

מבוא למדעי המחשב
בחינת מועד ב', סמסטר א' תשס"ו, 24.2.2006

מרצה: גב' יעל כהן-סיגל.
 מתרגלת: גב' ליאת לוונטל.

משך המבחן: שעתיים וחצי.
 חומר עזר: מותר כל חומר עזר, מלבד מחשב.
 הנחיות:

1. יש לענות על כל השאלות.
2. קראו היטב כל שאלה. ודאו כי אתם מבינים את השאלה לפני שתתחילו לענות עליה.
3. כתבו בכתב יד ברור וקריא. תשובות לא קריאות לא תיבדקנה.
4. הערות לתשובותיכם ניתן לכתוב בעברית, גם בגוף פונקציות C.
5. ניתן ונדרש להגדיר פונקציות עזר לפי הצורך.
6. ניתן להשתמש בכל פונקציה המופיעה במצגות ההרצאות והתרגולים ע"י הצהרה עליה בלבד (אין צורך להגדירה). כמו כן, ניתן להשתמש בפונקציות מתוך הספריות stdio.h ו-stdlib.h. לא ניתן להשתמש בפונקציות אחרות בלא להגדיר אותן במפורש.

שאלה 1 (40 נק')

מערך דו-מימדי נקרא ריבועי אם הוא מכיל ריבוע, לפחות מסדר 2×2 , של מספרים זהים. מערך דו-מימדי נקרא n-ריבועי אם הוא ריבועי וגודל הריבוע המקסימלי שהוא מכיל הוא $n \times n$. לדוגמא:
 המערך הדו מימדי הבא הינו 3-ריבועי (כיון שהריבוע המקסימלי שהוא מכיל הינו מסדר 3×3 , כפי שמסומן בטבלה):

| | | | | |
|---|---|---|---|---|
| 3 | 0 | 1 | 4 | 3 |
| 2 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |
| 3 | 3 | 8 | 3 | 5 |
| 3 | 3 | 0 | 0 | 3 |

המערך הבא אינו ריבועי כלל:

| | | |
|---|---|---|
| 4 | 1 | 1 |
| 3 | 1 | 8 |
| 3 | 3 | 0 |

הגדירו פונקציה המקבלת מערך דו-מימדי של מספרים ואת מימדיו ומחזירה 0 אם המערך אינו ריבועי, אחרת תחזיר את גודל הריבוע המקסימלי שבו.



שאלה 2 (30 נק')

נתונה רשימה מקושרת דו-כיוונית אשר כל תא בה מוגדר באופן הבא:

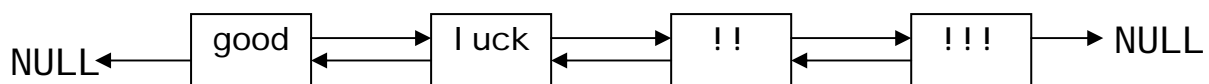
```
typedef struct cell{
    char *str;
    struct cell *prev;
    struct cell *next;
}Cell;
```

כאשר השדה str הינו מצביע למחרוזת.

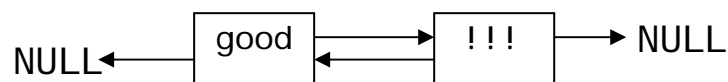
נגדיר חציון של רשימה כאיבר האמצעי ברשימה. אם מספר האיברים ברשימה זוגי אזי החציון שלה הינו שני האיברים האמצעיים. הגדירו פונקציה רקורסיבית המקבלת מצביעים לתחילת וסוף רשימה מקושרת דו-כיוונית כנ"ל ומוציאה את החציון שלה מן הרשימה.

לדוגמא:

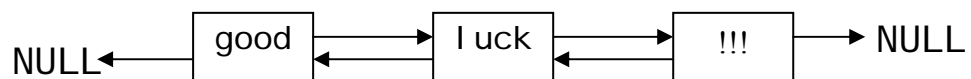
אם הפונקציה מקבלת את הרשימה (כלומר, מקבלת מצביע לתחילתה ולסופה):



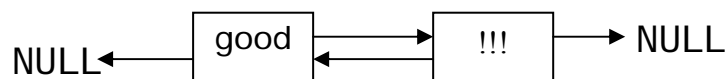
אזי לאחר ביצוע הפונקציה הרשימה תהיה:



אם הפונקציה מקבלת את הרשימה (כלומר, מקבלת מצביע לתחילתה ולסופה):



אזי לאחר ביצוע הפונקציה הרשימה תהיה:



שאלה 3 (30 נק')

נתון מערך של מספרים בגודל n ונניח כי $n > 1$. נסמן את איברי המערך כ- a_0, a_1, K, a_{n-1} . נרצה למצוא (אם קיים) $0 \leq i < n-1$ כך שמתקיים $a_0 + a_1 + K + a_i = a_{i+1} + K + a_{n-1}$. הגדירו פונקציה המקבלת מערך של מספרים ואת גודלו ומדפיסה למסך את כל האינדקסים עבורם מתקיים השוויון הנ"ל. ניתן להניח כי גודל המערך הינו לפחות 2 (אין צורך לבדוק זאת). אם לא קיים אינדקס עבורו מתקיים השוויון הנ"ל הפונקציה לא תדפיס דבר.

דרישות סיבוכיות:

סיבוכיות זמן: $O(n)$ סיבוכיות מקום: $O(n)$ (ניתן גם ב- $O(1)$)
פתרונות בסיבוכיות גבוהה יותר לא יתקבלו.

בהצלחה!



פתרון

Number 1:

```
#include <stdio.h>
#define N 5
#define M 4

int SquareMatrix(int mat[N][M], int n, int m);
int FindSquare(int mat[N][M], int n, int m, int i, int j);

int SquareMatrix(int mat[N][M], int n, int m)
{
    int i,j,max=0,sqr=0;
    for(i=0; i<n-1; i++)
    {
        for(j=0; j<m-1; j++)
        {
            sqr = FindSquare(mat,n,m,i,j);
            if(sqr > max)
                max = sqr;
        }
    }
    return max;
}

int FindSquare(int mat[N][M], int n, int m, int i, int j)
{
    int x,y;
    int size = 0, max_size = ((n-i) > (m-j)) ? (n-i) : (m-j);
    int val = mat[i][j];

    for(x=1; x<max_size; x++)
    {
        for(y=0; y<=x; y++)
        {
            if((mat[i+x][j+y] != val) || mat[i+y][j+x] != val)
                return size;
        }
        size = x+1;
    }
    return size;
}

/*
int main()
{
    int mat[5][4] = {{2,1,1,1},{3,1,1,6},{3,5,5,5},{4,5,5,5},{6,5,5,5}};
}
```

```

    printf("max square = %d\n", SquareMatrix(mat,5,4));
    return 0;
}*/

```

Number2:

```

void MedOut(Cell* first, Cell* last) /*we assume the list has at least 3 items*/
{
    if(first->next==last->prev || first->next->next==last->prev)
    {
        first->next = last;
        last->prev = first;
        return;
    }
    MedOut(first->next,last->prev);
    return;
}

```

Number3:

```

#include <stdio.h>
void FindIndex(int a[], int n)
{
    int i, sum=0, curr_sum=0;
    for(i=0; i<n; i++)
        sum += a[i];
    for(i=0; i<n; i++)
    {
        curr_sum += a[i];
        if(curr_sum*2 == sum)
            printf("%d ",i);
    }
    return;
}
/*
int main()
{
    int a[5] = {1,3,3,5,2};
    FindIndex(a,5);
    return;
}*/

```

