### **תרגיל בית 3: משחק זיכרון (Memory Game)**

#### **נושאים: מבנים, הקצאות דינמיות ומצביעים בשפת C**

### **תיאור כללי:**

בתרגיל זה תכתבו מימוש של **משחק זיכרון** (Memory Game) קלאסי. המשחק יופעל על לוח בגודל N x N כאשר N בהכרח זוגי. לוח המשחק יכיל מספר זוגות של קלפים, ועל השחקן לחשוף זוגות קלפים תואמים. בכל תור, השחקן יבחר שני קלפים, ואם הקלפים תואמים – הם יישארו גלויים. אם לא, הם יחזרו להיות מוסתרים.

המשחק יסתיים כאשר כל זוגות הקלפים התואמים נחשפים, והמשתמש יזכה.

### **דרישות התרגיל:**

לוח המשחק יהיה מיוצג באמצעות **struct GameBoard** שיוגדר כך:

// Structure to represent the game board

typedef struct {

int\*\* values; // Stores the card values

int\*\* revealed; // Stores the state of each card (0 = hidden, 1 = revealed)

int size; // board size is size \* size

1. **מבנה המשחק (Game Board)**:
   1. הלוח יהיה בגודל N x N
   2. כל תא בלוח המשחק יכיל קלף שמיוצג על ידי ערך מספרי, וכן מידע האם הקלף נחשף או עדיין מוסתר
2. **ניהול המשחק**:
   1. כל הקלפים יתחילו כשהם מוסתרים (לא גלויים לשחקן)
   2. השחקן יבחר שני קלפים על ידי ציון המיקום שלהם בלוח (שורה ועמודה). אם הקלפים תואמים, הם יישארו גלויים; אם לא, הם יוחזרו למצב מוסתר.
   3. השחקן יוכל להמשיך לשחק עד שכל הקלפים נחשפו.
   4. לאחר כל תור, הלוח יודפס מחדש כך שהשחקן יוכל לראות את מצב הלוח הנוכחי. בנוסף תהיה בדיקת ניצחון.
   5. ברגע שכל הקלפים נחשפו, המשחק יסתיים בהודעת ניצחון: Congratulations! You've won the game!
   6. בכל שלב המשתמש יכול להקליד -1 כדי לבחור לסיים את המשחק. במקרה זה יש לשחרר זיכרון ולהדפיס הודעה: Game Over! Better luck next time!
3. **שחרור זיכרון**:
   1. בסיום המשחק, יש לשחרר את כל הזיכרון שהוקצה דינמית עבור לוח המשחק.

### **מבנה התוכנית:**

1. **אתחול הלוח**:
   1. יש לכתוב פונקציה initBoard אשר מקצה זיכרון דינמית עבור לוח המשחק בגודל N x N ומאתחלת אותו בערכים. הפונקציה תקבל את גודל הלוח וכן מערך סטטי חד מימדי של N x N מספרים. הפונקציה תקצה דינמית ותאתחל בעזרתם את המערך values השייך למבנה GameBoard. יש להקצות ולאתחל גם את המערך revealed.
2. **הדפסת הלוח**:
   1. כתבו פונקציה printBoard אשר מדפיסה את מצב הלוח הנוכחי. קלפים מוסתרים יוצגו כ-\*, וקלפים שנחשפו יוצגו בערכם.
3. **תור משחק**:
   1. כתבו פונקציה playTurn אשר מקבלת שני מיקומים בלוח, חושפת את הקלפים שבמיקומים אלו, ומשווה ביניהם. אם הקלפים תואמים, הם יישארו גלויים. אם לא, הם יוסתרו שוב לאחר הדפסת הלוח.
4. **בדיקת ניצחון**:
   1. כתבו פונקציה checkVictory אשר בודקת האם כל הקלפים נחשפו. אם כן, הפונקציה תחזיר ערך 1, אחרת 0.
5. **שחרור זיכרון**:
   1. כתבו פונקציה freeBoard אשר תשחרר את כל המשאבים הדינמיים שהוקצו במהלך המשחק (זיכרון שהוקצה ללוח ולקלפים).

### **פונקציות עיקריות:**

* GameBoard\* initBoard(const int\* predefinedValues, int size);

הפונקציה מקבלת את גודל המערך ומאתחלת בעזרתו את size השייך למבנה. בנוסף, פונקציה מקצה זיכרון עבור הלוח בגודל נתון ומאתחלת אותו באמצעות מערך מספר שהתקבל כפרמטר

אם הפונקציה קיבלה לדוגמה את המערך הבא:

int values[] = {1,2,3,4,5,6,7,8,8,1,2,3,4,5,6,7};

היא תיצור לוח משחק שמסודר כך (בתחילת המשחק הקלפים כמובן מוסתרים):

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 8 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |

* **void printBoard(GameBoard\* gb)**

פונקציה שמדפיסה את הלוח, כאשר קלפים מוסתרים מוצגים כ-\*, וקלפים שנחשפו מוצגים בערכם. יש להדפיס 2 רווחים בין כל 2 ערכים בשורה.

* **int playTurn(GameBoard\* gb, int row1, int col1, int row2, int col2)**

פונקציה שמקבלת שני מיקומים בלוח, חושפת את הקלפים, ומשווה ביניהם.

במקרה של מיקום שחורג מגודל הלוח או של כרטיס שכבר גלוי, הפונקציה תדפיס Invalid location ותחזיר 1-.

אם התגלו קלפים תואמים הפונקציה תחזיר 1.

אם התגלו קלפים לא תואמים הפונקציה תחזיר 0.

* **int checkVictory(GameBoard\* gb)**

פונקציה שבודקת האם כל הקלפים נחשפו ומחזירה ערך 1 במקרה של ניצחון או 0 אם המשחק נמשך.

* **void freeBoard(GameBoard\* gb)**

פונקציה שמשחררת את כל המשאבים הדינמיים שהוקצו ללוח המשחק.

### **דוגמת ריצה:**

\* \* \* \*   
\* \* \* \*   
\* \* \* \*   
\* \* \* \*   
  
Choose the first location (row column): 0 0

Choose the second location (row column): 1 1

2 \* \* \*   
\* 2 \* \*   
\* \* \* \*   
\* \* \* \*

Match found! Continue playing...

2 \* \* \*   
\* 2 \* \*   
\* \* \* \*   
\* \* \* \*   
  
Choose the first location (row column): 0 0

Invalid location

Choose the first location (row column): 0 1

Choose the second location (row column): 2 2

2 5 \* \*   
\* 2 \* \*   
\* \* 7 \*   
\* \* \* \*

No match found. Continue playing...

2 \* \* \*   
\* 2 \* \*   
\* \* \* \*   
\* \* \* \*   
  
Choose the first location (row column):

### **הנחיות להגשה:**

1. ניתן להוסיף פונקציות עזר בתכנית ותכונות לstruct לפי הצורך, אך אין לשנות את חתימות הפונקציות המוגדרות.
2. יש לשמור על מבנה קוד מסודר וקריא.
3. יש להגיש את הקוד בקובץ C יחיד, בשם ex3.c
4. יש להגיש את קוד 5 הפונקציות בלבד ללא main. מותר להוסיף פונקציות נוספות לפי הצורך ולהגיש גם אותן.
5. יש לוודא שהתכנית עוברת את כל הPre-Submission test במערכת הvpl טרם ההגשה. תרגיל שיכשל ב Pre-Submission testיקבל אוטומטית ציון 0.