**תרגיל בית 4 – תיעוד חיצוני**

מטרת התוכנית: לאפשר משחק דמקה בין שני משתתפים כך שכל משתמש יכול להיות אדם או מחשב, ועבור כל משתמש מוגדר זמן מקסימלי לביצוע מהלך.

קלט התוכנית: התוכנית תקלוט מהמשתמש בעזרת המקלדת פקודות המתפעלות את התוכנית. כל פקודה תינתן לתוכנית כמחרוזת המתחילה בשם הפקודה ולאחריה הארגומנטים של הפקודה. להלן הפקודות החוקיות:

* **start [wPlayer] [wTime] [bPlayer] [bTime]**

פקודה זו יכולה להתבצע רק במידה ואין משחק פעיל.

הפקודה מתחילה משחק דמקה חדש בעזרת הארגומנטים הבאים:

[wPlayer] – שחקן לבן, יכול לקבל את הערך h עבור אדם או c עבור מחשב.

wTime]] – זמן מקסימלי לביצוע מהלך עבור השחקן הלבן.

[bPlayer] – שחקן שחור, יכול לקבל את הערך h עבור אדם או c עבור מחשב.

[bTime] – זמן מקסימלי לביצוע מהלך עבור השחקן השחור.

* **stop**

פקודה זו יכולה להתבצע רק במידה וישנו משחק פעיל.

הפקודה עוצרת את המשחק, מדפיסה את ערך ה- Winning Parameter עבור השחקן הנוכחי ומדפיסה את השחקן המנצח. במידה ויש תיקו התוכנית תדפיס הודעה מתאימה.

* **time**

פקודה זו יכולה להתבצע רק במידה וישנו משחק פעיל.

הפקודה תדפיס כמה זמן נותר לשחקן הנוכחי לבצע את תורו.

* **move [Source index] [Destination index]**

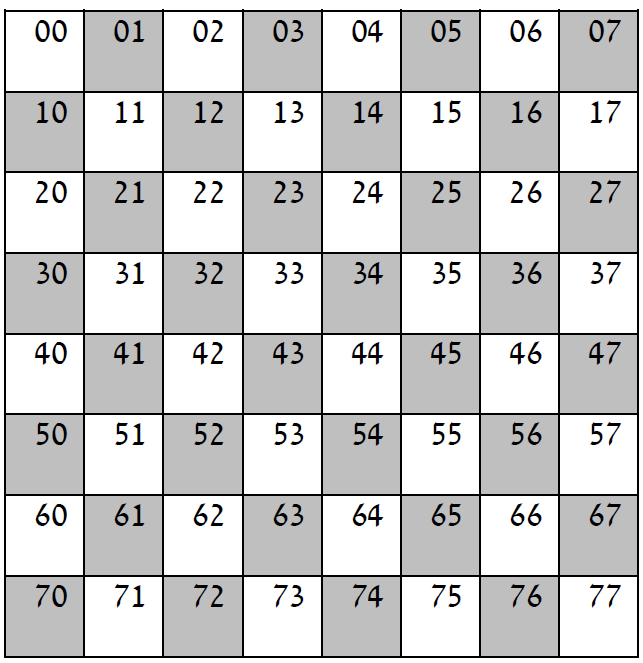
פקודה זו יכולה להתבצע רק כאשר ישנו משחק פעיל ובתורו של שחקן המוגדר כאדם.

הפקודה מזיזה כלי משחק של השחקן הנוכחי בעזרת הארגומנטים הבאים:

[Source index] – אינדקס המייצג משבצת בלוח המשחק אשר בו נמצא כלי השייך לשחקן הנוכחי אותו רוצים להזיז.

Destinationindex]] – אינדקס המייצג משבצת בלוח המשחק אשר אליו רוצים להזיז את הכלי.

כל אינדקס מאופיין ע"י שתי ספרות כאשר כל ספרה בין 0 ל- 7.



טבלת האינדקסים בלוח המשחק:

בכל מקרה בו הוכנסה פקודה שאינה קיימת או שהארגומנטים לא חוקיים תודפס הודעת השגיאה: Cannot execute "[command]" כאשר "[command]" תוחלף בפקודה שהוזנה.

פלט התוכנית: במהלך ריצת התוכנית תודפסנה הודעות שונות למסך המתארות את מצבה הנוכחי של התוכנית. כמו כן התוכנית תדפיס את הפקודות המוקלדות ע"י המשתמש ((echo.

דוגמה להודעות המודפסות במהלך ריצת התוכנית : בעת התחלת משחק חדש –

Starting a new game

White player is Human with 30 seconds per move

Black player is Computer with 5 seconds per move

ביצוע מהלך ע"י המחשב: כאשר שחקן המחשב רוצה לבצע מהלך במשחק הוא נעזר באלגוריתם MinMax אשר ממומש בפונקציה הרקורסיבית getMaxWP שכתבתנו בת"ב 3. כיוון שהמחשב מוגבל בזמן החישוב ולא בעומק הרקורסיה המחשב משתמש בפונקציה הנ"ל ע"י שימוש באלגוריתם Iterative Deepening. אלגוריתם זה מאפשר לו להגדיל את עומק החישוב ב-2 בכל פעם שהסתיים חישוב ועדיין לא תם הזמן. עם תום הזמן המחשב ייבצע את המהלך הטוב ביותר שחושב.

