מבוא למדעי המחשב
בחינת מועד א', סמסטר א' תשס"ו, 3.2.2006

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 (30 נק')

הגדירו פונקציה המקבלת מערך של תווים ואת גודלו ומסדרת את אברי המערך כך שבתחילתו יופיעו כל האותיות, לאחר מכן יופיעו כל הספרות ובסופו יופיעו כל שאר התווים.
לדוגמא: עבור המערך

s	5	=	r	1	!	t	t	y	9	@	<	4

פלט אפשרי הינו המערך

s	r	t	t	y	5	1	9	4	=	!	@	<

שימו לב כי קיימים פלטים אפשריים נוספים (ואין זה משנה מי מהם תחזיר הפונקציה).

דרישות סיבוכיות:
סיבוכיות זמן: $O(n)$ סיבוכיות מקום: $O(1)$
פתרונות בסיבוכיות גבוהה יותר לא יתקבלו.

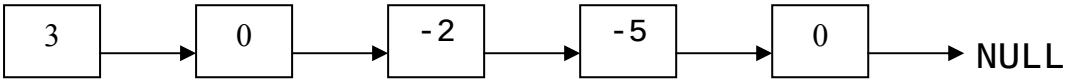
שאלה 2 (30 נק')

נתונה רשימה מקושרת אשר כל תא בה מוגדר באופן הבא:

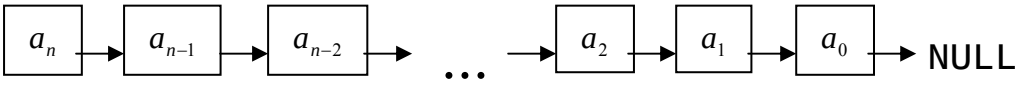
```
typedef struct cell{
    int num;
    struct cell *next;
} Cell;
```

באמצעות רשימה מקושרת כזו נוכל לייצג פולינום כאשר האיבר הראשון ברשימה מחזיק את המקדם של x בחזקה הגבוהה ביותר (שנסמנה n), האיבר השני מחזיק את המקדם של x בחזקת n-1 וכך הלאה. האיבר האחרון ברשימה מחזיק את המקדם של x בחזקת 0, כלומר את האיבר החופשי.

למשל, הפולינום $3x^4 - 2x^2 - 5x$ ייוצג ע"י הרשימה:



ובאופן כללי, הפולינום $a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + 5 \quad + a_2 x^2 + a_1 x + a_0$ ייוצג ע"י הרשימה:



הגדירו פונקציה רקורסיבית המקבלת מצביע לתחילת רשימה מקושרת כנ"ל ומספר x, ומחזירה את ערך הפולינום המיוצג ע"י הרשימה בנקודה x.

לדוגמא: עבור הפולינום $3x^4 - 2x^2 - 5x$ (המיוצג ע"י הרשימה לעיל) והמספר 2 תחזיר הפונקציה 30 (כיון ש $3 \cdot 2^4 - 2 \cdot 2^2 - 5 \cdot 2 = 30$).

רמז:

העזרו בעובדה כי

$$a_n x^n + a_{n-1}x^{n-1} + 5 \quad + a_1 x + a_0 = (((5 \quad ((((a_n x + a_{n-1})x + a_{n-2})x + a_{n-3})x + 5 \quad a_2)x + a_1)x + a_0$$

לדוגמא, עבור n=5 מתקיים

$$a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = ((((a_5 x + a_4)x + a_3)x + a_2)x + a_1)x + a_0$$

שאלה 3 (40 נק')

מטריצה מסדר $n \times n$ נקראת מושלמת אם היא מכילה אך ורק את המספרים 1,2,…,n כך שכל מספר מופיע בדיוק פעם אחת בכל שורה ובכל עמודה.

לדוגמא:

המטריצה $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 1 & 2 \end{pmatrix}$ היא מטריצה מושלמת אך המטריצות $\begin{pmatrix} 1 & 3 & 4 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}$ ו- $\begin{pmatrix} 1 & 3 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}$

אינן מושלמות.

