



מבוא למדעי מחשב מ' / ח' (234114 / 234117)

סמסטר אביב תשס"ה

מבחן מסכם מועד א', 5 יולי 2005

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
שם פרטי	שם משפחה	מספר סטודנט							

משך המבחן: 2.5 שעות.
חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

הנחיות והוראות:

- מלאו את הפרטים בראש דף זה.
- בדקו שיש 22 עמודים (5 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 לכתוב תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק.
- אין לכתוב הערות והסברים לתשובות אם לא נתבקשתם מפורשות לכך.
- בכל השאלות, הינכם רשאים להגדיר (ולממש) פונקציות עזר כרצונכם.
- אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה אלא אם צוין אחרת בשאלה.

צוות הקורס 234114
מרצים: איתן אביאור, דר' רועי פרידמן (מרצה אחראי).
מתרגלים: רן רובינשטיין, מיכל הולצמן-גזית.

צוות הקורס 234117
מרצים: רועי מלמד, דר' רועי פרידמן (מרצה אחראי).
מתרגלים: עידו פלדמן, גיא פליישר, אייל רוזנברג, אולג רוכלנקו.

שאלה	ערך	הישג	בודק
1	20		
2	20		
3	20		
4	20		
5	20		
סה"כ	100		

בהצלחה!



- 2 -



שאלה 1 (20 נקודות)

סעיף א (8 נקודות)

נתון האתחול הבא של משתנים בתוכנית C:

```
char *arr[]={"Programming", "using", "C language", "is fun!"};  
char *p = arr[3];  
char *q = *(arr + 1);  
*(q+2) = '\\0';
```

כתבו מה יודפס ע"י כל אחת מן הפקודות הבאות לאחר ביצוע האתחול הנ"ל:

<code>printf("%s", arr[2]);</code>	
<code>printf("%s", arr[2]+2);</code>	
<code>printf("%s", q);</code>	
<code>printf("%c", arr[2][0]);</code>	
<code>printf("%s", p+3);</code>	



- 4 -



סעיף ב (12 נקודות)

כתבו את סיבוכיות הזמן והמקום של הקריאות $\text{strange}(n, 1)$ ו- $\text{weird}(n, 1)$ כפונקציה של n (כאשר $n \geq 0$ טבעי):

1.

```
void strange(int n, int k)
{
    int i;

    if (k > n)
        return;

    for (i = k; i < n; i++)
        printf("?");

    strange(n, k+2);
    return;
}
```

סיבוכיות זמן: $\Theta(\quad)$ סיבוכיות מקום: $\Theta(\quad)$

2.

```
void weird(int n, int k)
{
    int i;

    if ((n <= 0) || (k <= 0))
        return;

    k *= 2;

    for (i = 0; i < k; i++) {
        printf("?");
        weird(n/2, -n/2);
    }

    weird(n/2, k);

    return;
}
```

סיבוכיות זמן: $\Theta(\quad)$ סיבוכיות מקום: $\Theta(\quad)$

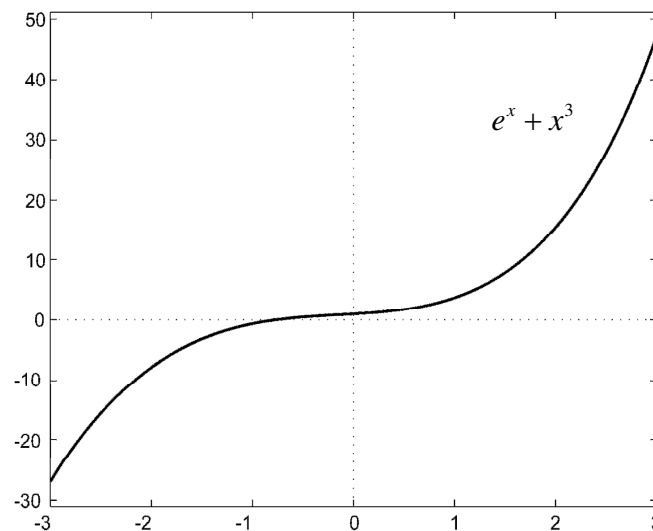


- 6 -



שאלה 2 (20 נקודות)

בשאלה זו נכתוב תוכנית המוצאת את נקודת האפס של הפונקציה $f(x) = e^x + x^3$ (ישנה נקודה יחידה כזו).
להלן גרף המתאר פונקציה זו באופן סכימטי:



סעיף א (10 נקודות)

בסעיף זה נכתוב פונקציה המחשבת את הפונקציה e^x . לשם כך, ידוע שלכל x מתקיים:

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \frac{x^4}{1 \cdot 2 \cdot 3 \cdot 4} + \dots$$

כאשר מובטח שהאיברים בזנב הטור הולכים וקטנים לאפס. השלימו את הפונקציה בעמוד הבא, המקבלת מספר x ומחשבת קירוב ל- e^x ; על הפונקציה לעצור כאשר התיקון שמתווסף לקירוב הנוכחי קטן בערכו המוחלט מ- ϵ (מוגדר כ-`#define`).

