



מכון טכנולוגי חולון  
Holon Institute of Technology

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108

סמסטר ב' תשע"א

מועד ב'

17.07.2011

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

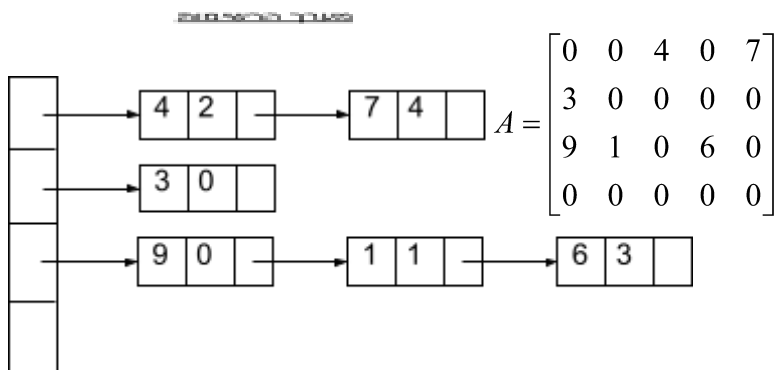
אין להשתמש בחומרי עזר, פרט, לדף מצורף לשאלון

שאלה 1 (28 נק')

מטריצה דלילה הינה מטריצה שמרבית איבריה בעלי ערך אפס.

כתוב פונקציה שמקבלת מטריצה דלילה  $A$  כמערך דו-ממדי סטטי. על הפונקציה לבנות מערך דינאמי של רשימות מקושרות אשר יישם מטריצה  $A$  בדרכ חסכונית באמצעות אחסון רק איברי המטריצה שאינם בעלי ערך אפס. רשימה מס'  $i$  של המערך תתאים לשורה מס'  $i$  של מטריצה  $A$ . איברי הרשימות יכללו בתוכם ערך של איבר המטריצה ומס' עמודה בה הוא נמצא במטריצה.

דוגמא:



הפונקציה תחזיר כתובת של האיבר הראשון במערך החדש.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

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

#define COLS 5 //example

element **sparse_matrix_to_lists (int a[][COLS], int rows, int cols)
{
    element **arr_ptr, *temp;
    int i, j;
    arr_ptr = calloc (rows, sizeof(element *));
    for (i=0; i<rows; i++)
    {
        arr_ptr[i] = NULL;
        for (j=cols-1; j>=0; j--)
            if (a[i][j] != 0)
            {
                temp = malloc(sizeof(element));
                temp->data = a[i][j];
                temp->col = j;
                temp->ptr_next = arr_ptr[i];
                arr_ptr[i] = temp;
            } //if
    } //for i
    return arr_ptr;
}

```

## שאלה 2 (28 נק')

כתוב פונקציה המקבלת מערך של מספרים שלמים.  
על הפונקציה לבנות מערך חדש דינאמי המורכב 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 המערך החדש הבא:

6	9
---	---

על הפונקציה להחזיר מצביע לאיבר הראשון של המערך החדש ולהעביר את הגודל שלו.  
בדוגמא הנ"ל הפונקציה תעביר 2.  
אם כל המספרים של המערך המקורי מופיעים יותר מפעם אחת אז הפונקציה תחזיר NULL ותעביר 0.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

```

int *alone_elements (int *a, int size, int *ptr_new_size)
{
    int i, j, found_equal, *new_arr, *mask;
    mask = calloc (size, sizeof(int));
    *ptr_new_size = 0;
    for (i=0; i<size; i++)
    {
        if (mask[i])
            continue; // for i
        found_equal = 0;
        for (j=0; j<i; j++)
            if (a[i] == a[j])
            {
                found_equal = mask[i] = mask[j] = 1;
                break; // for j
            }
        if (found_equal)
            continue; // for i
        for (j=i+1; j<size; j++)
            if (a[i] == a[j])
            {
                found_equal = 1;
                break; // for j
            }
        if (!found_equal)
            (*ptr_new_size)++;
    } // for i
    if (*ptr_new_size == 0)
    {
        free (mask);
        return NULL;
    }
    new_arr = calloc (*ptr_new_size, sizeof(int));
    j = 0;
    for (i=0; i<size; i++)
        if (!mask[i])
        {
            new_arr[j] = a[i];
            j++;
        }
    free (mask);
    return new_arr;
}

```

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 (void) אשר מקבלת מחרוזת המורכבת מסדרות של תווים שווים (לדוגמא, "\$\$\$\$\$\$^aaannnnFF"). ידוע כי מספר תווים בסדרה לא גדול מ-9. על הפונקציה להפוך כל סדרה במחרוזת לשני תווים כך שהתו הראשון יהיה ספרה המציגה את כמות התווים בסדרה והתו השני יהיה התו עצמו של הסדרה.

דוגמא:

\$\$\$\$\$\$^aaannnnFF  
3a4n2F1^7\$

מחרוזת לפני עיבוד:  
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;
    int i=0, j=0, count=0;

    temp = malloc (strlen(str) + 1);
    strcpy (temp, str);
    while (temp[i]!='\0')
    {
        for (count=1; temp[i+count] == temp[i+count-1]; count++);
        str[j] = '0' + count;
        str[j+1] = temp[i];
        i += count;
        j += 2;
    }
    str[j] = '\0';
    free (temp);
}
```

### שאלה 4 (16 נק')

נתונה התוכנית הבאה:

```
#include <stdio.h>

int sum_arr (int a[]);

void main()
{
    int i, a[3];
    char str1[10]="Hello", *str2;
    str1 = "Good bye"; //error: '=' : left operand must
                       //be l-value (str1 is a constant pointer)
    puts (str1);
    printf ("str2 = ");
    gets (str2); //running error: memory for string
                //wasn't allocated
```

```

    puts (str2);
    printf ("str1 = ");
    scanf ("%s", str1);
    puts (str1);
    str2 = str1;
    puts (str2);
    printf ("Enter array: ");
    for(i=0; i<3; i++)
        scanf("%d", *(a+i)); //running error: must be
                               //without *
    printf("Sum of the array members is %d\n",
        sum_arr(a));
}

/* הפונקציה מחשבת סכום איברי המערך */
int sum_arr (int a[])
{//sizeof returns size of pointer a but not size of array
    int i, sum=0, size=sizeof(a)/sizeof(int);
    for (i=0; i<size; i++)
        sum += a[i];
    return sum;
}

```

התוכנית שגויה!  
 'ש לתאר את כל שגיאות (קומפילציה, ריצה, לוגיות).

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

```

#include <stdio.h>

void main()
{
    printf ("%c", '&''&''&');
}

```

**&**