



מבוא למדעי המחשב מ' 234114
מבוא למדעי המחשב ח' 234117

מבחן מועד א', סמסטר חורף תשס"ד 04.2.04

שם משפחה	שם פרטי	מס' סטודנט										
<input type="text"/>	<input type="text"/>	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>										

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

משך המבחן: 3 שעות.
חומר עזר: אין להשתמש בחומר עזר.

הוראות לנבחנים ולנבחנות:

- 1 מלאו את הפרטים בראש דף זה (בעט).
- 2 בדקו שיש 13 עמודים (6 שאלות) כולל עמוד זה ונספח.
- 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: ד"ר יחיאל קמחי, צחי קרני
מתרגלים: שאדי סאבא, אורית עדן, ניר זפקוביץ', עמיר אדלר, רן רובנשטיין, רוסטם טייגר.

חורף 234117
מרצים: ד"ר יחיאל קמחי, יואב ציבין, רועי מלמד.
מתרגלים: שאדי סאבא, אבישי טל, בני גודלין, חיה זלברשטיין, עזרא אוחיון, עידו פלדמן, נלה גורביץ'

שאלה 1 (15 נקודות)
סעיף א.

```
#define N 10
void strange(int numbers[])
{
    int i, j, temp;
    for (i = 1; i < N; i++)
        for (j = 0; j < i; j++) {
            if (numbers[j+1] < numbers[j]) {
                temp = numbers[j+1];
                numbers[j+1] = numbers[j];
                numbers[j] = temp;
            }
        }
}
```

1. איך יראה המערך numbers הבא אחרי הרצת הפונקציה strange() עליו?
לפני:

3	0	4	5	6	7	8	9	2	1
---	---	---	---	---	---	---	---	---	---

אחרי:

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

1. מהי סיבוכיות האלגוריתם (זמן ומקום נוסף)? _____

סעיף ב

נתונים המערכים/משתנים הבאים בזיכרון המחשב:

char * a	=	"LiFe";
int b	=	a[4];
char*c	=	a+4;
int d[]	=	{ -1,-2,-3,-4};
char *e[]	=	{"Everything","Is","goNNa","Be","Allright"};
char g[2][3]	=	{ {'H','i','\0'}, {'F','i','\0'} };

כתבו את ערך הביטויים הבאים :

*(c+d[2])	
*c +d[2]	
*c+b	
**e	

כתבו מה ידפיסו הפקודות הבאות :

printf("%s",*(e+d[1]-d[2]));	
printf("%s",&(g[0][0])-d[-d[b+1]]);	

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly aged or off-white appearance. The edges of the paper are visible, showing it's a full sheet. There is no handwriting or other markings on the paper.

שאלה 3 (15 נקודות)

נתונה מטריצה ריבועית mat בגודל $N \times N$ (מוגדר ב `#define`). נגדיר כי הריבוע מסדר גודל 1 במטריצה הוא $mat[0][0]$ בלבד.

הריבוע מסדר גודל 2 מכיל את $mat[0][0], mat[0][1], mat[1][0], mat[1][1]$.

לכל i , הריבוע מסדר גודל i מכיל i^2 איברים מהמטריצה כך שארבעת קודקודי הריבוע הם:

$mat[0][0], mat[0][i-1], mat[i-1][0], mat[i-1][i-1]$.

נגדיר מטריצה ממוינת-ריבועים כמטריצה בה כל האיברים שנמצאים בריבוע מסדר גודל i ואינם נמצאים בריבוע מסדר גודל $i-1$ גדולים ממש מהאיברים בריבוע מסדר גודל $i-1$, לכל $1 \leq i \leq N-1$.

לדוגמא, $mat1$ הינה מטריצה ממוינת-ריבועים ואילו $mat2$ אינה מטריצה ממוינת-ריבועים. עליכם לממש את הפונקציה `exists` המקבלת מטריצה ממוינת-ריבועים בגודל $N \times N$, ומספר שלם כלשהו x . על הפונקציה להחזיר `TRUE` אם x נמצא במטריצה. אחרת, על הפונקציה להחזיר `FALSE`.

