

## מבוא למדעי המחשב – אביב 2023 – מטלה 4

**נושאים:** מערכים רב ממדיים, מחרוזות.

**משקל מציון התרגיל:** 15%

**תאריך הגשה:** 22.05.2023 23:50

**הגשה באיחור:** ניתן להגיש באיחור של יום (בהורדת 10 נקודות – ציון מקסימלי 90), יומיים (בהורדת 20 נקודות – ציון מקסימלי 80) או שלושה ימים (בהורדת 30 נקודות – ציון מקסימלי 70). לאחר מכן לא תתאפשר ההגשה (מלבד לאיחורים מוצדקים לפי תקנון האוניברסיטה).

### הנחיות כלליות

שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי** בפורום הייעודי למטלה הנמצא במודל.

בקשות להארכה מסיבות מוצדקות (מילואים, לידה, אשפוז וכו') יש לשלוח למייל [tom.ben-dor@biu.ac.il](mailto:tom.ben-dor@biu.ac.il) בצירוף: שם מלא, שם משתמש במערכת ההגשה, מספר תעודת זהות ומסמכים רלוונטיים לפי הצורך.

יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות. אין להוסיף או להשמיט רווחים או תווים אחרים, ואין להחליף אותיות גדולות בקטנות או להיפך. חוסר הקפדה על פרטים אלו עלול לגרור הורדה משמעותית ביותר בציון התרגיל עד כדי 0. ראו עצמכם הוזהרתם!

שימו לב שאתם עוקבים במדויק אחרי ההנחיות במסמך ה-Style Guide המפורסם באתר הקורס.

עליכם לכתוב קוד על פי הוראות התרגיל ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על שרתי המחלקה (planet) **ללא שגיאות וללא אזהרות**. תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו הורדת נקודות משמעותית בציון שלו.

להזכיר העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. העתקות נבדקות על ידי מערכת ההגשה האוטומטית, ותרגיל שהועתק יגרור בין השאר ציון 0 ופגיעה בציוני התרגול הסופיים **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.

בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה, ולהגיש אותם באמצעות מערכת ההגשה.

בפיתוח הקוד אין להיעזר בכלים מונחי למידת מכונה מכל סוג, על כל קטע קוד להיכתב על ידי המגיש/ה בלבד ובאופן עצמאי.

דוגמת הרצה של התרגיל נמצאת בסוף המסמך.

## תרגיל 4 (בשורה)

בתרגיל זה תממשו משחק ארבע בשורה (Connect Four).

בניגוד למשחק ארבע בשורה קלאסי, בתרגיל זה יהיו ארבעה נעלמים שתקבלו בתור DEFINE בשורת הקומפילציה כפי שנלמד:

- ROWS – מספר השורות בלוח המשחק.
- COLS – מספר העמודות בלוח המשחק.
- NUM\_PLAYERS – מספר המשתתפים במשחק.
- CONNECT – המשחק יהיה למעשה CONNECT-בשורה.

בעת הגשת התרגיל הפידבק שתקבלו יהיה לפי ערכי המשתנים הבאים:

ROWS = 6, COLS = 7, NUM\_PLAYERS = 2, CONNECT = 4

ערכים אלו ישתנו בעת הרצת הבדיקה המקיפה לאחר תום תקופת ההגשה, כך שהתוכנית שלכם צריכה לתמוך בערכים בגבול ההיגיון הסביר (לא יוכנסו ערכים קיצוניים או לא הגיוניים).

חוקי המשחק ללא שינוי ([https://en.wikipedia.org/wiki/Connect\\_Four](https://en.wikipedia.org/wiki/Connect_Four)).

שחקנים מזוהים באמצעות אותיות גדולות באנגלית, לפי הסדר. לדוגמה: אם יש שני משתתפים, המשתתף הראשון יהיה A והשני B. לא יוכנס מספר משתתפים גדול מ-26.

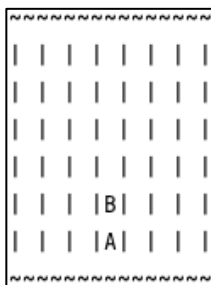
ראשית מאתחלים לוח ריק `char board[ROWS][COLS]`, כל התאים בו יכילו את התו רווח.

מהלך המשחק:

התורות הם לפי הסדר – שחקן A ראשון, שחקן B שני וכך הלאה.

לאחר תור השחקן האחרון מתחיל שוב סבב התורות עם שחקן A.

בכל תור שחקן בוחר עמודה ומכניס לתוכה את התו המזהה אותו. לדוגמה אם השחקן A בחר בעמודה 3 ולאחר מכן שחקן B בחר בעמודה 3, הלוח יראה כך:



שימו לב ששני תווים בלוח הם 'A' ו-'B' והשאר רווחים.

כאשר אחד השחקנים מצליח להציב CONNECT מהתווים שלו ברצף (בשורה, עמודה או באלכסון) הוא מנצח.

## הקדמה

בתרגיל זה לא יהיה תפריט - תתבקשו לממש פונקציות לפי חתימה. היצמדו **במדויק** לחתימה. המנעו משכפול קוד - כתבו פונקציות עזר לפי הצורך. אין לגשת לקבועים (define) מתוך הפונקציות, המידע שתצטרכו יתקבל כפרמטר.

## אתחול לוח ריק

```
void initBoard(char board[ROWS][COLS], int rows, int columns);
```

הפונקציה תקבל לוח עם ערכים לא ידועים ותאפס אותו כך שיהיה מוכן למשחק.

## הדפסת לוח

```
void printBoard(char board[ROWS][COLS], int rows, int columns);
```

הפונקציה תקבל לוח שעבר אתחול (יתכן שלוח באמצע משחק) ותדפיס אותו בפורמט הבא:

- מודפסות  $ROWS + 2$  שורות.
- בשורה הראשונה ובשורה האחרונה מודפסות  $2 \cdot COLS + 1$  טילדות (~).
- באמצע, עבור כל שורה בלוח מודפסים התווים בה, מופרדים בקווים מאונכים (|).

דוגמת הרצה:

```
~~~~~  
| | | | | | | |  
| | | | | | | |  
| | | | | | | |  
| | | |C| | | |  
| | |B|B| | | |  
|A|B|A|A|C|C|  
~~~~~
```

## ביצוע מהלך

```
int makeMove(char board[ROWS][COLS], int rows, int columns, int players, char player, int column);
```

הפונקציה תעדכן את הלוח להיות הלוח המתקבל לאחר שהשחקן הנתון בחר בעמודה הנתונה. אם מסיבה כלשהי הפעולה אינה אפשרית, יש להחזיר 0. אם הפעולה בוצעה בהצלחה, יש להחזיר 1.

## ביטול מהלך

```
int undoMove(char board[ROWS][COLS], int rows, int columns, int column);
```

הפונקציה תעדכן את הלוח להיות הלוח המתקבל לאחר ביטול הפעולה האחרונה שהתבצעה בעמודה הנתונה. אם מסיבה כלשהי הביטול אינו אפשרי, יש להחזיר 0. אם הפעולה בוצעה בהצלחה, יש להחזיר 1.

## מצב המשחק

```
int getStatus(char board[ROWS][COLS], int rows, int columns, int players, int connect);
```

הפונקציה תקבל לוח משחק תקין (כזה שהתקבל לאחר מספר כלשהו של תורות בהם בוצעו מהלכים חוקיים) ותכריע האם המשחק הסתיים ויש מנצח (תחזיר 1), המשחק הסתיים בתיקו (תחזיר 0) או שהמשחק עודנו מתנהל (תחזיר -1).

## מציאת מנצח

```
char getWinner(char board[ROWS][COLS], int rows, int columns, int players, int connect);
```

הפונקציה תקבל לוח משחק תקין (כזה שהתקבל לאחר מספר כלשהו של תורות בהם בוצעו מהלכים חוקיים) ותחזיר מי המנצח. במידה ואין מנצח הפונקציה תחזיר -1.

