

פרויקט מסכם תכנות מתקדם – סמסטר ב' תשפ"ג

הנחיות כלליות

יש לתעד את התכניות.
יש להקפיד על יעילות וחסכון בזמן ריצה וזיכרון.
יש להקפיד שגודל פונקציה לא יחרוג ממסך אחד.
יש לפנות זיכרון שהוקצה דינמית ואשר אין בו צורך יותר.
יש לבדוק אם הקצאות הזיכרון הצליחו.
יש לבדוק שפתיחת קובץ הצליחה.
יש לסגור קובץ לאחר סיום השימוש בו.
יש להשתמש ב-`#define` היכן שנחוץ.
אין לחרוג מה-`prototype`-ים שהוגדרו אך ניתן ליצור פונקציות עזר לפי הצורך.
את הפרויקט יש להגיש בזוגות (בני הזוג יכולים להשתייך לקבוצות שונות בקורס)

הגדרת הפרויקט

מטרת הפרויקט היא לתכנת גרסה פשוטה של משחק הדמקה.

בלוח דמקה יש שמונה שורות ושמונה עמודות של משבצות. מיקום משבצת מתואר ע"י מערך בן שני תאים – אות לשורה (A עד H) ומספר לעמודה (1 עד 8). לדוגמא: המיקום של המשבצת שמכילה X בלוח הבא מסומן ע"י B3 ומיקום המשבצת שמכילה Y הוא F1.

	1	2	3	4	5	6	7	8
A								
B			X					
C								
D								
E								
F	Y							
G								
H								

חוקי המשחק

למשחק שני שחקנים שנקראים T ו-B. המשחק מתנהל רק על המשבצות האפורות כאשר T נע מלמעלה כלפי מטה ו-B נע מלמטה כלפי מעלה. לכל שחקן בתחילת המשחק יש 12 כלי משחק אשר מסודרים כדלקמן:

	1	2	3	4	5	6	7	8
A		T		T		T		T
B	T		T		T		T	
C		T		T		T		T
D								
E								
F	B		B		B		B	
G		B		B		B		B

H	B		B		B		B	
---	---	--	---	--	---	--	---	--

כל שחקן מתקדם בתורו.

בכל תור השחקן מזיז את אחד מכלי המשחק שלו. כלי משחק יכול לנוע לאחת משתי המשבצות אשר בכיוון תנועתו במידה והיא פנויה. במקרה שהמשבצת אינה פנויה ומכילה כלי משחק של היריב אבל המשבצת שאחריה (בכיוון תנועת השחקן) פנויה, יכול כלי המשחק לדלג מעל כלי המשחק של היריב ובכך להוציא את כלי משחק זה מהמשחק. מהלך זה קרוי capture או אכילה. במידה ומצב זה מתקיים גם במשבצת שהגיע אליה, יכול כלי המשחק להמשיך ולדלג ולהוציא עוד כלי משחק של יריבו מהמשחק.

לדוגמא, במקרה שתורו של השחקן T ומצב הלוח הוא:

	1	2	3	4	5	6	7	8
A				T				
B			B		B			
C		*				*		
D					B		B	
E				*				*
F								
G								
H								

יכול כלי המשחק שבמשבצת A4 לנוע אל אחת מהמשבצות שמסומנות ב* כאשר כלי משחק שהוא מדלג מעליהם יוצאים מהמשחק. במקרה ויש מספר אפשרויות תנועה, **חייב** השחקן לבחור בזו עם מספר הדילוגים הגדול ביותר.

שימו לב: יתכן וישנן מספר אפשרויות עם מספר הדילוגים הגדול ביותר. מצב כזה נובע כאשר יש **לפחות משבצת אחת** שבה ניתן לנוע שמאלה או ימינה. במקרה כזה, **עבור המשבצת** (וכל משבצת בהמשך המסלול שעבורה יש יותר מאפשרות אחת), יש לבחור כך:

עבור T יש לבחור בכיוון המשבצת שמספרה (1-8) הוא הגדול ביותר.

עבור B יש לבחור בכיוון המשבצת שמספרה (1-8) הוא הקטן ביותר.

בדוגמא לעיל, חייב השחקן T לנוע אל המשבצת E8 (במשבצת הראשונה הוא יבחר בC6 ומ-C6 הוא יבחר ב-E8 – בשני מקרים אלו על-פי הכלל נבחרו עבור T המשבצות עם מספר העמודה בגדול ביותר) ומצב הלוח לאחר המהלך יהיה:

	1	2	3	4	5	6	7	8
A								
B			B					
C								
D					B			
E								T
F								
G								
H								

למצב בו אין דילוגים באף אחד מהמהלכים האפשריים, כמו בתחילת המשחק, יש להתייחס כ**אפס** דילוגים ולהפעיל את הכלל לעיל. דוגמא לכך ישנה בסוף המסמך.

