**תרגול מסכם - שקי חול**

**הקדמה**

הגעתם לתרגיל המסכם של שקי החול של קורס תכנות, ברכות!

את התרגיל יש לבצע ברצינות ובצורה מעמיקה, לאחר שביצעתם את כל התרגולים של הנושאים הקודמים והבנתם אותם לעומק.

יש לקרוא את הקובץ עד סופו **לפני** תחילת כתיבת הקוד!

**המשימה**

כתוב תכנית אשר תנהל משחק בין שני שחקנים.  
המשחק מתנהל על עולם משבצות דו-מימדי בגודל 30X30.

לכל שחקן יש מיקום בעולם זה; כמו-כן, בעולם קיים "שטיח המנצחים" - מלבן הנמצא בתוך העולם.  
 כל שחקן בתורו מתקדם בכיוון ע"פ בחירתו (בתוך גבולות העולם), עד אשר אחד השחקנים מגיע ל"שטיח המנצחים" – ברגע הגעתו הוא המנצח במשחק.

בתחילת המשחק יש לקלוט את המיקום של כל אחד מהשחקנים, ולאחר מכן את מיקום הפינה השמאלית העליונה של שטיח המנצחים, ואת אורכו ורוחבו של השטיח (גודל האורך בשטיח חייב להיות כגודל הרוחב, לכן ניתן לקלוט פרמטר אחד).  
  
במידה וקלט כלשהו לא תקין (אחד השחקנים מחוץ לתחום העולם / אחת ממשבצות השטיח מחוץ לתחום העולם) יש להודיע הודעה מתאימה ולקלוט את נתוני העצם האחרון שוב (כלומר מיקום השחקן הראשון / מיקום השחקן השני / מיקום וגודל השטיח).

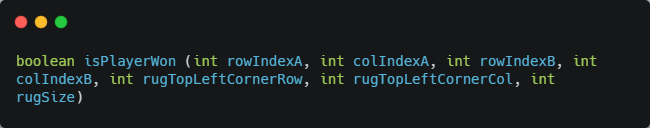
לאחר מכן, יש לנהל את התורות בין שני השחקנים, כאשר לאחר כל תור התכנית תדפיס את מפת העולם; משבצת ריקה תסומן ע"י תו רווח, משבצת השייכת ל"שטיח המנצחים" תסומן ע"י התו "\*" (כוכבית), ושחקן 1 ושחקן 2 יסומנו ע"י התווים "1" ו – "2" בהתאמה. הדפסת העולם תכלול גבול בעובי משבצת אחת מסביב לעולם אשר יסומן ע"י התו "#" (סולמית).   
כל שחקן רשאי לנוע לאחד מארבעת הכיוונים בעזרת הפקודות המספריות הבאות: 1 – משבצת למעלה; 2 – משבצת למטה; 3 – משבצת ימינה; 4 – משבצת שמאלה.

במידה ושחקן הקליד פקודה שגויה, יש לציין זאת בהודעה מתאימה, אולם השחקן יפספס את תורו.

כמו כן, השחקנים אינם רשאים לחרוג מגבולות המשחק – גם במקרה זה יש לציין זאת בהודעה וגם במקרה זה השחקן יפספס את תורו.

**שלב 1**  
ברגע שאחד השחקנים דורך על "שטיח המנצחים" יש להכריז עליו כמנצח ולסיים את המשחק.

* אחת הפונקציות (מבין כמה) שעליכם לכתוב על מנת לפתור את השלב הראשון היא הפונקציה:

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-java&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=boolean%2520isPlayerWon%2520(int%2520rowIndexA%252C%2520int%2520colIndexA%252C%2520int%2520rowIndexB%252C%2520int%2520colIndexB%252C%2520int%2520rugTopLeftCornerRow%252C%2520int%2520rugTopLeftCornerCol%252C%2520int%2520rugSize)%2520)

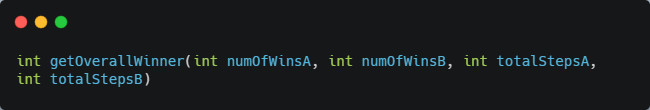
הפונקציה מחזירה TRUE אם קיים מנצח (A או B) אחרת, אם אף אחד לא ניצח, הפונקציה מחזירה false.

