

# מבוא למדעי המחשב מ'ח' (234114/7), סמסטר חורף 2017

## תרגיל בית 3

מועד אחרון להגשה: **יום ראשון 25/12/2016** עד שעה 23:00

המתרגל האחראי על תרגיל זה: דניאל עזוז

משרד: טאוב 420

E-mail: dandan@campus.technion.ac.il

שעות קבלה מיוחדות לשאלות על התרגיל (בנוסף על שעת הקבלה הרגילה) :

**יום רביעי, 14/12, 21/12, 15:00 עד 16:00**

אם שעות אלו אינן נוחות לכם ניתן לתאם פגישה בשעות אחרות.

החומר הרלוונטי לתרגיל : עד תרגול 6 כולל. מותר להשתמש בחומר שנלמד עד תרגול 8.

### הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרון.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל, עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל לפני תאריך הגשת התרגיל.
- ערעורים ניתן להגיש עד שבוע לאחר קבלת הציון.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה בחלקה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**

- השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
- וודאו את האותיות הגדולות והקטנות לפי הדוגמאות וההסברים בתרגיל.
- אין להדפיס רווחים שלא התבקשתם להדפיס (בתחילת שורה או בסופה).
- בכל סוף שורה יש להדפיס תו ירידת שורה, גם בשורה האחרונה.
- השתמשו באתר הבדיקה העצמית.

- בתרגיל זה מותר להשתמש בפונקציות מהספרייה `stdio.h`, `stdlib.h` **בלבד**, שנלמדו בהרצאות ובתרגולים. החומר הנדרש לתרגיל זה שייך להרצאות 1-6 ולתרגולים 1-6. אין להשתמש בחומר שאינו מופיע במצגות אלה.
- ההגשה הינה אלקטרונית וב**בודדים** דרך אתר הקורס. קובץ ההגשה יהיה מסוג `zip` (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
  - קובץ `students.txt` עם מספר תעודת הזהות שלך וכתובת האי-מייל שלך.
  - קובץ פתרון `hw3q1.c`.
- חובה לשמור את קוד אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!** אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

## הנחיות לתכנון וכתיבת קוד: חשוב לקרוא לפני התרגיל(!)

בתרגיל זה המטרה היא לתרגל אתכם בפירוק בעיה לגורמים קלים לתכנות (ותכנון). לצורך כך, ישנן כמה מגבלות על כתיבת הקוד. שימו לב, התוכנית תיבדק באופן ידני, ויורדו נקודות על חריגה מהכללים. בנוסף, הדגש בתרגיל זה הוא על תכנון נכון של הקוד, ובהתאם לכך יינתן לבדיקה הידנית משקל גבוה יותר מהרגיל: 67% מהציון.

1. אורך כל פונקציה לא יעלה על 16 שורות קוד (ראו הגדרות מדויקות בהמשך הדף). הגבלה זו תקפה לכל הפונקציות, כולל main. יש לרשום את האורך בהערה לפני הפונקציה כדי להקל על הבדיקה. קראו מסמך נלווה שמפרט איך אנו סופרים שורות קוד.
  - a. בנוסף 5 נקודות אם הקוד יעמוד בכל הדרישות, ואורך הפונקציה המקסימלית >= 13.
  2. רוחב כל שורה (כולל הערות והזחות) לא יעלה על 75 תווים. ניתן לבדוק אורך של שורה בקודבלוקס (ראו תמונה מצורפת עם התרגיל). אם השורות ארוכות הן יגלשו בהדפסה, מה שיקשה על בדיקת התרגיל שלכם (ויגרור הורדת ניקוד).
  3. לפני כל פונקציה, יש לכתוב בהערה (בקצרה) מה הפונקציה עושה.
    - a. ההערה צריכה להסביר מה הפונקציה עושה ולא כיצד היא עושה זאת. לדוגמה:  
"הפונקציה מקדמת את num אם isMove חיובי" אינו הסבר, לעומת "אם בוצע מהלך חוקי, הפונקציה מקדמת את מספר המהלכים החוקיים".
    - b. ההערה צריכה להיות באנגלית. לצערנו עברית מודפסת כג'יבריש.
  4. חובה לתת שמות משמעותיים לפונקציות ולמשתנים.
    - a. השם צריך לשקף את פעולת הפונקציה או את מטרת המשתנה.
  5. חובה להשתמש בהזחות תקינות כפי שנלמדו בתרגולים.
  6. חובה להשתמש ב-define להגדרת קבועים בעלי משמעות (מספרים קבועים או תווים קבועים עם מטרה מוגדרת). שם הקבוע צריך להיות לפי המטרה שלו, ולא לפי התוכן. הימנעו מהגדרות כגון: #define ZERO 0, כיוון שהגדרה כזו לא מוסיפה מידע בקוד ומיותרת.
  7. אסור להשתמש במשתנים גלובאליים או סטאטיים.
  8. אסור לשכפל קוד שלא לצורך, למשל לכתוב שתי פונקציות שעושות פעולה דומה רק עם קבועים שונים.