שימו לב שאפשרויות התנועה של שחקן יוצרות עץ בינארי.

המשחק מסתיים באחד מהמקרים הבאים:

- א. כאשר אחד השחקנים נותר ללא כלי משחק
- ב. כאשר אחד מכלי המשחק של T מגיע אל שורה H
- ג. כאשר אחד מכלי המשחק של B מגיע אל שורה A.

מימוש הפרויקט

על-מנת לייצג משבצת בלוח דמקה, עושים שימוש בהגדרה הבאה:

```
typedef struct _checkersPos
{
    char row, col;
} checkersPos;
```

על-מנת לייצג לוח עושים שימוש בהגדרה הבאה:

```
typedef unsigned char Board[BOARD_SIZE][BOARD_SIZE];
```

על-מנת לייצג שחקן הוגדר הטיפוס

```
typedef unsigned char Player;
```

סעיף 1

כדי לתאר את העץ הבינארי של אפשרויות התנועה של כלי משחק אחד עושים שימוש בהגדרות הבאות:

```
typedef struct _SingleSourceMovesTreeNode{
    Board          board;
    checkersPos    *pos;
    unsigned short  total_captures_so_far; // מספר הדילוגים עד כה
    struct _SingleSourceMovesTreeNode *next_move[2]; // יעדי התנועה
} SingleSourceMovesTreeNode;
```

```
typedef struct _SingleSourceMovesTree {
    SingleSourceMovesTreeNode *source;
} SingleSourceMovesTree;
```

יש לכתוב את הפונקציה:

SingleSourceMovesTree *FindSingleSourceMoves(Board board, checkersPos *src)

הפונקציה מקבלת לוח משחק עם מצב נתון ומשבצת בלוח.

אם קיים כלי משחק במשבצת, הפונקציה מחזירה עץ בינארי של כל אפשרויות התנועה שלו. אחרת, מחזירה הפונקציה NULL.

שימו לב: יש לכלול את כל המסלולים ואין להפעיל את כלל הבחירה שתואר בעמוד 2. כלל זה יופעל בהמשך.

סעיף 2

כדי לתאר את התנועה שנבחרה מבין כל אפשרויות התנועה של כלי משחק אחד עושים שימוש בהגדרות הבאות:

```
typedef struct _SingleSourceMovesListCell {
    checkersPos          *position;
    unsigned short        captures;
```

```

    struct _SingleSourceMovesListCell    *next;
} SingleSourceMovesListCell;

```

```

typedef struct _SingleSourceMovesList {
    SingleSourceMovesListCell *head;
    SingleSourceMovesListCell *tail;
} SingleSourceMovesList;

```

יש לכתוב את הפונקציה:

**SingleSourceMovesList *FindSingleSourceOptimalMove(
SingleSourceMovesTree *moves_tree)**

הפונקציה מקבלת עץ של כלי משחק מסוים ומחזירה את מסלול התנועה עם מספר הדילוגים (captures) הרב יותר מבין כל מסלולי התנועה האפשריים שמתוארים בעץ על-פי כלל הבחירה שתואר לעיל.

סעיף 3

כדי לתאר את כל אפשרויות התנועה של שחקן (B או T), עושים שימוש ברשימה מקושרת אשר מכילה עבור כל אחד מכלי המשחק של אותו שחקן את מסלול התנועה הטוב ביותר מהמשבצת שכלי המשחק נמצא בה:

```

typedef struct _multipleSourceMovesListCell {
    SingleSourceMovesList    *single_source_moves_list;
    struct _multipleSourceMovesListCell *next;
} MultipleSourceMovesListCell;

```

```

typedef struct _multipleSourceMovesList {
    MultipleSourceMovesListCell *head;
    MultipleSourceMovesListCell *tail;
} MultipleSourceMovesList;

```

כתבו את הפונקציה:

MultipleSourceMovesList *FindAllPossiblePlayerMoves(Board board, Player player)

הפונקציה מקבלת לוח משחק ושחקן ומחזירה את רשימת המהלכים הטובים ביותר מכל משבצת בלוח אשר מכילה כלי משחק של שחקן זה ואשר ניתן לנוע ממנה.

סעיף 4

ממשו את הפונקציה:

void Turn(Board board, Player player)

הפונקציה מקבלת לוח משחק ושחקן ומבצעת את המהלך הטוב ביותר מבין המהלכים האפשריים בלוח הנתון עבור שחקן זה. במידה והמהלך הנבחר מכיל דילוגים, יש להסיר מלוח המשחק את כלי המשחק של היריב שמדלגים מעליהם. במקרה שיש מספר משבצות אשר המסלולים האופטימליים זהים מבחינת מספר הדילוגים, יש לבחור את המסלול באופן הבא: עבור השחקן B יש לבחור את המסלול שמתחיל במשבצת עם מספר העמודה והשורה הקטנים ביותר (שורה A קטנה משורה B לצורך זה). יש קודם להשוות את השורה ואח"כ את העמודה (לדוגמא $A3 < A4$ ו- $B1 > A5$). עבור השחקן T, יש לבחור את המסלול שמתחיל במשבצת עם מספר העמודה והשורה הגדולים ביותר.