אנו ממליצים על פתרון מודולארי (חלוקה לפונקציות) כפי שנלמד בכיתה. ניתן לכתוב את הפונקציות הנוספות אחרי מימוש הפונקציה `exists`.

דרישות סיבוכיות: זמן: $O(N)$, מקום: $O(1)$. פתרונות נכונים אשר לא יעמדו בדרישות הסיבוכיות לא יתקבלו.

-2	2	5
-1	2	6
4	3	7

mat1

-2	2	7
-1	2	6
4	3	2

mat2

```
#define N ... /*N is defined to some integer*/
typedef enum { FALSE, TRUE } boolean;
boolean exists (int mat[N][N], int x)
```

שאלה 4 (15 נקודות)

נתונים שני מערכים של שלמים:

```
int arr1[N+1], arr2[N];
```

כאשר N קבוע כלשהו נתון.

המערך `arr1` מכיל $N + 1$ ערכים שונים והמערך `arr2` מכיל את כולם, פרט לאחד מהם. אתם צריכים לנסות להגיע לפתרון עם סיבוכיות המקום והזמן הטובה ביותר שתוכלו להשיג, פתרון **נכון** עם סיבוכיות לא אופטימאלית יזכה בניקוד חלקי.

סעיף א.

כתבו פונקציה המקבלת את שני המערכים ואת גודלם

ומוצאת (מחזירה) את האיבר החסר ב-arr2. מהי סיבוכיות הפתרון (זמן ומקום)

בסעיף זה מותר להשתמש באלגוריתמים (פונקציות) שנלמדו בכיתה, אולם יש לציין במדויק את חתימת כל פונקציה, ולכלול אותה בניתוח הסיבוכיות.

חתימת הפונקציות אשר נכללו ולא מומשו	סיבוכיות זמן	סיבוכיות מקום

```
int findMissing(int arr1[],int arr2[],int len_arr2)
{
```

This image shows a full page of blank white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

סיבוכיות זמן:

סיבוכיות מקום נוסף:

סעיף ב.

פתרו את חלק א' כאשר נתון ששני המערכים ממוינים

```
int findMissingInSorted(int arr1[],int arr2[],int len_arr2)
{
```

```
}
```

סיבוכיות זמן:

סיבוכיות מקום נוסף:

שאלה 5 (20 נקודות)

נגדיר מבנה כדלקמן המתאר נקודה במישור הקואורדינטות

```
struct point {
    double x, y;
};

typedef struct point Point;
```

הגדרה מערך מספרים דו-ממדי $a[M][N]$ נקרא ממין בעמודות אם לכל $0 \leq i < M-1$ ולכל $0 \leq j < N$ מתקיים $a[i][j] \leq a[i+1][j]$ (דהיינו כל האיברים באותה העמודה מסודרים בסדר לא יורד).

הגדרה מערך מספרים דו-ממדי $a[M][N]$ נקרא ממין בשורות אם לכל $0 \leq i < M$ ולכל $0 \leq j < N-1$ מתקיים $a[i][j] \leq a[i][j+1]$ (דהיינו כל האיברים באותה השורה מסודרים בסדר לא יורד).

נתון מערך דו-ממדי $a[M][N]$ Point כאשר הקבועים M ו- N הם חזקות של 2 ומוגדרים באמצעות `#define`. מטרת השאלה היא לכתוב פונקציה בשפת C אשר ממיינת את המערך כך שקואורדינטות x של איבריו יהיו ממוינות בשורות וקואורדינטות y של איבריו יהיו ממוינות בעמודות.

דוגמא המערך $a[3][3]$ הבא עונה על דרישות השאלה:

$a[0][0] = (1,1)$	$a[0][1] = (2,5)$	$a[0][2] = (9,1)$
$a[1][0] = (1,2)$	$a[1][1] = (4,6)$	$a[1][2] = (4,2)$
$a[2][0] = (1,7)$	$a[2][1] = (3,9)$	$a[2][2] = (8,7)$

(כאשר סדר הקואורדינטות בדוגמא הוא (x,y)).

סעיף א

השלם את המקומות החסרים בפונקציה `merge_by_x` אשר ממזגת שני מערכים חד-ממדים מטיפוס Point הממוינים בסדר לא יורד לפי ערך של קואורדינטה x .

```
void merge_by_x (Point a[], int na, Point b[], int nb,
Point c[])
{
    int ia, ib, ic;
    for(ia = ib = ic = 0; (ia < na)&&(ib < nb); ic++)
    {
        if( _____ ) {
            c[ic] = a[ia];
            ia++;
        }
        else {
            _____;
            ib++;
        }
    }
    for(; ia < na; ia++, ic++) _____;
    for( _____ ) _____;
}
```

סעיף ב

השלם את המקומות החסרים בפונקציות merge_sort_by_x (ופונקצית עזר רקורסיבית internal_msort_by_x) אשר ממיינת מערך חד-ממדי מטיפוס Point בסדר לא יורד לפי ערך של קואורדינטה x.

```
void merge_sort_by_x (Point a[], int n)
```

```
{
    _____ tmp_array;
    tmp_array = (_____)malloc(____);
    internal_msort_by_x(a, n, tmp_array);
    free(tmp_array);
}
```

```
void internal_msort_by_x (_____ a[], _____ n, _____ tmp_array[])
```

```
{
    int left, right;
    if (n < 2) return;
    left = n / 2;
    right = n - left;
    internal_msort_by_x (_____, _____, _____);
    internal_msort_by_x (_____, _____, _____);
    merge_by_x (_____, _____, _____, _____, _____);
    memcpy(a, tmp_array, _____);
}
```

סעיף ג

השלם את הפונקציה `sort_by_x_and_y` אשר ממיינת את המערך כך שקואורדינטות x של איבריו תהיינה ממוינות בשורות וקואורדינטות y של איבריו תהיינה ממוינות בעמודות. הפונקציה תעבוד בסיבוכיות זמן $O(MN(\log M + \log N))$.

הפונקציה יכולה להיעזר (אך לא חייבת להשתמש בכולם) ב:

- פונקציות `malloc` ו-`memcpy`.
- פונקציות מסעיפים א'-ב' הממזגות וממיינות מערך חד-ממדי לפי ערכים של קואורדינטות x .
- פונקציות אשר דומות לפונקציות מסעיפים א'-ב', אך ממזגות וממיינות מערך חד-ממדי לפי ערכים של קואורדינטות y . החתימות של הפונקציות הן:

```
void merge_by_y (Point a[], int na, Point b[], int nb, Point c[])
void merge_sort_by_y (Point a[], int n)
```

הערה

שימו לב שבמערך דו ממדי כלשהו - לדוגמה `a[10][10]`, ניתן לשלוח את השורה `a[3]` (או כל שורה אחרת) לפונקציה שמצפה לקבל מערך חד ממדי עם הגודל המתאים של המערך למשל `f(a[3],10)`.

פתרון:

```
void sort_by_x_and_y (Point a[M][N])
{
```

```
    merge_sort_by_y( &a[0][0], N*M );
```

```
}
```

הסבר את הפתרון שלך. מדוע הפונקציה שכתבת עומדת בדרישות הסיבוכיות?

סעיף א:

עבור הקלט: $x=12$, $\text{int } a[] = \{5,12,1,4,7\}$ (בהתאם לערך k):

הפיתרון צריך להשתמש בשיטת ה- `backtracking`.

[illegible]

סעיף ב:

השלם את הפונקציה f2 כך שאם קיימים j ($j \leq k$) איברים **שונים** ב-a שסכומם הוא x, הפונקציה תדפיס את j האינדקסים הללו. אחרת לא יודפס דבר. הפונקציה אמורה להדפיס את **כל** הפתרונות החוקיים לבעיה (כלומר את כל הקבוצות בעלות j ($j \leq k$) האיברים השונים שסכומם הוא x). במידה ו f2 דומה ל f1 ניתן לתאר רק את השינויים הדרושים בקוד שכתבת עבור סעיף א כך שהקוד יממש את f2 [תציין ותשכתבו מחדש את השורות ששונות], או לכתוב את הפונקציה במלואה.

```
int f2 (unsigned int a[],int b[],int lena,int lenb,unsigned int x,unsigned int k)
{
```

```
}
```