

מבוא למדעי המחשב בשפת C

סמסטר א', תשפ"ג

תרגיל 7

נושא: רקורסיה

תאריך הגשה: יקבע באתר הקורס

הנחיות לגבי התרגיל:

- בתרגיל זה אסור להשתמש בכל חומר שנלמד אחרי הנושא של רקורסיה.
- שימו לב שבתרגיל זה עליכם לממש אוסף של פונקציות ועליכם לממש את כל הפונקציות הללו.
- בנוסף, בתרגיל זה אתם מקבלים קובץ עם main מוכן, ועליכם לשלב את הפונקציות המבוקשות. מותר לכם רק להוסיף שורות קוד אל הקובץ, אסור למחוק אף אחת מהשורות או להוסיף שורות קוד ל-main או לפונקציות העוזר שכבר כתובות. עליכם לוודא שהתוכנית עובדת כהלכה.
- תעדו את התוכנית כמו שצריך ובפרט את הפונקציות. בראש כל פונקציה יש לכתוב מה היא עושה, ואם יש הנחות כלשהן לגבי הפרמטרים שלה.
- הקפידו על שמות משמעותיים לפונקציות, למשתנים, ועל הגדרת קבועים מתאימים היכן שנדרש. כמו כן הקפידו על קוד קריא, מסודר ולא מסורבל (עד כמה שניתן).
- בתרגיל זה, אין צורך לבדוק את תקינות הקלט בשום צורה. יש להניח שהמשתמש הכניס קלט תקין.
- בתוכנית זו אפשר להשתמש בפונקציה strlen מתוך ספריית string. אין להשתמש בפונקציות ספרייה נוספות.

פונקציה 1

כתבו פונקציה רקורסיבית:

```
bool isEven(int num, int dig);
```

המקבלת כקלט מספר טבעי num וספרה dig. על הפונקציה להחזיר true אם ורק אם מספר המופעים של הספרה dig ב-num הוא זוגי.

למשל: `IsEven(9545, 5)` הוא true.

ואילו `IsEven(9545, 4)` הוא false.

פונקציה 2

בהינתן נקודה (x,y) במישור כאשר x ו- y הם מספרים שלמים אי-שליליים, נקרא לנקודות בעלות הקואורדינטות השלמות במלבן שקודקודיו הם $(0,0)$, $(0,y)$, (x,y) , $(x,0)$ "הרשת של (x,y) ".

למשל הרשת של $(2,3)$ כוללת את הנקודות הבאות:

(0,3) (1,3) (2,3)
(0,2) (1,2) (2,2)
(0,1) (1,1) (2,1)
(0,0) (1,0) (2,0)

מבוא למדעי המחשב בשפת C

סמסטר א', תשפ"ג

תרגיל 7

נושא: רקורסיה

תאריך הגשה: יקבע באתר הקורס

כתבו פונקציה רקורסיבית המקבלת כפרמטרים את זוג הקואורדינטות (x, y) , ומחזירה את מספר המסלולים השונים ברשת של (x, y) מהנקודה $(0, 0)$ לנקודה (x, y) . ההתקדמות במסלול, המותרת בכל שלב היא תזוזה של יחידה אחת ימינה או תזוזה של יחידה אחת למעלה.

```
int howManyPaths(int x, int y);
```

הערה: אין צורך לפרט את המסלולים.

לדוגמא: אל הנקודה $(2, 1)$ ישנם שלושה מסלולים שונים והם:

$(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (2, 1)$

$(0, 0) \rightarrow (1, 0) \rightarrow (2, 0) \rightarrow (2, 1)$

$(0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (2, 1)$

פונקציה 3

כתבו מימוש **רקורסיבי** של הפונקציה

```
int biggestLowPower(int x, int num);
```

הפונקציה מקבלת שני מספרים שלמים חיוביים (גדולים מאפס) x ו- num , ומחזירה את החזקה המכסימאלית של x (x^y) עבור y שלם אי-שלילי) הקטנה או שווה ל num . אפשר להניח ש $x \geq 2$.

דוגמאות

- $biggestLowPower(2, 256)$ תחזיר 2^8 .
- $biggestLowPower(2, 200)$ תחזיר 2^7 .
- $biggestLowPower(12, 20)$ תחזיר 12^1 .
- $biggestLowPower(12, 2)$ תחזיר 12^0 .

מבוא למדעי המחשב בשפת C

סמסטר א', תשפ"ג

תרגיל 7

נושא: רקורסיה

תאריך הגשה: יקבע באתר הקורס

פונקציה 4

כתבו מימוש **רקורסיבי** של הפונקציה:

```
int partSum(int num);
```

הפונקציה מקבלת כקלט מספר חיובי num . הערך המוחזר מהפונקציה הוא סכום הספרות ב- num למעט הספרה הימנית ביותר - ספרת ה LSD. (סכום זה מוגדר להיות אפס אם ב- num יש ספרה אחת).

דוגמאות:

- $partSum(27231)$ תחזיר 14 כי $14 = 2 + 7 + 2 + 3$.
- $partSum(2)$ תחזיר 0.

פונקציה 5

כתבו פונקציה **רקורסיבית**:

```
void intToStr(int num, char s[]);
```

המכניסה למערך s את מחרוזת התווים המייצגים את הערך העשרוני של השלם האי-שלילי num . אפשר להניח שבמחרוזת s יש מספיק מקום.

דוגמאות:

- $intToStr(234, s)$ תציב ב- s את המחרוזת "234".
- $intToStr(0, s)$ תציב ב- s את המחרוזת "0".

פונקציה 6

כתבו מימוש **רקורסיבי** של הפונקציה:

```
void fillMaxPrefixesArray (int numbers[], int n,  
                           int maxPrefixesArray[]);
```

הפונקציה מקבלת כקלט מערך $numbers$ של מספרים ואת גודלו n (שלם חיובי). על הפונקציה למלא את n התאים הראשונים במערך $maxPrefixesArray$ כך שבמקום ה- i יופיע המספר המקסימאלי מבין המספרים $\{numbers[0], numbers[1], \dots, numbers[i]\}$.

הערה

ניתן להניח שבמערך $maxPrefixesArray$ יש מספיק מקום להכיל את התוצאה הנדרשת.

מבוא למדעי המחשב בשפת C

סמסטר א', תשפ"ג

תרגיל 7

נושא: רקורסיה

תאריך הגשה: יקבע באתר הקורס

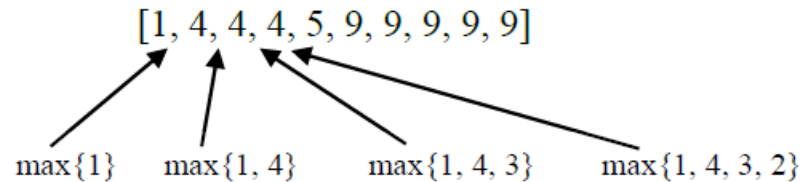
למשל numbers הוא המערך

```
numbers=[1,4,3,2,5,9,7,3,7,8]
```

ההפעלה

```
fillMaxPrefixesArray (numbers, 10, maxPrefixesArray)
```

תעדכן את עשרת התאים הראשונים של maxPrefixesArray להיות:



פונקציה 7

כתבו מימוש **רקורסיבי** של הפונקציה:

```
void getMinToStart (int numbers[], int n);
```

הפונקציה מקבלת כקלט מערך numbers של מספרים ואת גודלו n (שלם חיובי). הפונקציה תשנה את סדר איברי המערך numbers כך שבסוף ריצתה, numbers[0] יכיל את המספר הקטן ביותר מבין המספרים ב-n. הסדר של שאר המספרים במערך אינו משמעותי.

דוגמאות

עבור המערך [-3,5,-2,1,-7,8,9,3] פלט אפשרי הוא [-7,-3,5,-2,1,8,9,3].

עבור המערך [7,9,2,0,0,1] פלט אפשרי הוא [0,7,9,2,0,1].

פונקציה 8

כתבו מימוש **רקורסיבי** של הפונקציה:

```
void combineArrays (int sortedArr1[], int size1,  
int sortedArr2[], int size2)
```

הפונקציה מקבלת כקלט מערך ממוין בסדר עולה sortedArr1 שגודלו הלוגי size1 (אי שלילי), ומערך נוסף ממוין בסדר עולה sortedArr2 שגודלו הלוגי size2 (אי שלילי). ידוע שבמערך sortedArr2 יש מספיק מקום ל size1+size2 נתונים. הפונקציה תשלב בצורה ממוינת בסדר עולה את הנתונים בשני המערכים לתוך המערך sortedArr2.

מבוא למדעי המחשב בשפת C

סמסטר א', תשפ"ג

תרגיל 7

נושא: רקורסיה

תאריך הגשה: יקבע באתר הקורס

בשאלה זו אין להשתמש במערך עזר.

דוגמאות

- אם לפני הפעלת הפונקציה המערכים היו:
`sortedArr1=[-3,1,5], sortedArr2=[-7,-1,-1,1,6]`
עבור `size1=3` ו `size2=5`.
אז אחרי הפעלת הפונקציה המערך `sortedArr2` שווה ל:
`sortedArr2=[-7,-3,-1,-1,1,1,5,6]`
- אם לפני הפעלת הפונקציה המערכים היו:
`sortedArr1=[], sortedArr2=[-7,-1,-1,1,6]`
עבור `size1=0` ו `size2=5`.
אז אחרי הפעלת הפונקציה המערך `sortedArr2` שווה ל:
`sortedArr2=[-7,-1,-1,1,6]`

פונקציה 9

כתבו מימוש **רקורסיבי** של הפונקציה:

```
int countSmaller(int arr[], int start, int end, int num)
```

המקבלת כקלט מערך **ממוין** המכיל מספרים שלמים השונים זה מזה, אינדקס התחלה `start`, אינדקס סיום `end` ומספר שלם `num`.

על הפונקציה להחזיר את מספר המספרים במערך (בין האינדקסים `start` ל-`end`, כולל), שקטנים ממש מ-`num`.

הערות: * המספר `num` לא בהכרח נמצא במערך.

* שימו דגש על יעילות הפונקציה !!!

דוגמאות

- עבור הקלט `arr = [1,3,4,7,8,13]`, `start = 0`, `end = 5`, `num = 8`
הפונקציה תחזיר 4 (מאחר שהמספרים 1,3,4,7 קטנים ממש מ-8).
- עבור הקלט `arr = [1,3,4,7,8,13]`, `start = 0`, `end = 5`, `num = 5`
הפונקציה תחזיר 3 (מאחר שהמספרים 1,3,4 קטנים ממש מ-5).