



**מבוא למדעי מחשב מ' 234114**  
**מבוא למדעי מחשב ח' 234117**

**סמסטר חורף תשס"ה**  
**2 בפברואר 2005**

**מבחן מועד א'**

**פרטי הנבחן/נת**

**שם פרטי**

**שם משפחה**

**מספר סטודנט**

- משך המבחן 3 שעות. לא תינתן הארכת זמן -**  
**- השימוש בכל בחומר עזר מודפס או אלקטרוני אסור בהחלט -**

**הנחיות והוראות:**

- יש למלא את הפרטים לעיל.
- בדקו שיש 18 עמודים (6 שאלות) כולל עמוד זה.
- תחילה מומלץ לקרוא את כל המבחן ולתכנן את זמנכם בהתאם.
- יש לכתוב את התשובות על טופס המבחן בלבד ובמקומות המיועדים לכך. שימו לב שהמקום המיועד לתשובה אינו מעיד על אורך התשובה הנכונה. במקרה הצורך, תוכלו להשתמש בעמודים הריקים בסוף הבחינה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק.
- אין לכתוב הערות והסברים לתשובות.
- בכל השאלות, הינכם רשאים להגדיר (למשל) פונקציות עזר.
- בכל השאלות ניתן להניח שהקלט תקין.

**צוות הקורס 234114**

מרצים: פרופ' ח' רוני קימל, אלכס גונטמכר.

מתרגלים: סאהר אסמיר, רן רובינשטיין, גיא פליישר,  
עידו פלדמן, ליטל משיח.

**צוות הקורס 234117**

מרצים: רועי מלמד, אנה מוס.

מתרגלים: עמיר אדלר, גיא פליישר, אייל רוזנברג,  
מיכל הולצמן-גזית, שיאון שחורי, חיה זילברשטיין.

שאלה	ערך	הישג	בודק
1	16		
2	8		
3	14		
4	20		
5	22		
6	20		
סה"כ	100		

**בהצלחה!**

## שאלה 1 (16 נק')

מילה בשפה העתיקה "ארברית" מיוצגת ע"י מבנה (struct) המכיל שני שדות: מחרוזת ואורך המחרוזת.

```
struct ArabaritWord {
    char* s;    // a null terminated string
    int len;    // number of letters
};
```

להפתעת ההיסטוריונים נתגלה כי השפה ה-"ארברית" משתמשת באותו ה-"א"ב האנגלי המוכר לכולנו אולם חל בשפה איסור מוחלט על ערבוב בין אותיות קטנות (lower case) לגדולות (upper case) ולכן בתחילת כל מילה מופיעות כל האותיות הקטנות ורק אחריהן כל האותיות הגדולות. כן ייתכן שבמילה יש רק אותיות קטנות או רק אותיות גדולות.

כתבו פונקציה `num_of_lower()` המקבלת כפרמטר מילה ב-"ארברית" (משתנה מהטיפוס `struct ArabaritWord` כפי שתואר לעיל) ומחזירה את מספר האותיות הקטנות במילה. לדוגמה, עבור המלה "sababiBABI" הפונקציה תחזיר 6.

על הפונקציה להיות יעילה ככל האפשר (מבחינת זמן ומקום). עליכם לכתוב חסם הדוק ככל האפשר על סיבוכיות הזמן והמקום הנוסף (כפונקציה של len) במקומות המיועדים לכך בתחתית עמוד זה. פתרון נכון אך לא יעיל לא יזכה בנקודות.

```
int num_of_lower (struct ArabaritWord word)
```

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

סיבוכיות מקום נוסף:  $O(\quad)$

סיבוכיות זמן:  $O(\quad)$

## שאלה 2 (8 נק')

כתבו פונקציה `is_sub_number()` אשר בהינתן שני מספרים שלמים חיוביים  $x$  ו- $y$  בודקת האם מספר  $y$  מופיע כרצף ספרות בתוך  $x$ .

**דוגמאות:**

- המספר 114 מופיע בתוך המספר 234114.
- המספר 117 מופיע בתוך המספר 21171171.
- המספר 777 אינו מופיע בתוך המספר 21177.

הפונקציה תחזיר TRUE או FALSE בהתאם.

