



## מבוא למדעי המחשב מ' / ח' (234117 / 234114)

סמסטר אביב תשע"ב - מבחן מועד א', 6 יולי 2012

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

מספר סטודנט

• רשום/ה לקורס: 234117 / 234114

• תואר ראשון / לימודי חוץ / אחר (לפרט): \_\_\_\_\_

- משך הבחינה – 3 שעות.
- בדקו שיש 13 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- השימוש בחומר עזר כלשהו, כתוב או אלקטרוני, **אסור**.
- המבחן כתוב בלשון נקבה אך מתיחס לנבחנים ולנבחנות כאחד.
- ניתן להשתמש בפונקציות קלט-פלט סטנדרטיות והקצאת זיכרון ב-C. שימוש בכל פונקציה אחרת, לרבות כזו שהוגדרה במהלך הקורס, אסור. אתן יכולות להגדיר פונקציות עזר כרצונכן. אין צורך להצהיר עליהן.
- כל זיכרון שאתן מקצות, אתן חייבות בשחרורו.
- אין צורך בהוכחות לחישובי הסיבוכיות.
- ניתן לכתוב בעיפרון ולהשתמש במחק.

**צוות הקורס :**  
**סמסטר אביב תשע"א :**  
**מרצים:** ד"ר ניר אילון (מרצה אחראי),  
דן רביב, תמיר לוי  
**מתרגלים:** חביאר טורק (מתרגל אחראי),  
אבישי גרץ, תהילה מייזלס, רן זילברשטיין,  
רועי פורן

שאלה	ערך	ציון	בודק
1	25		
2	25		
3	25		
4	25		
סה"כ	100		

**בהצלחה !**

2





## הטכניון, מכון טכנולוגי לישראל מבוא למדעי המחשב מ'/ח'

[illegible]



## שאלה 2 (25 נקודות) :

עליכן לכתוב פונקצייה שחתימתה :

```
int compress(int a[], int n);
```

הפונקציה מקבלת כקלט מערך של מספרים **חיוניים** (הפרמטר  $a$ ) וכן את אורכו (הפרמטר  $n$ ) ו"דוחסת" את תוכן המערך, לפי הכלל הבא. אם מספר  $x$  מופיע ברצף  $k$  פעמים במערך ו-  $k$  **לפחות** 3, אז יש להחליף את הרצף בזוג המספרים  $k$  - ואחריו  $x$ . על הפונקציה להחזיר את אורך המערך לאחר הדחיסה.

לדוגמא, אם הקלט הוא המערך  $a$  הבא (באורך 7, משמאל לימין) :

3	4	4	5	3	6	7
---	---	---	---	---	---	---

אז הוא לא יעבור שום שינוי, מאחר שאין רצף של מספר המופיע יותר מפעמיים. הערך שיוחזר הוא 7.

לעומת זאת, הקלט הבא :

5	5	6	3	7	8	8	8	9
---	---	---	---	---	---	---	---	---

יידחס בפלט למערך הבא :

5	5	6	3	7	-3	8	9
---	---	---	---	---	----	---	---

והערך שיוחזר הוא 8.

דגשים :

1. הפלט נכתב באותו מערך  $a$  שבו מגיע הקלט.
2. אין צורך להתייחס למקום העודף במערך. לדוגמא, אם הקלט באורך 8 והפלט באורך 5, אז תוכן המקומות 5, 6 ו- 7 במערך הפלט לא משנה.
3. אין להקצות זיכרון באופן דינמי. פיתרון שיקצה זיכרון לא יזכה בניקוד מלא.

(המשך בעמוד הבא)

6

[illegible]

8



**שאלה 3 (25 נקודות) :**

שאלה זו עוסקת בפולינומים. פולינום הוא פונקציה מתמטית מהצורה הבאה:

$$f(x) = a_0 + a_1x + \dots + a_dx^d$$

כפי שרואים, פולינום נקבע ע"י מספר טבעי  $d \geq 0$  הנקרא *דרגה* (*degree*), ופרמטרים ממשיים  $a_0, a_1, \dots, a_d$  הנקראים *מקדמים* (*coefficients*).