## **בקרת עמיתים:**

בקורס מבוא למדעי המחשב תלמדו את הבסיס לתכנות. בסיס זה ישמש אתכם בהמשך התואר, וכן בעבודה בחברות הייטק. בעוד שבקורסים סטנדרטים במדעי המחשב המיקוד הוא על כתיבת קוד, בתעשייה תדרשו בעיקר "לתקן" קוד קיים – להוסיף יכולות או לתקן באגים קיימים. מעבר ליכולת לכתוב קוד, נדרש גם (ובעיקר) לקרוא קוד קיים, ולהבין אותו לעומק. בנוסף, מכיוון שמתכנתים מתעסקים בעיקר בקריאת קוד קיים, הכתיבה של קוד קריא וברור הפכה ליכולת חשובה מאוד.

על תרגיל זה תופעל סדנה של בקרת עמיתים. בסדנה זו כל סטודנט יתבקש להעריך קוד של 2 סטודנטים, והקוד שהגיש יוערך ע"י 2 סטודנטים אחרים. הסטודנטים יספקו לעמיתיהם משוב וציון על קוד שקראו, ויקבלו ציון ומשוב מהסטודנטים שהעריכו את הקוד שלהם. חלק מציון המטלה יורכב מממוצע הציונים אשר יינתנו ע"י סטודנטים עמיתים. הציון על התרגיל יורכב מהציון שיינתן על הקוד ומציון על איכות ההערכה שהסטודנט נתן. ההשתתפות בבקרת העמיתים אינה חובה, הציון שיינתן למשתתפים יהיה 2% מהציון הסופי בקורס.

**הגשה:** בנוסף להגשת התרגיל דרך אתר הקורס (באופן הרגיל) יש להגיש את התרגיל גם דרך מודל: <http://moodle.technion.ac.il>. למודל יש להגיש רק את הקובץ hw3q1.c. הקובץ חייב להיות אנונימי ולא להכיל את שמכם או פרט מזהה אחר. הסיסמא הראשונית למודל היא הת"ז שלכם.

**הערכה:** ההערכה תתחיל לאחר סיום תאריך ההגשה של התרגיל. ניתן למצוא טופס הערכה לדוגמה ביחד עם תרגיל זה.

לכל שאלה בקשר לבקרת עמיתים אתם מוזמנים לפנות ליעל ארז (המתרגלת האחראית) באימייל [yael.erez@gmail.com](mailto:yael.erez@gmail.com) או שעת קבלה.

## איקס-עיגול משודרג

בתרגיל זה נממש גרסה של המשחק המוכר איקס עיגול, בדומה למתואר בלינק:  
<https://mathwithbaddrawings.com/2013/06/16/ultimate-tic-tac-toe/>

במשחק זה, המיועד לשני שחקנים, לוח המשחק בגודל  $9 \times 9$ , ומורכב מ-9 תתי-לוחות רגילים בגודל  $3 \times 3$  של המשחק איקס עיגול. השחקן הראשון מסמן את מהלכיו בעזרת התו X, והשחקן השני בעזרת התו O. השחקנים ממקמים את התו שלהם לסירוגין, במקומות פנויים בלוח, עם מגבלה נוספת: **המיקום בתת-לוח ששחקן בוחר מגדיר את תת הלוח בו יצטרך לשחק השחקן הבא**. לדוגמא, אם השחקן הראשון מיקם תו בפניה השמאלית העליונה של אחד מתתי הלוחות, השחקן הבא יצטרך למקם תו בתת הלוח השמאלי העליון, אלא אם כן אין מקומות פנויים בתת לוח זה (במקרה כזה יוכל לבחור תת לוח כרצונו). כדי לנצח בתת-לוח מסוים, יש ליצור רצף ישר (מאונך, מאוזן או אלכסון) באורך 3. ניתן להמשיך לשחק בתת-לוח גם אחרי שאחד השחקנים ניצח בו (במקומות הפנויים), בשונה מהתיאור בלינק. כדי לנצח בכל המשחק, יש ליצור רצף ישר של 3 נצחונות בתתי-לוחות, כמודגם בלינק.