הנח ש-TRUE ו-FALSE הוגדרו באמצעות `#define`

ניתן להניח את קיומן של שתי פונקציות העזר הבאות:

```
unsigned int digits(unsigned int x); // returns the number of
                                     // digits in x
```

```
unsigned int pow10(unsigned int x); // returns 10x
```

```
int is_sub_number(unsigned int x, unsigned int y)
```

**שאלה 3 (14 נק')**

תמונה ספרתית מיוצגת במחשב כמערך  $x$  באורך  $n$  של "פיקסלים", המקבלים כל אחד ערך בתחום  $\{0, 1, \dots, 255\}$ . עליכם לחשב:

- מספר  $a$  אשר ייצג את הפיקסלים במובן של מזעור השגיאה הריבועית:  $\sum_{i=0}^{n-1} (a - x[i])^2$ .
- מספר  $b$  אשר ייצג את הפיקסלים במובן של מזעור השגיאה בערך מוחלט:  $\sum_{i=0}^{n-1} |b - x[i]|$ .

**סעיף א'**

מבין האפשרויות המובאות להלן, המשמעות של המספר  $a$  היא \_\_\_\_\_ והמשמעות של המספר  $b$  היא \_\_\_\_\_

(נא למלא את המספר המציין את התשובה הנכונה)

- (1) מספר הערכים השונים במערך
- (2) הערך השכיח (בעל מספר המופעים הגדול ביותר)
- (3) החציון
- (4) הממוצע
- (5) הערך המקסימאלי
- (6) הערך המינימאלי

**סעיף ב'**

מלאו את החסר בפונקציה הבאה, המקבלת מערך  $x$  באורך  $n$  המייצג תמונה ספרתית, ומדפיסה את שני המספרים הדרושים.

```
void calc_ab(int x[], int n) {  
    int hist[_____] = {0};  
    int i, sum = 0, b = 0;  
    for (i=0; i<n; i++) {  
        hist[_____]++;  
        sum += _____;  
    }  
    for( i=0; i<(_____/2); i += hist[_____]++ ) ;  
    printf("a is %lf\n", _____);  
    printf("b is %d\n", _____);  
}
```



## שאלה 4 (20 נק')

## סעיף א'

נתון מערך  $a$  של שלמים שיתכנו בו חזרות (ערכים זהים). כתבו פונקציה `partial_sort()` המקבלת כפרמטרים את כתובת המערך  $a$  הנ"ל ואת גודלו  $n$  ומסדרת את איבריו כך שכל האיברים השווים לערך הקטן ביותר במערך יופיעו בתחילתו (אין מגבלה על הסדר היחסי בין שאר האיברים). על הפונקציה להחזיר את מספר האיברים במערך השווים לערך הקטן ביותר. לדוגמה, עבור המערך  $a$  הבא:

2	3	2	7	4	5	2
---	---	---	---	---	---	---

המערך  $a$  עשוי להפוך לאחר ריצת הפונקציה למערך הבא (זוהי אחת האפשרויות):

2	2	2	3	7	4	5
---	---	---	---	---	---	---

והפונקציה תחזיר 3.

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

פתרון אשר אינו עונה על דרישות הסיבוכיות הנ"ל לא יזכה בניקוד.

מותר לבצע שינויים במערך  $a$  אך ורק ע"י שימוש בפונקציה `swap()` אשר הוגדרה בכיתה ומובאת כאן לנוחיותכם:

```
void swap(int* x, int* y)
{
    int tmp = *x;
    *x = *y;
    *y = tmp;
}
```

נא לכתוב את התשובה במקום המיועד לכך בעמוד הבא...

```
int partial_sort (int a[], int n)
```

[illegible]



## סעיף ב'

נתון מערך  $a$  של שלמים בגודל  $n$ , וידוע שבין  $n$  איברי המערך יש רק  $k$  ערכים שונים זה מזה (כלומר ישנם איברים החוזרים על עצמם במערך). כתבו פונקציה רקורסיבית  $\text{sort}()$  המקבלת את המערך  $a$  וגודלו  $n$  וממינת את המערך. לפונקציה  $\text{sort}()$  מותר להשתמש בפונקציה  $\text{partial\_sort}()$  מהסעיף הקודם גם אם לא עניתם על סעיף א'.