## אימות לוח

```
int isValidBoard(char board[ROWS][COLS], int rows, int columns, int players, int connect);
```

הפונקציה תקבל לוח ותכריע האם יתכן שהוא לוח חוקי שהתקבל בהתחלה, באמצע או בסיום משחק תקני לפי הכללים שתוארו במסמך. השחקנים אינם בהכרח רציונליים. אם הלוח חוקי יוחזר 1, אחרת יוחזר 0.

שימו לב, אופן פתרון סעיף זה מהווה חידה – עליכם לחשוב אילו קריטריונים מתאימים לצורך אימות לוח (רמז: יש יותר מאחד) ולחשוב כיצד ניתן לבדוק אותם.

רמז נוסף: רקורסיה.

ניתן להניח שבלוח לא יהיו יותר מ-20 תאים תפוסים (הנחה זו לא מקלה על הפתרון אבל מבטיחה זמן ריצה סביר – עד 3 שניות בשרת).



## הוראות הגשה

יש להגיש את הפונקציות בקובץ בודד בשם ex\_4.c לקבוצה מספר 14. מצורף הקובץ ex\_4.h (בנספח). הורידו אותו לתיקייה בה נמצא ex\_4.c, כללו אותו בקובץ ex\_4.c באמצעות include אך אל תגישו אותו. אין לצרף פונקציית main. אין להגדיר את ערכי הקבועים באופן מפורש בקובץ.

## דוגמת הרצה

כאמור, דוגמה זו היא **חלקית**, עליכם לחשוב על כלל מקרי הקצה האפשריים בתוכנית ולבדוק אותם.

```
main.c
#include "ex_4.h"

int main() {
    char board[ROWS][COLS];
    initBoard(board, ROWS, COLS);

    printBoard(board, ROWS, COLS);

    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'A', 3);
    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'B', 3);
    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'A', 0);
    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'B', 1);
    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'A', 3);
    makeMove(board, ROWS, COLS, NUM_PLAYERS, 'B', 2);
    undoMove(board, ROWS, COLS, 3);
    undoMove(board, ROWS, COLS, 3);

    printBoard(board, ROWS, COLS);

    int status = getStatus(board, ROWS, COLS, NUM_PLAYERS, CONNECT);
    if (status == 1) {
        printf("Game-over\n");
    } else if (status == 0) {
        printf("Tie\n");
    } else if (status == -1){
        printf("Game is in progress\n");
    } else {
        printf("Invalid return value from getStatus(...)\n");
    }

    char winner = getWinner(board, ROWS, COLS, NUM_PLAYERS, CONNECT);
    if (winner == -1) {
        printf("No winner\n");
    } else {
        printf("Winner is '%c'\n", winner);
    }

    int valid = isValidBoard(board, ROWS, COLS, NUM_PLAYERS, CONNECT);
    if (valid) {
        printf("Board is valid\n");
    } else {
        printf("Board is invalid\n");
    }
}
```



## נספח

ניתן למצוא את הקבצים main.c ו-ex\_4.h גם [בקישור](#).

### ex\_4.h

```
/**
 * @file ex_4.h
 * @brief Header file for ex_4.c
 * @note Do not modify this file or submit it, we will use our own copy of it.
 */

// Do not include <stdio.h> in ex_4.c, only include "ex_4.h"
#include <stdio.h>

void initBoard(char board[ROWS][COLS], int rows, int columns);
void printBoard(char board[ROWS][COLS], int rows, int columns);
int makeMove(char board[ROWS][COLS], int rows, int columns, int players, char player, int column);
int undoMove(char board[ROWS][COLS], int rows, int columns, int column);
int getStatus(char board[ROWS][COLS], int rows, int columns, int players, int connect);
char getWinner(char board[ROWS][COLS], int rows, int columns, int players, int connect);
int isValidBoard(char board[ROWS][COLS], int rows, int columns, int players, int connect);
void encode(const char board[ROWS][COLS], int rows, int columns, char *code);
void decode(const char *code, char board[ROWS][COLS]);
```