### **מצב התחלתי**

בתחילת המשחק הלוח ריק, והשחקן הראשון הוא בעל התו X.

### **מהלך סבב בודד**

בכל סבב יודפס הלוח הנוכחי והשחקן הנוכחי יידרש למקם תו על הלוח. במידה והשחקן הנוכחי יכול לבחור באיזה תת-לוח ברצונו לשחק (בסבב הראשון או במידה ותת הלוח שהוגדר על-ידי הצעד הקודם מלא), ראשית יידרש לבחור באיזה תת לוח ברצונו לשחק. לאחר שהוגדר תת הלוח (ע"י הצעד הקודם או ע"י השחקן), השחקן יידרש לבחור מיקום בתת הלוח הנבחר.

### **סיום המשחק**

המשחק מסתיים כאשר יש רצף ישר של שלושה נצחונות של אחד השחקנים בתתי לוחות, או כאשר הלוח מתמלא.

### **הנחות לגבי הקלט**

לא ניתן לבצע הנחות לבי הקלט. במידה ומצפים לקבל מהמשתמש מספר בתחום מסוים אך המספר חורג מהתחום, המשתמש יתבקש שוב להזין מספר בתחום הנדרש. במידה ומצפים לקבל מהמשתמש מספר אך הוזן תו שאינו מספר, יש לסיים את התכנית עם הודעת שגיאה. כל ההודעות מפורטות בדוגמא בהמשך.

## הדפסות במהלך המשחק

בתחילת המשחק תוצג הודעת פתיחה:

```
Welcome to Ultimate Tic-Tac-Toe!
```

לאחר מכן יודפס מצב הלוח, והודעה לשחקן הראשון עם בקשה לבחור תת-לוח בו הוא ירצה למקם את התו X:

```
1 1 1 2 2 2 3 3 3
1   |   |   |3
1   |   |   |3
1   |   |   |3
-----
4   |   |   |6
4   |   |   |6
4   |   |   |6
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
7 7 7 8 8 8 9 9 9
```

Player X, please enter the index of the wanted sub-board:

השחקן יזין מספר בין 1 ל-9, אינדקס תת-הלוח בו הוא ירצה למקם תו. האינדקסים של תתי הלוחות מתחילים מהפינה השמאלית העליונה ומסודרים לפי שורות, ומודפסים מסביב ללוח עבור נוחות המשחק. לאחר מכן יתבקש השחקן לבחור אינדקס בתת הלוח שבחר בו ירצה למקם את התו X. לדוגמא, אם השחקן בחר בתת-לוח מספר 1:

```
Player X, please enter the wanted position in sub-board 1:
```

במידה והשחקן הקליד אינדקס לא תקין (מחוץ לטווח), של תת-לוח או בתוך תת-לוח מסוים, תודפס ההודעה:

```
Index out of range. Please enter a value between 1 and 9:
```

במידה והשחקן בחר בעצמו בתת לוח מלא, נניח תת-לוח מספר 6, תודפס ההודעה:

```
Sub-board 6 is full. Please choose another one:
```

במידה והשחקן בחר אינדקס תפוס בתוך תת לוח מסוים (כאשר יש מקומות פנויים אחרים באותו תת-לוח), תודפס ההודעה:

Position is taken, please choose another one:

במידה והשחקן הזין תו שאינו ספרה או רווחים, תודפס הודעת שגיאה (ובמקרה זה התכנית תסתיים):

Input error!

לאחר שהשחקן בחר מיקום תקין, יודפס הלוח החדש, והודעה לשחקן O שמבקשת להזין מיקום בתת הלוח בו עליו לשחק (כפי שנקבע לפי התור הקודם):

```
1 1 1 2 2 2 3 3 3
1  X  |   |   |3
1    |   |   |3
1    |   |   |3
-----
4    |   |   |6
4    |   |   |6
4    |   |   |6
-----
7    |   |   |9
7    |   |   |9
7    |   |   |9
7 7 7 8 8 8 9 9 9
```

Player O, please enter the wanted position in sub-board 2:

וכך הלאה. כאשר שחקן יצר רצף ישר באורך 3 באחד מתתי הלוחות, תודפס ההודעה:

Player X has won in sub-board 6!

כאשר שחקן יצר רצף ישר של שלושה נצחונות בתתי לוחות, תודפס ההודעה:

Game over, X is the winner!

והמשחק יסתיים. במידה ולא נשארו מקומות פנויים בלוח המשחק אך אף שחקן לא ניצח, תודפס ההודעה:

Game over, it's a tie!

והמשחק יסתיים.