סעיף 5

ממשו את הפונקציה

void PlayGame(Board board, Player starting_player)

הפונקציה מקבלת לוח ושחקן שמתחיל את המשחק.

הפונקציה משחקת משחק שבו המחשב מחליט על מהלכי שני הצדדים (המחשב משחק מול עצמו) כאשר הוא מתחיל מהשחקן שהתקבל בפרמטר `starting_player`. הפונקציה תדפיס את הלוח שהתקבל ובכל תור הפונקציה תדפיס את השחקן שזה תורו, את המהלך שהתבצע עבורו ואת הלוח לאחר המהלך. בסיום המשחק יש להדפיס את הפרטים הבאים: מי המנצח, כמה דילוגים ביצע המנצח, מה מספר הדילוגים הגדול ביותר שנעשה וע"י איזה שחקן. על הפלט להיות בפורמט הבא (להלן דוגמא לפלט – T מתחיל):

```

+---+---+---+---+---+---+
+ |1|2|3|4|5|6|7|8|
+---+---+---+---+---+---+
|A| |T| |T| |T| |T|
+---+---+---+---+---+---+
|B|T| |T| |T| |T| |
+---+---+---+---+---+---+
|C| |T| |T| |T| |T|
+---+---+---+---+---+---+
|D| | | | | | | |
+---+---+---+---+---+---+
|E| | | | | | | |
+---+---+---+---+---+---+
|F|B| |B| |B| |B| |
+---+---+---+---+---+---+
|G| |B| |B| |B| |B|
+---+---+---+---+---+---+
|H|B| |B| |B| |B| |
+---+---+---+---+---+---+

```

T's turn:

C8->D7

```

+---+---+---+---+---+---+
+ |1|2|3|4|5|6|7|8|
+---+---+---+---+---+---+
|A| |T| |T| |T| |T|
+---+---+---+---+---+---+
|B|T| |T| |T| |T| |
+---+---+---+---+---+---+
|C| |T| |T| |T| | |
+---+---+---+---+---+---+
|D| | | | | | |T| |
+---+---+---+---+---+---+
|E| | | | | | | |
+---+---+---+---+---+---+
|F|B| |B| |B| |B| |
+---+---+---+---+---+---+
|G| |B| |B| |B| |B|
+---+---+---+---+---+---+
|H|B| |B| |B| |B| |
+---+---+---+---+---+---+

```

B's turn:

F1->E2

```

+---+---+---+---+---+---+
+ |1|2|3|4|5|6|7|8|
+---+---+---+---+---+---+
|A| |T| |T| |T| |T|
+---+---+---+---+---+---+
|B|T| |T| |T| |T| |
+---+---+---+---+---+---+
|C| |T| |T| |T| | |
+---+---+---+---+---+---+
|D| | | | | | |T| |
+---+---+---+---+---+---+
|E| |B| | | | | | |
+---+---+---+---+---+---+
|F| | |B| |B| |B| |
+---+---+---+---+---+---+
|G| |B| |B| |B| |B|
+---+---+---+---+---+---+
|H|B| |B| |B| |B| |
+---+---+---+---+---+---+

```

.
.
.
.

T wins!

T performed 22 moves.

B performed the highest number of captures in a single move - 4

שימו לב בתחילת המשחק מספר הדילוגים של כל השחקנים שמהם מתאפשר מהלך הוא אפס ולכן יש להפעיל את החוק מעמוד 2 לכל משבצת עבור אפס דילוגים. אח"כ, יש להפעיל את הכלל מסעיף 4 כדי לבחור את המהלך מבין המשבצות השונות. בפרט:

עבור T יש את האפשרויות הבאות:

- ממשבצת C2 ניתן לזוז אל D1 או אל D3. על פי כלל הבחירה מעמוד 2 ייבחר D3. באופן דומה, עבור C4, C6, C8 ייבחרו בהתאמה D5, D7, D7. על-פי כלל הבחירה מסעיף 4, ייבחר המהלך מ-C8 אל D7.

בתור הבא, עבור B יש את האפשרויות הבאות:

- ממשבצת F1 ניתן לזוז אל E2. ממשבצת F3 ניתן לזוז אל E4 או E2 - על פי כלל הבחירה מעמוד 2 ייבחר E2. באופן דומה, עבור F5, F7 ייבחרו בהתאמה E4, E6. על פי כלל הבחירה מסעיף 4, ייבחר המהלך מ-F1 אל E2.

וכן הלאה.