



מבוא למדעי המחשב מ' 234114

מבוא למדעי המחשב ח' 234117

מבחן מועד א', סמסטר חורף תשס"ב (תאריך 29.1.03)

מס' סטודנט

--	--	--	--	--	--	--	--	--	--

שם פרטי

--

שם משפחה

--

שאלה	ערך	הישג
1	15	
2	15	
3	16	
4	18	
5	16	
6	20	
סכום	100	

חומר עזר: אין להשתמש בחומר עזר, מלבד דף העזר שחולק.

הוראות לנבחנים ולנבחנות:

1. מלאו את הפרטים בראש דף זה (בעט).
2. בדקו שיש 13 עמודים (6 שאלות) כולל עמוד זה.
3. התשובות ייכתבו על טופס המבחן.
4. כתבו בכתב-יד נקי וברור (מומלץ להשתמש בעפרון ומחק).
5. אין לכתוב הערות והסברים לתשובות.
6. בכל השאלות ניתן להניח שהקלט תקין.

חורף 234114 :

מרצים: פרופ' ח' רוני קימל, ד"ר יחיאל קמחי.

מתרגלים: רועי מלמד, רוני רב, יונתן לזרסון, יונתן נאור, שגיא שיין, עמיר אדלר, מוחמד מחמיד.

חורף 234117 :

מרצים: רוני למפל, גבי זודיק.

מתרגלים: רועי מלמד, רוני רב, עומר יחזקאלי, רון בגליטר, אורית כהן, נלה גורביץ'.

שאלה 1 (15 נקודות)

חלק א' (8 נקודות)

נתונים המערכים/משתנים הבאים בזיכרון המחשב:

```
int a[] = { 6,5,4,3,2,1};
char *b[] = {"Today","is","a","lovely","day","for","writing","A","test"};
char * c = "hAppY";
char* d[] = {b[a[2]], b[a[0] - a[5]]};
int e[2][3] = { {0,3,4}, {1,2,7} };
int f[] = { b[3][3] - b[4][0] , a[2]+a[4],e[1][2] - e[0][1]};
char* g = b[3];
```

כתבו את ערך הביטויים הבאים :

c[1]	_____	(1 נק')
f[2]+*(f+1)	_____	(1 נק')
*(f+a[5])	_____	(1 נק')
**(d+1)	_____	(1 נק')
(g + 3)[2]	_____	(2 נק')
((b +a[2])+e[0][0])	_____	(2 נק')

חלק ב' (7 נקודות)

חלק זה מתבסס על המערכים/משתנים מסעיף א'.

```
struct test {
    int x;
    double y;
    char * s;
    int a[3];
};
```

נתון קטע הקוד הבא:

```
typedef struct test Test;
Test h, *k;
k=&h;
h.x=1;
h.y=4.7;
h.s=b[4];
h.a[0]=a[2];
h.a[1]=a[1];
h.a[2]=a[0];
```

כתבו את ערך הביטויים הבאים :

h.s[1]	_____	(1 נק')
k->s[2]	_____	(1 נק')
h.x+k->y	_____	(1 נק')
*(f+h.x)	_____	(2 נק')
*(h.s+k->x)	_____	(2 נק')

שאלה 2 (15 נקודות)

בהינתן המערך **arr** המכיל **n** מספרים טבעיים בטווח קבוע וידוע מראש $[1, k]$ (הטווח כולל את 1 ואת k), נרצה לענות **בזמן $O(1)$** על השאלתה הבאה:
 "עבור **a, b** טבעיים (שונים מ-0, וכן $a \leq b$), כמה מאברי המערך **arr** הם בטווח $[a, b]$?"

לדוגמא: בהינתן המערך **arr** הבא המכיל $n=10$ מספרים מהטווח $[1, 20]$ ($k=20$):

5	1	20	9	7	7	5	4	1	1
---	---	----	---	---	---	---	---	---	---

אלה יהיו התשובות על השאלות הבאות, עבור a ו-b הנ"ל:
 עבור $a=1, b=2$ נענה **3** (המספרים שבטווח הם $arr[1]=1, arr[8]=1, arr[9]=1$),
 עבור $a=7, b=7$ נענה **2** (המספרים שבטווח הם $arr[4]=7, arr[5]=7$),
 עבור $a=8, b=209$ נענה **2** וכו'..