דרישות סיבוכיות: זמן  $O(k \cdot n)$  ומקום נוסף  $O(k)$ .  
פתרון אשר אינו עונה על דרישות הסיבוכיות הנ"ל / אינו רקורסיבי לא יזכה בניקוד.

```
void sort (int a[], int n)
```

## שאלה 5 (22 נק')

נאמר שמערך  $a$  הינו מערך ממויין-בדילוגי- $m$  אם כל איבר  $a[i]$  במערך קטן או שווה לכל האיברים  $a[i+l \cdot m]$ , כאשר  $l \in \mathbb{N}$ . לדוגמה, המערך

7	<u>2</u>	1	5	7	<u>6</u>	3	6	8	<u>9</u>	5	8
---	----------	---	---	---	----------	---	---	---	----------	---	---

הינו מערך ממזין-בדילוגי-4 (סדרת האיברים ברקע הלבן ממוינת, כנ"ל סדרת האיברים ברקע השחור וכו').  
בשאלה זו נניח כי אורכי המערכים וערכי  $m$  הינם חזקות שלמות של 2.

## סעיף א'

ממשו את הפונקציה `mreduce()` המקבלת את הפרמטרים הבאים:

a	מערך ממויין-בדילוג- $m$ .
aux	מערך-עזר באותו אורך כמו $a$ . ניתן להשתמש בו כרצונכם ואין מגבלה על הערכים שבו בתום ביצוע הפונקציה.
n	אורך המערכים $a$ ו- $aux$ ; ניתן להניח שזוהי חזקה של 2.
m	גודל ה'דילוג' של הממויינות במערך $a$ ; ניתן להניח שזוהי חזקה של 2.

והופכת את  $a$  ממערך ממויין-בדילוגי- $m$  למערך ממויין-בדילוגי- $m/2$ .

**דוגמה:**

המערך לעיל עשוי להפוך לאחר ריצת הפונקציה למערך הבא (זוהי אחת האפשרויות):

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

אשר בו כל רצף של איברים בדילוגים של 2 מאחד לשני הינו רצף ממויין.  
**דרישות סיבוכיות:**  $O(n)$  זמן.

**פתרון אשר אינו עונה על דרישות הסיבוכיות הנ"ל לא יזכה בניקוד.**

```
void mreduce(int a [], aux[], int n, int m)
```

[illegible]



This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the paper.

## סעיף ב'

כתבו פונקציה `msort()` המשתמשת בפונקציה `mreduce()` מסעיף א' לצורך מיון מערך (אתם רשאים להשתמש בה גם אם לא עניתם על סעיף א').

על הפונקציה לקבל את הארגומנטים הבאים:

a מערך ממויין-בדילוגי- $m$ .

aux מערך-עזר באותו אורך כמו a. ניתן להשתמש בו כרצונך ואין מגבלה על הערכים שבו

בתום ביצוע הפונקציה.

n אורך המערכים  $aux-i$ ; ניתן להניח שזוהי חזקה של 2.

גודל ה'דילוג' של הממיינות במערך <sup>a</sup>; ניתן להניח שזוהי חזקה של 2.

**דרישות סיבוכיות:**  $O(n \cdot \log(m))$  זמן.

**פתרון אשר אינו עונה על דרישות הסיבוכיות הנ"ל לא יזכה בניקוד.**

```
void msort(int a [], int aux[], int n, int m)
```

This image shows a full page of blank, lined paper. It features approximately 20 evenly spaced horizontal grey lines across the entire width of the page, typical of notebook or composition paper. The background is white, and there are no margins, text, or other markings present.



## שאלה 6 (20 נק')

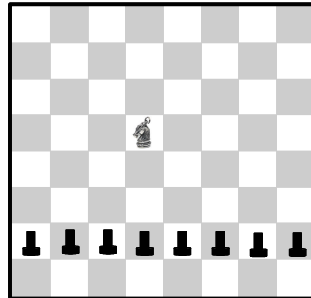
בתרגולים נלמדה בעיית הפרש: בהינתן נקודת התחלה על לוח שחמט  $N \times N$ , עלינו למצוא מסלול של צעדי פרש המתחיל בנקודה זו ומבקר בכל משבצות הלוח פעם אחת בדיוק. הקוד לבעיה זו, כפי שנלמד בכיתה פרט לפונקציה `solve()`, מופיע בהמשך.