שימו לב: ניתן להעזר בפונקציה `fabs()` שמחשבת ערך מוחלט של מספר ממשי, וחתירתה היא

```
double fabs(double x);
```

אין להעזר בכל פונקציה אחרת בקוד שלכם (ובפרט לא בפונקציה `pow()`).



- 8 -



```
#define EPSILON 1e-10

double myexp(double x)
{
    double result = 1;
    _____
    _____
    _____

    do {
        _____
        _____
        _____
        _____
        _____
        _____

    } while ( _____ > EPSILON);

    return result;
}
```

לנוחותכם, הנוסחה מובאת כאן בשנית:

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \frac{x^4}{1 \cdot 2 \cdot 3 \cdot 4} + \dots$$

סעיף ב (10 נקודות)

נתונה הפונקציה הבאה המחשבת את $f(x) = e^x + x^3$ בנקודה x , ועושה שימוש בפונקציה מהסעיף הקודם:

```
double f(double x) {
    return myexp(x) + x*x*x;
}
```



- 10 -

הטכניון, מכון טכנולוגי לישראל מבוא למדעי המחשב מ"ח'

כתבו פונקציה שמוצאת את נקודת האפס של $f(x)$ (הנקודה בה ערך הפונקציה הוא 0). פונקציה זו תקבל שתי נקודות קצה x_{\min} ו- x_{\max} , כאשר $x_{\min} < x_{\max}$, ומתקיים ש- $f(x_{\min}) < 0$ ו- $f(x_{\max}) > 0$; על הפונקציה להחזיר את נקודת האפס של f הנמצאת בין שתי נקודות הקצה (מובטח שיש כזו). הפונקציה נדרשת לעבוד בדיוק DELTA (קבוע נוסף המוגדר כ-#define), כשהכוונה היא שהנקודה x המוחזרת צריכה להיות לכל היותר במרחק DELTA מנקודת האפס האמיתית x_0 . (שימו לב שהתנאי על הנקודה המוחזרת x איננו תלוי בערך הפונקציה עצמו בנקודה זו.)

בסעיף זה מותר להשתמש בפונקציה $f(\cdot)$ הנתונה גם אם לא פתרם את הסעיף הקודם.

```
#define DELTA 1e-8
```

```
double findzero(double xmin, double xmax) {
```

This image shows a full page of blank, lined paper. It features approximately 20 evenly spaced horizontal grey lines across its entire width, providing a template for handwriting practice or general note-taking. The margins are consistent on all sides.



- 12 -



- 14 -



```
int common2(int a[], int n) {
```

[illegible]



- 16 -



- 18 -



שאלה 5 (20 נקודות)

בשאלה זו עליכם לכתוב פונקציה המקבלת שני מספרים טבעיים n ו- k , ומדפיסה את כל האפשרויות להציג את n כסכום של k מספרים טבעיים (לשאלה זו מספר טבעי הוא שלם ≥ 0). יש להדפיס כל אפשרות פעם אחת בלבד, ואין לחזור על אותה האפשרות פעמיים – גם לא בסדר שונה. על הפתרון להיות רקורסיבי ולעבוד בשיטת ה-**backtracking**.

שימו לב: בשאלה זו ניתן להשתמש בפונקציית הספרייה `malloc()`.

לדוגמה, עבור $n=5$ ו- $k=3$ הפונקציה תדפיס את הפלט הבא:

```
5 0 0
4 1 0
3 2 0
3 1 1
2 2 1
```

עבור $n=5$ ו- $k=2$ יודפס הפלט הבא:

```
5 0
4 1
3 2
```

ממשו את הפונקציה `printsum()` המבצעת את הפעולה המתוארת (שימו לב שביכולתכם לממש פונקציות עזר על פי הצורך). בפתרונכם, ניתן להשתמש בפונקציה הבאה:

```
void printarray(int a[], int n);
```

המקבלת מערך a ואת גודלו n , ומדפיסה את תוכנו.

```
void printsum(unsigned int n, unsigned int k) {
```



- 20 -



- 21 -



- 22 -