בסעיף א' של השאלה נבצע שלב מקדים שיאפשר את ביצוע השאלתה (סעיף ב' של השאלה) **בזמן $O(1)$** .

עמוד 4

א. (8 נקודות) על הפונקציה `init_help_array` לאתחל את מערך העזר, `help_arr`, על מנת שכל שאילתא בסעיף ב' תתבצע בזמן $O(1)$.

הפונקציה תקבל:

- `arr` - מערך המכיל n המספרים בטווח $[k, 1]$.
- `n` - גודל `arr`.
- `help_arr` - מערך העזר בגודל k .
- `k` - מגדיר את גודל הטווח של אברי `arr`.

דרישות: סיבוכיות זמן $O(n+k)$, סיבוכיות מקום $O(1)$.

```
void init_help_array( unsigned int arr[],  
                     unsigned int n,  
                     unsigned int help_arr[],  
                     unsigned int k)  
{
```

```
}
```

ב. (7 נקודות) כעת נכתוב את פונקצית השאילתה אשר בהינתן a, b תחזיר בזמן $O(1)$ כמה מאברי המערך `arr` הם בטווח $[a, b]$ (תחום סגור הכולל את a, b).

הפונקציה תקבל:

- `help_arr` – מערך העזר, בגודל k , שבנינו בסעיף הקודם.
- `k` - גודל מערך העזר.
- `a` - גבול תחתון של הטווח.
- `b` - גבול עליון של הטווח.

דרישות: סיבוכיות זמן ומקום נוסף $O(1)$.

```
int count_numbers_in_range_O_1(unsigned int help_arr[ ],
                                unsigned int k,
                                unsigned int a,
                                unsigned int b)
{

}

}
```

שאלה 3 (16 נקודות)

נתונים כקלט 2 מערכים, האחד $a[]$ באורך m והשני $b[]$ באורך n כך ש- $m < n$.
א. (6 נקודות) הצע שיטה יעילה לבדוק האם המערך הקצר $a[]$ מופיע כתת מערך של המערך הארוך $b[]$.

דוגמא 1: עבור הקלט:

$a[] = \{3, 5, 2\}$
 $b[] = \{1, 7, 3, 5, 2, 2, 1, 0\}$

Match Found

התוכנית תדפיס:

דוגמא 2: עבור הקלט:

$a[] = \{5, 1, 2\}$
 $b[] = \{1, 7, 3, 5, 2, 2, 1, 0\}$

Did not find a match

התוכנית תדפיס:

```
void find_pattern (int a[ ], unsigned int m, int b[ ], unsigned int n)
{
```

```
}
```

עמוד 6

ב. (2 נקודות) מהי סיבוכיות התכנית שרשמת? נמקו בקצרה.

ג. (8 נקודות) כעת ידוע בנוסף כי:

- המערך b ממין מהקטן לגדול וכל ערך מופיע בו לכל היותר פעם אחת.
- המערך a ממין מהקטן לגדול וכל ערך מופיע בו לכל היותר פעם אחת.

האם ניתן לשפר את סיבוכיות הפונקציה `find_pattern()`?
אם כן – כתבו גירסה יעילה יותר של הפונקציה בעמוד הבא, ורשמו את סיבוכיותה.
אם לא – הסבירו מדוע.

הערה: ניתן להעזר בפונקציית העזר הבאה, `search()`, המחפשת את האיבר c במערך a ממין a , כאשר אזור החיפוש מתחיל באינדקס `start` ומסתיים ב-`end` (כולל). הפונקציה מחזירה את האינדקס של מיקום האיבר c במערך, או -1 אם האיבר לא קיים במערך.

```
int search (int a[], int start, int end, int c)
{
    int med;

    if (start==end) {
        if (a[start]==c) return start;
        else return -1;
    }

    med = (start+end)/2;
    if (a[med]<c)
        return search (a, med+1, end, c);
    else
        return search (a, start, med, c);
}
```

```
void find_pattern (int a[ ], unsigned int m, int b[ ], unsigned int n)
{

}

}
```

מה סיבוכיות גרסה זו של הפונקציה? נמקו בקצרה.