שגרות:

* movPiece – מקבלת במחסנית כתובת ללוח המשחק, אינדקס התחלה ואינדקס יעד של כלי משחק ומזיזה את הכלי.
* printStr – מקבלת כתובת של string בזיכרון ומדפיסה אותו למסך.
* asciConv – מקבלת במחסנית ערך דצימלי ומחזירה במחסנית כתובת לתחילת מחרוזת שתכיל את הערך ה- ASCII של הערך הדצימלי שהוכנס.
* chgTurn – מקבלת כמשתנה גלובלי את turn ומחליפה את ערכו לערך השחקן השני.
* printB – מקבלת כמשתנה גלובלי את כתובת הלוח הנוכחי ומדפיסה אותו.
* initBoard – מקבלת כמשתנים גלובליים את כתובת לוח המשחק וכתובת לוח תבנית (כפי שנראה בתחילת משחק) ומעתיקה את הלוח התבנית ללוח המשחק.
* Chkwins – מקבלת כמשתנים גלובליים את השחקן הנוכחי ואת כתובת הלוח. הפונקציה בודקת האם יש מנצח, ובמידה ויש מדפיסה את המנצח ועוצרת את המשחק.
* getMaxWP – פונקצית המעטפת של הפונקציה הרקורסיבית getWPRec. הפונקציה מקבלת במחסנית כתובות לשמירת ה- WP, SrcPos, DstPos אשר יחושבו ע"י הפונקציה. פונקציה זו נקראת כאשר המחשב נדרש לבצע מהלך.
* getWPRec – פונקציה רקורסיבית הנקראת ע"י getMaxWP ומחזירה במשתנים גלובליים את המהלך המיטבי שחושב.
* checkWin – מקבלת במחסנית ערך המייצג שחקן וכתובת של לוח המשחק ומחזירה במחסנית 1 אם השחקן ניצח, 2 אם ישנו תיקו (שני השחקנים נותרו חסרי מהלכים חוקיים עם מספר כלים זהה) או מינוס 1 בכל מקרה אחר.
* calcWP – מקבלת במחסנית את כתובת הלוח וערך השחקן הנוכחי ומחזירה ב R4 את ערך ה- WP עבור השחקן הנוכחי.
* countWB – מקבלת במחסנית את כתובת לוח המשחק ומחזירה במחסנית את מספר כלי המשחק מכל צבע.
* hasMove – מקבלת כמשתנה גלובלי את כתובת לוח המשחק ובמחסנית את ערך המייצג שחקן ומחזירה במחסנית 1 אם ישנו לפחות מהלך חוקי אחד עבור שחקן זה או 0 אחרת.
* decode – מקבלת כמשתנה גלובלי מחרוזת המייצגת פקודה ומבצעת שורות קוד רלוונטיות בהתאם לפקודה שהתקבלה במחרוזת.

משתנים גלובליים:

* Command – מחרוזת המכילה את הפקודה האחרונה שהוקלדה. בעזרת מחרוזת זו נפענח את הפקודה ונבצע קוד רלוונטי.
* pointer – מצביע למחרוזת Command בעזרתו נסרוק את המחרוזת.
* curChar – ערך ASCII האחרון שהוכנס.
* gameOn – ערך המייצג האם המשחק פעיל (1) או לא (0).
* errorCom ו- errPtr – מחרוזת ומצביע בעזרתם נדפיס הודעת פקודה לא חוקית.
* putDolar – ערך המייצג האם צריך לשים $ (1) או לא (0).
* turn – ערך המייצג את השחקן הנוכחי.
* source – כתובת בלוח המייצגת את המשבצת בה כלי המשחק אותו רוצים להזיז.
* dest – כתובת בלוח המייצגת את המשבת אליה רוצים להזיז את כלי המשחק.
* WinParam – ערך בו שמור ה WP שחושב בעזרת calcWP .
* Steps –עומק הרקורסיה שברצוננו לחשב. ערך **מועתק** ל- getWPRec.
* wPlayer ו- bPlayer – ערכים המייצגים האם השחקן הלבן/שחור בהתאמה הם אדם (1) או מחשב (2).
* wTime ו- bTime – ערכים המייצגים את הזמן בשניות שמוגדר לכל שחקן.
* wTimeStr, bTimeStr – מחרוזות המכילות זמן מקסימלי (כ- ASCII) לכל שחקן.
* wTimePtr, bTimePtr – מצביעים בעזרתם נדפיס את המחרוזות לעיל.
* TIMOUT – מציין האם נגמר הזמן עבור השחקן הנוכחי (1) או לא (0).
* tcounter – מונה הקטן ב- 1 בכל פסיקת שעון. בעזרתו נמדוד זמנים.
* msctr – מכיל את המילי-שניות של השניה הנוכחית.

מחרוזות:

הוגדרו מחרוזות שונות בעזרתן נדפיס הודעות שונות כדוגמת:

* Cannot execute
* Welcome message
* Game over

תרשים המתאר בכלליות את אופן ריצת התוכנית:

