

מבוא למדעי מחשב מ' / ח' (234114 / 234117) סמסטר חורף תשע"ד

מבחן מסכם מועד ב', 27 מרץ 2014

עח פרנוי	שח משפחה			127	וודו	יר ח	ากท

משך המבחן: 3 שעות.

חומר עזר: אין להשתמש בכל חומר עזר (מודפס או ממוחשב).

הנחיות כלליות:

- מלאו את הפרטים בראש דף זה ובדף השער המצורף, בעט בלבד.
 - . בדקו שיש 26 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- כתבו את התשובות על טופס המבחן בלבד, במקומות המיועדים לכך. שימו לב שהמקום המיועד לתשובה אינו מעיד בהכרח על אורך התשובה הנכונה.
- העמודים הזוגיים בבחינה ריקים. ניתן להשתמש בהם כדפי טיוטה וכן לכתיבת תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תיבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. <u>ניתן בהחלט להשתמש בעיפרון ומחק,</u> פרט לדף השער אותו יש למלא בעט.
- בכל השאלות, הנכם רשאים להגדיר ולממש פונקציות עזר כרצונכם. לנוחיותכם, אין חשיבות לסדר מימוש הפונקציות בשאלה, ובפרט ניתן לממש פונקציה לאחר השימוש בה.

הנחיות תכנות כלליות, אלא אם מצוין אחרת בשאלה:

- אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה, למעט פונקציות קלט/פלט והקצאת זיכרון (malloc)
 - אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
 - ניתן להשתמש בהקצאות זיכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זיכרון.
 - stdbool.h-ביתן להשתמש בטיפוס bool haura euro

צוות הקורס 234114/7

מרצים: גב' אנסטסיה דוברובינה, ד"ר רן רובינשטיין, פרופ' רון קימל (מרצה אחראי).

מתרגלים: נמרוד סבן פרטוש, אריק יודין, נדיה לבאי, אביב סגל, בת-חן גולדן, רן ברנשטיין, דור הריס, ברק פת, נחשון כהן (מתרגל אחראי).

בהצלחה!



_



<u>שאלה 1 (25 נקודות)</u>

<u>סעיף א</u>

נתונות הפונקציות func1(), func2(), func3() להלן. כתבו מתחת לכל פונקציה מה היא מדפיסה:

```
int f(int n)
{
    int x=0;
    while (x*x < n) {
        x++;
    }
    return x*x;
}

void funcl()
{
    printf("%d\n", f(8));
    printf("%d\n", f(f(f(8))));
    printf("%d\n", f(-8));
}</pre>
```

:func1() פלט של





```
void g(int* p, int* q)
{
    *(p + *q) = 2*(*q);
    q++;
}

void func2()
{
    int a[6];
    for (int i=0; i<6; i++) {
        g(a,&i);
    }
    for (int i=0; i<6; i++) {
        printf("%d\n", a[i]);
    }
}</pre>
```

:func2() פלט של







```
void swap(char* p, char* q)
    char tmp = *p;
    *p = *q;
    *q = tmp;
}
int h(char* s, int n)
    if (s[n] == 0) {
        return 0;
    if (s[n+1]==0) {
        return 1;
    int k = h(s+1,n+1);
    swap(s,s+k+1);
    return k+2;
}
void func3()
    char s[] = "intro to cs";
    int x = h(s,0);
    printf("x = %d\n", x);
    printf("s = %s\n", s);
}
```

:func3() פלט של



-	
[



<u>סעיף ב</u>

עבור כל אחת מהפונקציות הבאות (func4(), func5(), כתבו את סיבוכיות הזמן והמקום שלהן <u>כתלות ב-n ו-m</u> במקום המתאים. כמו כן, תארו בקצרה (משפט אחד) מה מחזירה הפונקציה (func4().

```
int func4(int a[], int n, int m)
{
  int h[m];
  for (int i=0; i<m; ++i) {
    h[i] = 0;
  }
  for (int i=0; i<n; ++i) {
    h[a[i]]++; // 0..m-1 μπρι μπρι μπρι |
    }
  int x=0;
  for (int i=0; i<m; ++i) {
    x += i*h[i];
  }
  return x;
}</pre>
```

Θ ()	סיבוכיות מקום:	O () Θ (
		? func4() מהו ערך ההחזרה של הפונקציה

```
void func5(int a[], int n, int m, int b[])
{
   if (n==0) {
      return;
   }
   *b = func4(a,n,m);
   func5(a+1,n-1,m,b+1);
}
```



-	



<u>שאלה 2 (25 נקודות)</u>

<u>סעיף א</u>

נתון מערך מטיפוס double, ממוין בסדר עולה. עליכם לממש פונקציה המקבלת את המערך []a, את גודלו n, וערך נוסף x, ומחזירה את אינדקס האיבר מ-a שקרוב ביותר בערכו ל-x. לדוגמה, עבור המערך הבא:

1.3 5.2 6.0	7.1	9.8	12.3	14.5
-------------	-----	-----	------	------

. במערך x- במערך הקרוב ביותר ל-x ביותר ל-ניוון ש-a[1] הינו הערך הקרוב ביותר ל-x במערך.

<u>הערות:</u>

- .a ניתן להניח שאין איברים החוזרים על עצמם במערך
- במידה ויש שני מספרים במרחק זהה ל-x, יש להחזיר אחד מהם לבחירתכם.
- מומלץ לממש פונקציית עזר המחשבת את המרחק (בערך מוחלט) בין שני מספרים.
 - . $oldsymbol{o}(1)$ על הפונקציה לעמוד בסיבוכיות זמן Oig(log(n)ig) וסיבוכיות מקום נוסף •

<pre>int closest(double a[], int n, double x) {</pre>





<u>סעיף ב</u>

ממשו פונקציה המקבלת מערך [a], את גודלו n, ושני מספרים min ו-max (כאשר max), ומחזירה (משר max > min את מספר האיברים [a] במערך המקיימים min < a[i] במערך המקיימים a[i] במערך המקיימים min < a[i] למשל, עבור המערך מהסעיף הקודם ועבור amax-10, min =5.2, 6.0, 7.1, 9.8 בתחום בין

<u>הערות:</u>

- .a ניתן להניח שאין איברים החוזרים על עצמם במערך
- .max או min-ניתן להניח שאין במערך a איברים השווים ל
- בסעיף זה ניתן להשתמש בפונקציה מהסעיף הקודם, גם אם לא מימשתם אותה.
- . $oldsymbol{o}(1)$ על הפונקציה לעמוד בסיבוכיות זמן O(log(n)) וסיבוכיות מקום נוסף $oldsymbol{o}$

<pre>int range(double a[], int n, double min, double max) {</pre>







<u>שאלה 3 (25 נקודות)</u>

סעיף א

בסעיף זה נתון מערך []s של מחרוזות, כאשר ידוע שהמחרוזות בעלות אורך שונה זה מזה. עליכם לממש פונקציה (בעמוד הבא) המקבלת את המערך s ואת אורכו n, ומחזירה את האינדקס של המחרוזת <u>השנייה הקצרה ביותר במערך,</u> כלומר, המחרוזת שהייתה מופיעה בתא השני במערך לו היינו ממיינים אותו בסדר עולה של אורך המחרוזת. למשל, עבור המערך הבא:

```
char* s[] = { "abcde", "a", "bc", "234114" };
```

הפונקציה תחזיר 2, כיוון ש-[2]s (המחרוזת "bc") הינה המחרוזת השנייה הקצרה ביותר במערך (אחרי המחרוזת "a" שהיא הקצרה ביותר).

<u>הערות:</u>

1. לצורך השאלה נתונה לכם פונקציית העזר:

```
int strlen(char* str);
```

המחזירה את אורכה של המחרוזת str.

2. ניתן להניח שיש לפחות שתי מחרוזות במערך.

<u>דרישות סיבוכיות:</u>

- מקום: על הפתרון לעמוד בסיבוכיות מקום (O(1)
- זמן: הניחו שפעולת חישוב של אורך מחרוזת הינה פעולה "כבדה" שיש לצמצם בביצועה. לפיכך, בפתרון השאלה מותר להפעיל את הפונקציה () strlen לכל היותר פעם אחת עבור כל מחרוזת במערך (וסה"כ ח פעמים לכל היותר). למען הסר ספק, כל פעולת חישוב של אורך מחרוזת, גם ללא קריאה ל-strlen.
 תחשב כקריאה ל-strlen.



-	



<pre>int second_shortest(char* s[], int n) {</pre>	



I	
I	



<u>סעיף ב</u>

בסעיף זה נתון מערך []s של מחרוזות, כאשר ידוע שכל המחרוזות בו בעלות אורך של לכל היותר m. שימו לב שבסעיף זה ייתכנו כמה מחרוזות בעלות אותו האורך במערך s. עליכם לממש פונקציה המקבלת את המערך s, את אורכו n, וכן את אורך המחרוזת המקסימאלית m, וממיינת את s בסדר עולה לפי אורך המחרוזת. למשל, עבור המערך הבא:

.O(n+m) וסיבוכיות מקום O(n*m) <u>דרישות סיבוכיות</u>: על הפונקציה לעמוד בסיבוכיות זמן

שימו לב שאין חשיבות לסדר בין מחרוזות שאורכן זהה.

הערה: בסעיף זה ניתן להשתמש בפונקציה (strlen מהסעיף הקודם.

<pre>void sort(char* s[], int n, int m) {</pre>	
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_



-	



<u>שאלה 4 (25 נקודות)</u>

ממשו פונקציה המקבלת מערך של שלמים []a, את גודלו n, וערך שלם diff, ובודקת אם ניתן לחלק את איברי המערך לשתי קבוצות כך שההפרש בין סכומיהן הינו diff. הפונקציה מחזירה true אם קיימת חלוקה כזו, או false אחרת. לדוגמה, עבור המערך:

6	1	2	4	8	5	3

ועבור ההפרש diff=1, הפונקציה תחזיר true כיוון שניתן לחלק את המערך לשתי קבוצות באופן הבא:

$$6+1+8=15$$
 $2+4+5+3=14$

<u>:הערות</u>

- שימו לב שכל איבר ב-a חייב להיות שייך לאחת משתי הקבוצות.
 - יש לפתור את השאלה ברקורסיה.

bool	split(int	a[], i	nt n, i	nt diff)	{	



1	



	_
	_
	_
	_
- 	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
	_
-	_
	_
	_
	_
	_
	_
	_
	_ _ _
	_ _ _
	_
	_





l ————————————————————————————————————