שאלה 4 (18 נקודות)

נתונה הפונקציה הבאה:

```
void f(double * p1, double * p2, int len, int jump)
{
    double temp;
    int j, offset;

    for ( offset = j = 0 ; j < len ; j++, offset += jump )
    {
        temp = p1[ offset ];
        p1[ offset ] = p2[ offset ];
        p2[ offset ] = temp;
    }
}
```

הסעיפים הבאים מתייחסים להגדרות הבאות:

```
#define NROWS (10)
#define NCOLS (20)
double matrix[NROWS][NCOLS];
double square[NROWS][NROWS];
```

א. (4 נק') קראו לפונקציה $f()$ כך שתחליף את הערכים שבשורות ה- k וה- m של $matrix$.

ב. (4 נק') קראו לפונקציה $f()$ כך שתחליף את הערכים שבעמודות ה- r וה- s של $matrix$.

ג. (4 נק') קראו לפונקציה $sum()$ שלהלן כך שתסכם את אלכסון המטריצה $square$ (כלומר את האיברים $square[t][t]$ עבור $t = 0, \dots, NROWS-1$).

```
double sum(double * p, int len, int jump)
{
    double s = *p;

    while ( --len )
    {
        p += jump;
        s += *p;
    }
    return s;
}
```


עמוד 9

ד. (6 נק') השלימו את הפונקציה הבאה, האמורה לקרוא מקובץ סדרת מספרים ממשיים, ולהחזיר את מספר המספרים הגבוהים ממוצע הסדרה. המספר הראשון בקובץ הינו שלם אי שלילי המהווה את אורך סדרת המספרים הממשיים שלאחריו. הפונקציה מקבלת את שם קובץ הקלט כפרמטר.

דוגמה לקובץ קלט חוקי: 1.0 10.3 4.1 1.2 3.4 5
ממוצע חמשת המספרים הינו 4, ולכן על הפונקציה להחזיר את הערך 2.

הנחיות:

יש לקרוא את קובץ הקלט פעם אחת בלבד.
יש להחזיר את הקבוע FILE_PROBLEMS אם לא ניתן לפתוח את קובץ הקלט.
יש להחזיר את הקבוע MEMORY_PROBLEMS עבור בעיה של מחסור בזכרון.

```
#include <stdio.h>
#include <stdlib.h>

#define FILE_PROBLEMS    (-1)
#define MEMORY_PROBLEMS (-2)

int more_than_average (char * file_name)
{
    FILE * f;
```

```
}
```

שאלה 5 (16 נקודות)

בשאלה זו עליכם להשלים פונקציה המדפיסה הוראות לפתרון בעיית מגדלי הנוי, תחת אילוץ נוסף: בכל צעד, מותר להזיז חישוק רק בין שני מגדלים סמוכים. לפיכך, לא ניתן בצעד בודד להעביר חישוק ממגדל A למגדל C (או להיפך) – ממגדלים C,A ניתן להעביר חישוקים למגדל B בלבד.

קראו לפונקציה print_move() שלהלן על מנת להדפיס פקודה להזזת חישוק.

```
typedef enum {A,B,C} tower_t;
```

```
void print_move (int disc_num, tower_t from, tower_t to)
{
    printf("Move disc %d from tower %c to tower %c\n",
           disc_num , 'A'+from, 'A'+to
    );
}
```

```
void hanoi (int n, tower_t from, tower_t to)
{
    tower_t via = (A+B+C) – from – to;
    if ( n == 0 ) return;

    if ( via != B )
    {
        _____
        _____
        _____
    }
    else /* via == B */
    {
        _____
        _____
        _____
        _____
        _____
    }
    return;
}
```

שאלה 6 (20 נקודות)

שאלה זו (על כל סעיפיה) מתייחסת להגדרות הבאות:

```
#define NROWS (10)
#define NCOLS (50)
```

```
typedef enum { ADD, SUBTRACT } plusminus_t;
```

חלק א (3 נקודות)