הגדירו פונקציה המקבלת מטריצה (כלומר, מערך דו-מימדי של מספרים) ואת גודלה ומחזירה 1 אם המטריצה מושלמת ו- 0 אחרת.

דרישות סיבוכיות:

סיבוכיות זמן: $O(n^2)$ סיבוכיות מקום: $O(n)$

פתרונות בסיבוכיות גבוהה יותר לא יקבלו את כל הנקודות.

בהצלחה!

מבוא למדעי המחשב
בחינת מועד א', סמסטר א' תשס"ו, 3.2.2006

Solution

Number1:

```
#include <stdio.h>

void Sort(char a[], int n);
int IsLetter(char c);
int IsDigit(char c);
void swap(char* x, char* y);
void Print(char a[],int n);

enum {FALSE,TRUE};

void Sort(char a[], int n)
{
	int i=0, /*point to the start of the array*/
		j=n-1; /*point t the end of the array*/
	while(i<j)	/*arrange the letters*/
	{
		while(IsLetter(a[i])) i++;
		while(!IsLetter(a[j])) j--;
		if (i<j)
		{
			swap(&a[i],&a[j]);
			i++;
			j--;
		}
	}
	j = n-1;
	while(i<j)	/*arrange the digits and the other characters*/
	{
		while(IsDigit(a[i])) i++;
		while(!IsDigit(a[j])) j--;
		if(i<j)
		{
			swap(&a[i],&a[j]);
			i++;
			j--;
		}
	}
	return;
}

int IsLetter(char c)
{
	if((c>='a' && c<='z') || (c>='A' && c<='Z'))
```

```
                return TRUE;
        return FALSE;
}

int IsDigit(char c)
{
        if(c>='0' && c<='9')
                return TRUE;
        return FALSE;
}

void swap(char* x, char* y)
{
        char temp = *x;
        *x = *y;
        *y = temp;
        return;
}

void Print(char a[],int n)
{
        int i;
        for (i=0;i<n;i++)
                printf("%c ",a[i]);
        printf("\n");
        return;
}

int main()
{
        char a[10] = {'a','1','*','0','d','s','=','+','3','4'};
        printf("before: ");
        Print(a,10);
        Sort(a,10);
        printf("After: ");
        Print(a,10);
        return 0;
}
```

Number 2:

```
int Poly(Cell* first, int x)
{
	static res = 0;
	if(first == NULL)
		return 0;
	res = res*x + first->num;
	Poly(first->next,x);
	return res;
}
```

Number3:

```
#include <stdio.h>

#define N 4
enum {FALSE,TRUE};

int IsPerfect(int mat[][N], int n)
{
	int i,j,row_arr[N+1],col_arr[N+1];
	for(i=0; i<n; i++)		/*check if the values are between 1-n*/
		for(j=0;j<n;j++)
			if((mat[i][j]<1) || (mat[i][j]>N))
				return FALSE;
	for(i=0;i<n;i++)
	{
		for(j=0;j<=n;j++)	/*initialize the arrays*/
			row_arr[j] = col_arr[j] = 0;
		for(j=0;j<n;j++)
		{
			row_arr[mat[i][j]]++;
			col_arr[mat[j][i]]++;
		}
		for(j=1;j<=n;j++)	/*check if the i'th col and the i'th row are perfect*/
			if((row_arr[j]==0) || (col_arr[j]==0))
				return FALSE;
	}
	return TRUE;
}

int main()
{
	int mat1[N][N]={{1,2,3,4},{2,3,4,1},{3,4,1,2},{4,1,2,3}};
	int mat2[N][N]={{1,2,3,4},{2,3,4,1},{3,4,1,2},{4,1,2,3}};

	if(IsPerfect(mat1,N)) printf("mat1 is perfect\n");
	if(IsPerfect(mat2,N)) printf("mat2 is perfect\n");
```

```
        return 0;
}
```