**שלב 2**

לאחר שסיימתם את שלב 1 ווידאתם שהוא אכן עובד כמצופה, יש לשכלל את המשחק לפי ההוראות הבאות:

יש לכתוב את התוכנית בצורה שתנהל מספר סיבובים של המשחק המתואר בתרגיל הקודם:  
בכל שלב יש לקלוט מהמשתמש האם ברצונו להמשיך לשחק, ‘y’ יהיה הקלט עבורו השחקן מעוניין להמשיך לשחק והקלטה ‘n’ יהיה עבור מקרה בו השחקן אינו מעוניין להמשיך לשחק. כאשר המשתמש מחליט לבסוף להפסיק לשחק, יש להדפיס סטטיסטיקות של כל המשחקים – כלומר, בכמה סיבובים ניצח כל אחד מהשחקנים ומי המנצח הכולל במשחק לפי כמות הניצחונות הגבוהה ביותר.  
במידה ויש שוויון, נשווה לפי כמות הצעדים הכוללת של השחקן. רק במידה ולשחקנים יש את אותה כמות צעדים כוללת ואותה כמות ניצחונות, נבחר מנצח בצורה שרירותית.

* אחת (מבין כמה) פונקציות שאתם צריכים לכתוב על מנת לפתור את השלב השני, היא הפונקציה:

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-java&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=int%2520getOverallWinner(int%2520numOfWinsA%252C%2520int%2520numOfWinsB%252C%2520int%2520totalStepsA%252C%2520int%2520totalStepsB))

הפונקציה מחזירה 1 אם A הוא המנצח (לפי הקריטריונים שהוגדרו לעיל), במידה וB הוא המנצח הפונקציה מחזירה 2.

**שלב 3**

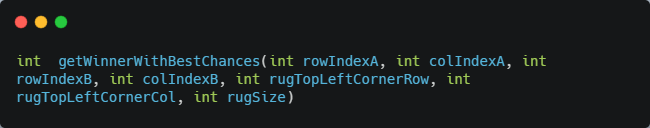
היות והרבה שחקנים התחילו לשחק במשחק, כמה שחקנים התחילו להיות מאוד טובים בו – מה שהפך את המשחק לדי משעמם...

בכדי להפוך את העניינים להיות קצת יותר מאתגרים, עלינו לכתוב את הפונקציה הבאה:

הפונקציה צריכה להחזיר איזה שחקן יכול לנצח בפחות צעדים!

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **O** | **O** | **O** |  |
|  |  |  |  |  |  | **O** | **O** | **O** |  |
|  |  |  |  |  |  | **O** | **O** | **O** |  |
| **A** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | **B** |

בדוגמה הנ"ל ניתן לראות **שלשחקן B יש יותר סיכוי**!  
שחקן B יכול לנצח ב 6 צעדים, ואילו שחקן A יכול לנצח ב7.  
יש להניח שאף אחד מהשחקנים לא נמצא בתוך השטיח.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-java&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=int%2520%2520getWinnerWithBestChances(int%2520rowIndexA%252C%2520int%2520colIndexA%252C%2520int%2520rowIndexB%252C%2520int%2520colIndexB%252C%2520int%2520rugTopLeftCornerRow%252C%2520int%2520rugTopLeftCornerCol%252C%2520int%2520rugSize))במידה ולשחקן A יש סיכוי גבוה יותר לנצח הפונקציה תחזיר 1.  
במידה ולשחקן B יש סיכוי גבוה יותר לנצח הפונקציה תחזיר 2.

אחרת, הפונקציה תחזיר 0

**דגשים**

* יש להשתמש בכלים שנלמדו בשקי החול על מנת לפתור את התרגול המסכם, כולל חלוקה לפונקציות, סטנדרטים וכו'.
* בסיום ביצוע התרגול המסכם, **חובה** לשלוח את קובץ הsource (הקובץ בו כתוב הקוד) אל כתובת המייל [coursetichnutidf@gmail.com](mailto:coursetichnutidf@gmail.com).

יש לכתוב בנושא המייל את הפרטים הבאים באופן הבא:

[שם מלא] - [מספר אישי: XXXXXXX] - [ת"ז: XXXXXXXXX]

לדוגמה:

עומר גדות - מספר אישי: 8232827 ת"ז 206755436

* חשוב להקפיד על השמות של הפונקציות
* יש להגיש בפורמט [java 8](https://www.oracle.com/java/technologies/java8.html)

**בהצלחה!**