השלימו את הפונקציה הבאה, המקבלת מערך חד מימדי `arr` מטיפוס `unsigned int` ואת אורכו `len`, ומחזירה את הערך הנמוך ביותר במערך.

```
unsigned int min_value(unsigned int arr[], unsigned int len)
{
```

```
}
```

חלק ב (3 נקודות)

השלימו את הפונקציה הבאה, המקבלת ארבעה פרמטרים:

- מערך חד מימדי `arr` מטיפוס `unsigned int`.
- מערך חד מימדי `sum` מטיפוס `int`.
- אורך `len` (שני המערכים `arr` ו-`sum` הינם מאורך זה).
- ערך `operation` מטיפוס `plusminus_t`.

אם `operation` הינו `ADD`, על הפונקציה לחבר למערך `sum` את המערך `arr` (חיבור איבר-איבר).
אם `operation` הינו `SUBTRACT`, על הפונקציה להחסיר מהמערך `sum` את המערך `arr` (חיסור איבר-איבר).
בשני המקרים, הכוונה היא לעדכן כל כניסה במערך `sum` בהתאם לערך הכניסה המקבילה במערך `arr` ולערך הפרמטר `operation`.

```
void update_sum ( unsigned int arr[],
                  int          sum[],
                  unsigned int len,
                  plusminus_t op )
```

```
{
```

```
}
```

חלק ג (14 נקודות)

בסעיף זה עליכם יהיה להשלים את הפונקציה `covering_rows`, המקבלת:
 1. מערך דו-מימדי מטיפוס `unsigned int` שבו `NROWS` שורות ו- `NCOLS` עמודות.

2. מספר שלם `setsize`
 ומדפיסה את כל הקבוצות של `setsize` שורות בדיוק כך שמתקיים התנאי הבא:
 סכום `setsize` השורות (איבר-איבר, כחיבור וקטורים) אינו מכיל אפס.

דוגמה: עבור המטריצה הבאה ו- `setsize=3`

השורה ה- 0
 השורה ה- 1
 השורה ה- 2
 השורה ה- 3

1	3	0	1	0	1	0
1	8	1	0	0	0	1
0	7	0	1	1	0	1
0	5	0	0	0	1	1

על הפלט להיות שתי השורות הבאות (סדר השורות אינו משנה):

0,1,2

1,2,3

2	18	1	2	1	1	2
---	----	---	---	---	---	---

כי הסכום (איבר-איבר) של שלוש השורות הראשונות הוא

1	20	1	1	1	1	3
---	----	---	---	---	---	---

סכומן (איבר-איבר) של שלוש השורות האחרונות הוא

וסכום של כל קבוצה אחרת של שלוש שורות מכיל אפס בלפחות אחת מהעמודות.

הדרכה:

- בפונקציה `covering_rows` הגדירו (ואתחלו) שטחי עזר.
- הפונקציה `covering_rows` נעזרת בפונקציה `covering_help` – השלימו גם אותה.
- העזרו בפונקציות שכתבתם בסעיפים הקודמים של שאלה זו, ובפונקצית ההדפסה הבאה:

```
void print_positive ( unsigned int arr[], int len)
{
    int k;

    for ( k = 0; k < len; k++)
        if ( arr[k] ) printf ("%d ", k) ;
    printf("\n");
}
```

```
void covering_rows ( unsigned int matrix[][NCOLS], int setsize )
{
    unsigned int help_mem1[_____] = {0};

    int help_mem2[_____] = {0};

    covering_help ( matrix, setsize, _____, _____, _____ );
    return;
}
```

עמוד 13

```

void covering_help ( unsigned int      matrix[][NCOLS],
                    int      setsize,
                    unsigned int * subset,
                    int      * sum,
                    _____ )
{
    /*******/

    if ( !setsize )
    {
        if ( _____ )

            print_positive(_____);

        return;
    }

    if ( setsize + _____ > _____ ) return;

    /*******/

    subset [ _____ ] = _____;

    update_sum ( _____, _____, _____, _____ );

    covering_help (matrix, _____, _____, _____, _____ );

    /*******/

    _____;

    _____;

    _____;

    return;
}

```

