

תרגיל מס' 7 – רקורסיה (ינון צוקרמן)

הוראות הגשה

- מועד אחרון להגשה: כמפורט באתר הקורס
- שם ההגשה: rec7
- כאשר אתם מתבקשים לקלוט מספר או אכן נכניס בטסטים מספר ולא אות.
- הפתרונות לתרגילים חייבים להיות רקורסיביים. פתרון לא רקורסיבי יזכה בציון 0.
- תרגיל rec71 שקול ל-50% מהציון, rec72 שקול ל-50% מהציון.

תרגיל 1:

כיתבו תוכנית המקבלת מחרוזת ומאפשרת לבצע את הפעולות הבאות על המספר (כאשר כל פעולה תהיה פונקציה רקורסיבית נפרדת):

- פעולה 1 – "תו אחרון". פעולה זו תדפיס את התו האחרון של המחרוזת. לדוגמא: בהנתן המחרוזת 12345, יודפס המספר 5 למסך.
- פעולה 2 – "מחיקת תו אחרון". פעולה זו תדפיס למסך את המחרוזת בלי התו האחרון. לדוגמא: בהנתן המחרוזת "23hg4" יודפס למסך המחרוזת "23hg".
- פעולה 3 – "מחיקת תו במיקום n". פעולה זו תדפיס למסך את המחרוזת בלי התו הנמצא במקום n. לדוגמא: בהנתן המחרוזת "23hg4" ו $n = 3$, יודפס למסך המחרוזת "23g4". במידה ו n לא בטווח תודפס הודעה "error" למסך.
- פעולה 4 – "בדיקת פולינדרום". פולינדרום הוא מחרוזת שנקראת אותו דבר מקדימה אחורה ומאחורה קדימה. לדוגמא "1881" ו "Able was I ere I saw Elba". פעולה זו תקבל מחרוזת ותדפיס "yes" אם המחרוזת היא פולינדרום או "no" במידה והיא לא פולינדרום.
- פעולה 5- יציאה מהתוכנית.

הערות:

- כל הפונקציות המבצעות את פעולות 1 עד 4 צריכות להיות רקורסיביות.
- אתם יכולים להניח נכונות קלט.
- כאשר המשתמש בוחר פעולה שלא מופיעה בתפריט, יש לתת הודעת "error" ולחכות למספר נוסף.
- יציאה מהתוכנית אפשרית רק בעזרת אופציה 5.
- אופן פעולת התוכנית מודגם בדוגמא הבאה:

```
>> 1 //option 1
>>"abcde" //input string from user
<< e //program output
>> 2 //option 2
>> "abcde" //input string from user
<< "abcd" //program output
>> 3 //option 3
>> "abcde" //input string from user
>> 3 //input char index from user
<< "abde" //program output
>> 3 //option 3
>> "abcde" //input string from user
>> 7 //input char index from user
<< "error" //error because the string has 5 characters
>> 4 //option 4
```

```

>> "abcde" //input string from user
<< no //not a polindrom
>> 4 //option 4
>>"abcdedcba" //input string from user
<< yes // a polindrom
>> 6 //option not exists
<< "error" //therefore the error output
>> 5 //exit the application.

```

תרגיל 2:

בתרגיל זה תכתבו פונקציה רקורסיבית שמנסה לפתור מבוך. מבוך הוא מטריצה דו-מימדית המכילה תווים 0 ו 1. התו 0 מציין דרך חסומה (שלא ניתן לעבור דרכה) ואילו התו 1 מציין דרך פנויה. לדוגמא:

```

11101
10110
11011
00010
00011

```

הפינה השמאלית העליונה של המבוך היא נקודת ההתחלה והפינה הימנית תחתונה היא נקודת הסיום של המבוך. ניתן לנוע רק בצעדים ימינה/שמאלה/מעלה/מטה (צעדים אלכסוניים אסורים).

המטרה שלכם היא לכתוב פונקציה (רקורסיבית כמובן) שפותרת את המבוך (אתם יכולים להניח שכל מבוך שתקבלו כקלט יכיל פתרון אחד בדיוק).

התוכנית תדפיס כפלט את המבוך הפתור עם השינוי הבא : דרכים פנויות (עם התו 1) שביקרתם בהם והם חלק מפתרון המבוך יהפכו לתו 9. לדוגמא : המבוך הקודם בגרסה הפתורה שלו יראה כך :

```

99901
10990
11091
00090
00099

```

הערות:

- הפונקציה המרכזית שבוחנת את עבירות המבוך חייבת להיות רקורסיבית !!!
- התוכנית אמורה להיות מסוגלת להתמודד עם מבוכים בגדלים שונים.
- המבוך יקלט באופן שכל שורה תוכנס כמחרוזת ותסתיים בלחיצה על enter. סיום הכנסת השורות יקלט על ידי הכנסת המחרוזת end ולחיצה על enter.
- אופן פעולה התוכנית מודגם בדוגמא הבאה :

```

>>11101
>>10110
>>11010
>>00010
>>00011
>>end
<<99901
<<10990
<<11091
<<00090
<<00099

```