        y = (*pf[i])(x);
    i++;
    return y;
}

int fun1 (int x)
{
    static int i=0;
    int a[]={10,20,30};
    i++;
    a[i] += x;
    return a[i];
}

int fun2 (int x)
{
    return x*x;
}

```

```
int fun3 (int x)
{
    return 3*x;
}
```

מהו פלט התוכנית?

21

4

9

האם יהיה תקין למחוק את המילה `static` בפונקציה `fun`? נמק אם לא, הראה איך ישתנה הפלט אם כן.

לא תקין עקב שגיאת ריצה: ניסיון לעדכן איבר שלא קיים במערך `a` של פונקציה `fun1`

האם יהיה תקין למחוק את המילה `static` גם בפונקציה `fun` וגם בפונקציה `fun1`? נמק אם לא, הראה איך ישתנה הפלט אם כן.

תקין.

פלט:

21

22

23

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

```
#include <stdio.h>

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

1