## מקרה הרצה לדוגמה (קלט מסומן באדום):

Welcome to Ultimate Tic-Tac-Toe!

```
1 1 1 2 2 2 3 3 3
1   |   |   |3
1   |   |   |3
1   |   |   |3
```

```
-----
4   |   |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
```

```
7 7 7 8 8 8 9 9 9
```

Player X, please enter the index of the wanted sub-board:

1

Player X, please enter the wanted position in sub-board 1:

2

```
1 1 1 2 2 2 3 3 3
1  X |   |   |3
1   |   |   |3
1   |   |   |3
```

```
-----
4   |   |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
```

```
7 7 7 8 8 8 9 9 9
```

Player O, please enter the wanted position in sub-board 2:

-1

Index out of range. Please enter a value between 1 and 9:

5

```
1 1 1 2 2 2 3 3 3
1  X |   |   |3
1   |   O |   |3
1   |   |   |3
```

```
-----
4   |   |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
```

```
7 7 7 8 8 8 9 9 9
```

Player X, please enter the wanted position in sub-board 5:

1

```
1 1 1 2 2 2 3 3 3
1 X |   |   |3
1   |   O |   |3
1   |   |   |3
```

```
-----
4   |X   |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
7 7 7 8 8 8 9 9 9
```

Player O, please enter the wanted position in sub-board 1:

5

```
1 1 1 2 2 2 3 3 3
1 X |   |   |3
1 O |   O |   |3
1   |   |   |3
```

```
-----
4   |X   |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
7 7 7 8 8 8 9 9 9
```

Player X, please enter the wanted position in sub-board 5:

2

```
1 1 1 2 2 2 3 3 3
1 X |   |   |3
1 O |   O |   |3
1   |   |   |3
```

```
-----
4   |X X |   |6
4   |   |   |6
4   |   |   |6
```

```
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
7 7 7 8 8 8 9 9 9
```

Player O, please enter the wanted position in sub-board 2:

3



1	1	1	2	2	2	3	3	3
1	X				O			3
1	O			O				3
1								3

4		X	X				6
4							6
4							6

7							9	
7							9	
7							9	
7	7	7	8	8	8	9	9	9

Player X, please enter the wanted position in sub-board 3:

6

1	1	1	2	2	2	3	3	3
1	X				O			3
1	O			O			X	3
1								3

4		X	X				6
4							6
4							6

7							9	
7							9	
7							9	
7	7	7	8	8	8	9	9	9

Player O, please enter the wanted position in sub-board 6:

5

1	1	1	2	2	2	3	3	3
1	X				O			3
1	O			O			X	3
1								3

4		X	X				6
4					O		6
4							6

7							9	
7							9	
7							9	
7	7	7	8	8	8	9	9	9

Player X, please enter the wanted position in sub-board 5:

3

```

Player X has won in sub-board 5!
1 1 1 2 2 2 3 3 3
1 X |   O |   |3
1 O |   O |   X|3
1   |   |   |3
-----
4   |X X X|   |6
4   |   |   O |6
4   |   |   |6
-----
7   |   |   |9
7   |   |   |9
7   |   |   |9
7 7 7 8 8 8 9 9 9
Player O, please enter the wanted position in sub-board 3:
a
Input error!

```

## דגשים נוספים:

- עליכם לוודא את נכונות הקוד שלכם באמצעות אתר הבדיקה בכתובת <http://csl2.cs.technion.ac.il/~cs234114> כפי שנדרשתם בתרגילים הקודמים. גם הפעם מסופקים לכם באתר הקורס קבצי הקלט והפלט המצופה לטסטים שבאתר הבדיקה.
- בשונה מהתרגילים הקודמים, הפעם תוכלו למצוא באתר גם "תוכנית לדוגמא" (קובץ הרצה). הפלט שהתוכנית מוציאה הוא הפלט המצופה, ולכן תוכלו להיעזר בה כדי לבדוק את עצמכם על מקרים נוספים מעבר לבדיקות המפורסמות באתר. לדוגמא: תוכלו לכתוב קובץ קלט משלכם, להעביר אותו לתוכנית באמצעות redirection ולהשוות את הפלט המתקבל לפלט המתקבל מהתוכנית שלכם. כמו כן, בכל במקרה שההתנהגות הרצויה אינה ברורה, עליכם לבדוק מה התוכנית לדוגמא עושה.
- עליכם לשים לב להדפיס את הפלט כפי שהוא מופיע במדויק בטסטים ובתוכנית לדוגמא, בדגש על: רווחים (בפרט בתחילת שורה או בסופה), מעברי שורה (בפרט בסוף הקובץ) ואותיות גדולות וקטנות.

**בהצלחה !**