## פולינומים נייצג באמצעות מבנה (struct) בצורה הבאה :

```
struct polynomial {
    int d;
    double* a;
}
typedef struct polynomial POLY;
```

דוגמא:

כדי לייצג את הפולינום  $1+2x+3x^2$  יש לאתחל את d ל-2, ואת a למערך באורך לפחות 3 המאותחל כך ש  $a[0]==1, a[1]==2, a[2]==3$ .

כתבו פונקצייה שמקבלת כקלט מצביע לפולינום  $p$  ומשתנה ממשי  $x$ , ומחזירה את הערך של הפולינום בנקודה  $x$ . על הפונקציה לעבוד בסיבוכיות זמן  $O(d)$ . סיבוכיות המקום הנוסף הדרושה היא  $O(1)$ . חתימתה:

```
double evaluate(POLY* p, double x);
```

לדוגמא: אם הפולינום הוא  $3x^2 + 2x + 1$  ו- $x=4$  אז יש להחזיר 57.

הערות: בשאלה זו אינכם נדרשות לכתוב פונקציות אתחול או שחרור של פולינום. ניתן להניח שפולינום הקלט כבר מאותחל כראוי, כלומר ש-  $d \geq 0$  וששדה המקדמים מצביע למערך באורך לפחות  $d+1$ .

אם כתבתם פתרון בסיבוכיות זמן שהיא לא  $O(d)$ , תוכלו לקבל בחזרה חלק מהנקודות אם תכתבו כאן את הסיבוכיות של הפיתרון שלכם:

```
double evaluate(POLY* p, double x) {
```

10



## שאלה 4 (25 נקודות) :

בבנק הישראלי הידוע capital-pig, בכל פעם שלקוח מבצע משיכה ונכנס למשיכת יתר (מינוס), הבנק גובה 5 ש"ח עמלה, גם אם הלקוח כבר היה במשיכת יתר לפני הפעולה. לדוגמא:

נניח שהמאזן של הלקוח הוא 30 בזכות, והוא מבצע משיכה בסכום של 40 ואח"כ עוד משיכה בסכום של 7. לאחר הפעולה הראשונה תהייה היתרה 15 - (40 ירדו בגלל המשיכה עצמה, ועוד 5 עמלת משיכת היתר). לאחר הפעולה השנייה תהייה היתרה 27 - (7 ירדו בגלל המשיכה עצמה, ועוד 5 עמלת משיכת היתר).

לעומת זאת, אם הלקוח היה הופך את סדר הפעולות, כלומר קודם מבצע משיכה של 7 ורק אח"כ משיכה של 40, אז היתרה שלו בסוף היתה 22 - מכיוון שהפעולה הראשונה לא הכניסה אותו למשיכת יתר.

שימו לב: אין עמלות על הפקדות.

הבנק החליט לשנות את סדר הפעולות של הלקוח כדי למקסם את רווחיו (של הבנק). כדי שהלקוח לא ישלם לב, אסור שאף פעולה תזוז ביותר מ-3 מקומות. לדוגמא, בטבלה הבאה סידור א' הוא חוקי בעוד שסידור ב' לא חוקי מכיוון שהפעולה 14 - עברה ממקום 6 למקום 2, כלומר זזה 4 מקומות.

i	0	1	2	3	4	5	6
transactions[i]	10	-15	3	4	20	30	-14
סידור א'	10	-15	3	4	-14	20	30
סידור ב'	10	-15	-14	3	4	20	30

עליכן לכתוב פונקציה שמקבלת מערך פעולות transactions באורך n ויתרה התחלתית x. (פעולה חיובית מציינת הפקדה, ופעולה שלילית מציינת משיכה). על הפונקציה להדפיס את מקסימום העמלה שהבנק יכול לגבות מסידור חוקי של הפעולות. חתימת הפונקציה:

```
void max_capital_pig(int transactions[], int n, int x);
```

לדוגמא, אם  $n=7$  ומערך ה-transactions הוא כמו בטבלה ו- $x=0$ , אז יש להדפיס Maximal fees: 10

(פתרון זה מתקבל מסידור א'. אין זה הסידור היחיד שמוביל לעמלה כזו).

יש להשתמש בטכניקת ה-backtracking שנלמדה בשיעור, ויש לחתור לקיצוץ מירבי של מרחב אפשרויות החיפוש. אין להשתמש במשתנים סטטיים או גלובליים. מותר להגדיר פונקציות עזר.

[illegible]

[illegible]