

## חומרים שהוכנו על-ידי

## משתתפי קורס מורים מובילים תשע"ה

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

כתיבה ועריכה:

ענת שלוס-סגל

## דפים מדורגים בנושא רקורסיה

# 8 מלכות



### דף מספר 1

#### בעיית שמונה המלכות

מלכה (Queen) במשחק שח-מט יכולה לנועה בשורה, בטור או באלכסון יחסית למקום שהיא עומדת בו. המלכה יכולה לנוע מספר בלתי מוגבל של משבצות בכיוון הנבחר, כל עוד כלי אחר אינו חוסם את דרכה, וכן כל עוד לא יצאה מהלוח. מלכה יכולה להכות כל כלי העומד בדרכה אך אחרי שהכתה כלי (כלומר הוציאה אותו מהמשחק ונעמדה במקומו) היא אינה יכולה לנוע יותר עד לתור הבא. מלכה מאיימת על כל הכלים הנמצאים בשורה, בעמודה או באלכסון של המשבצת בה היא ניצבת (כלומר יכולה להוציאם בתורה).

כתבו פעולה רקורסיבית שתציב שמונה מלכות על לוח שחמט ריק (מערך דו-ממדי בגודל  $8 \times 8$ ), כך שאף אחת לא תאיים על האחרות.

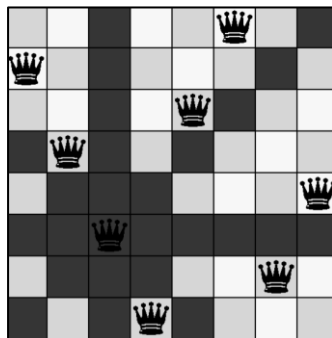
(מתוך הספר עיצוב תוכנה מבוסס עצמים פרק 4 – שאלה 31)

### דף מספר 2

כל מלכה תמוקם בשורה נפרדת, הבעיה הנותרת היא למצוא עבור כל מלכה את העמודה הנכונה. כדי לדמות לוח נעשה שימוש במערך דו-ממדי של ערכים בוליאניים. בכל משבצת בה תמוקם מלכה יושם הערך true.

מקמו את המלכה הראשונה על הלוח במקום הפנוי הראשון.

המשיכו בהצבת שאר המלכות בשורות הבאות כך שאף אחת לא תאיים על האחרות.



### דף מספר 3

לפניכם שלד הפעולה **PlaceQueens** המציבה את המלכות על הלוח:

```
static bool PlaceQueens(bool[,] board, int row)
{
    ...
    board[row, ?] = true;
    ...
}
```

הפעולה תקבל את מספר השורה של המלכה אותה אנו רוצים למקם כעת על הלוח. הפעולה תחזיר true במידה וההצבה הצליחה, אחרת במידה ואין פתרון לבעיה הפעולה תחזיר false.

נתחיל בהצבת המלכה הראשונה בשורה 0:

```
static void Main(string[] args)
{
    bool[,] board = new bool[8, 8];
    if (PlaceQueens(board, 0))
        // יש פתרון
}
```

המשיכו בהצבת שאר המלכות על הלוח באופן רקורסיבי.

### דף מספר 4

מה קורה כאשר המיקום של המלכה בעמודה אינו בטוח (מלכות אחרות מאיימות עליה)? כיצד נמצא את העמודה הנכונה?

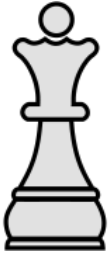
מה קורה אם לא ניתן למקם את המלכה באף אחת מהעמודות בשורה הנתונה?

אולי המיקום שבחרנו לאחת מהמלכות הקודמות שגוי?

צפו בסרטון הבאה והמשיכו במשימה:

<https://www.youtube.com/watch?v=ZM7RGa8W2KA>





## דף מספר 5

מה קורה כאשר המיקום של המלכה בעמודה אינו בטוח?  
נעבור לעמודה הבאה. 💡

\* איך בודקים אם המיקום בטוח? נשאר זאת להמשך

מה קורה אם לא ניתן למקם את המלכה באף אחת מהעמודות בשורה?  
אין פתרון, עלינו להחזיר false. 💡

אולי המיקום שבחרנו לאחת מהמלכות הקודמות שגוי?  
נחזור אחורה (backtracking) ונחפש עברה מיקום חדש. 💡

לפניכם הפעולה **PlaceQueens** השלימו את הוראות ה return בהתאם:

```
static bool PlaceQueens(bool[,] board, int row)
{
    if ( ... )
        return ...;

    for (int col = 0; col < N; col++)
    {
        if ( האם המיקום בעמודה בטוח ? )
        {
            board[row, col] = true;
            if (PlaceQueens(board, row + 1))
                return ...;
            board[row, col] = false;
        }
    }
    return ...;
}
```

שימו ♥ - לא לשכוח את תנאי העצירה!

## דף מספר 6

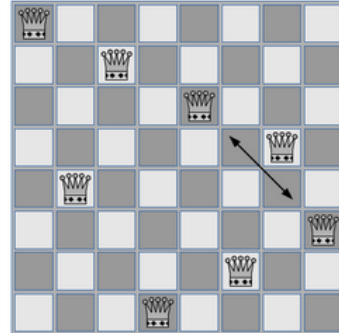
כיצד מוודאים שאף מלכה לא מאוימת ע"י האחרות?

כתבו פעולה המקבלת את לוח המשחק ומיקום של מלכה ובודקת שהיא אינה מאוימת ע"י המלכות שכבר מוצבות על הלוח.

## דף מספר 7

לפניכם מימוש חלקי של הפעולה הבודקת שהמלכה אינה מאוימת ע"י המלכות הקודמות:

```
static bool Safe(bool[,] board, int row, int col)
{
    for (int i = 0; i <= row; i++)
    {
        if (board[i, col])
            return false;
        if (. . .)
            return false;
        if (. . .)
            return false;
    }
    return true;
}
```



השלימו את הפעולה (זכרו לבדוק גם את האלכסונים).

## דף מספר 8

לפניכם הפעולה המלאה שבודקת שמיקום המלכה אינו מאויים ע"י המלכות הקיימות:

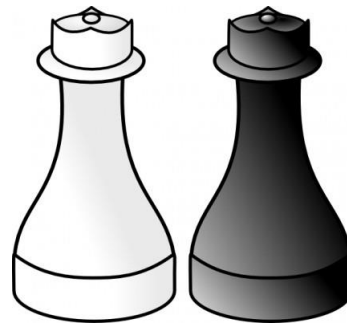
```
static bool Safe(bool[,] board, int row, int col)
{
    for (int i = 0; i <= row; i++)
    {
        if (board[i, col]) // אותה עמודה
            return false;
        if (i <= col && board[row - i, col - i]) // אלכסון ראשי
            return false;
        if (col + i < N && board[row - i, col + i]) // אלכסון משני
            return false;
    }
    return true;
}
```

שבצו אותה במקום המתאים בפעולה `PlaceQueens`.

## דף מספר 9

לפניכם הפתרון המלא לבעיה:

```
static bool PlaceQueens(bool[,] board, int row)
{
    if (row == N)
        return true;
    for (int col = 0; col < N; col++)
    {
        if (Safe(board, row, col))
        {
            board[row, col] = true;
            if (PlaceQueens(board, row + 1))
                return true;
            board[row, col] = false;
        }
    }
    return false;
}
```



### משימות נוספות:

1. הריצו את התכנית עבור גדלים שונים של לוחות.

2. הוסיפו הדפסה של הלוח עם הפתרון.