בשאלה זו נפתור וריאציה על בעיית הפרש. נתונים  $N$  חיילים אשר ניצבים בשורה האחת לפני האחרונה (השורה ה- $N-2$ ) ומוגדרים במערך גלובלי בשם `line` (ראה שורה נטויה בקוד). `line[i] == 0` אם ורק אם החייל בעמודה ה- $i$  עדיין לא נאכל על ידי הפרש (כאשר הפרש מגיע למשבצת בא מצוי חייל נאמר כי הפרש "אכל" את החייל).

הבעיה שעליכם לפתור (על ידי השלמת הפונקציה `solve_exam()`) היא כדלקמן: בהינתן נקודת התחלה על לוח שחמט  $N \times N$  (הנח כי בתחילה הפרש אינו ממוקם במשבצת בה ניצב חייל), עליכם למצוא מסלול של צעדי פרש מנקודת ההתחלה כך שבמסלול זה הפרש "אוכל" את כל  $N$  החיילים, וזאת מבלי לעבור דרך אותה משבצת פעמיים. הפונקציה תחזיר 1 אם נמצא מסלול כזה ו-0 אחרת.

שימו לב: בשאלה זו על המסלול להסתיים כאשר החייל האחרון נאכל (כלומר כאשר החייל האחרון נאכל על הפונקציה להחזיר 1 באופן מיידי). בפרט, אין חובה על הפרש לבקר בכל משבצות המטריצה: מטריצת מהלכי הפרש צריכה להכיל אפסים במשבצות שבהן הפרש לא ביקר.

על הפתרון להיות בשיטת ה-backtracking.



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

#define N 8

int board[N][N] = {0};
int line[N] = {0};

int path_len = 0;

//=====

int inboard(int row, int col)
{
    if (row>=0 && row<N && col>=0 && col<N)
        return 1;
    return 0;
}

//=====

int legal_pos(int row, int col)
{
    return (inboard(row,col) && board[row][col]==0);
}
```



```

void get_legal_moves(int row, int col, int moves[8][2], int *n)
{
    *n = 0;
    if (legal_pos(row-2,col-1)){
        moves[*n][0]=row-2; moves[*n++][1]=col-1;}
    if (legal_pos(row-2,col+1)){
        moves[*n][0]=row-2; moves[*n++][1]=col+1;}
    if (legal_pos(row-1,col-2)){
        moves[*n][0]=row-1; moves[*n++][1]=col-2;}
    if (legal_pos(row-1,col+2)){
        moves[*n][0]=row-1; moves[*n++][1]=col+2;}
    if (legal_pos(row+1,col-2)){
        moves[*n][0]=row+1; moves[*n++][1]=col-2;}
    if (legal_pos(row+1,col+2)){
        moves[*n][0]=row+1; moves[*n++][1]=col+2;}
    if (legal_pos(row+2,col-1)){
        moves[*n][0]=row+2; moves[*n++][1]=col-1;}
    if (legal_pos(row+2,col+1)){
        moves[*n][0]=row+2; moves[*n++][1]=col+1;}
}

//=====

void print_board()
{
    int i, j;
    for (i=0; i<N; ++i) {
        for (j=0; j<N; ++j)
            printf("%5d", board[i][j]);
        printf("\n\n");
    }
}

//=====

// The function you have to implement
int solve_exam (int row, int col);

//=====

int main()
{
    int begin_row = 0, begin_col = 0;

    board[begin_row][begin_col] = 1;
    path_len = 1;

    if (solve_exam(begin_row,begin_col)) {
        printf("\nsolution found for N=%d!\n\n", N);
        print_board();
    }
    else
        printf("no solution for N=%d.\n", N);
    return 0;
}

```

```
int solve_exam (int row, int col)
```

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the page.

## הטכניון מ.ט.ל - הפקולטה למדעי המחשב

[illegible]

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the page.



This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the paper.

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or printed text on the page.