

תרגיל 1

ממשו את הפונקציה הרקורסיבית

long ReverseNum (unsigned long n)

המקבלת מספר שלם חיובי מסוג long ומחזירה מספר חדש עם סדר ספרות הפוך.

דוגמה:

עבור GetReverse (123456789) יוחזר 987654321

עבור GetReverse (111222333) יוחזר 333222111

תרגיל 2

ממשו את הפונקציה הרקורסיבית

int isSubsetSum (int a[], int size, unsigned int k)

אשר מקבלת מערך של מספרים שלמים, את גודלו, ומספר שלם חיובי k ומחזירה 1 אם קיימת תת קבוצה של איברים מהמערך אשר סכומם הינו k. אחרת הפונקציה תחזיר 0. תת הקבוצה יכולה להיות בפרט איבר בודד.

דוגמה: בהינתן

int a[4] = {-1, -3, 5, 2};

הפונקציה isSubsetSum (a, 4, 6)

תחזיר 1 שכן סכום האיברים בתת הקבוצה {1,-2,5} הוא 1

תרגיל 3

כתבו את הפונקציה הרקורסיבית int lastN(int num,int n)

הפונקציה מקבלת מספר שלם חיובי n ומספר שלם חיובי n. הפונקציה מחזירה מספר חדש המכיל את N הספרות האחרונות של המספר num. אם n גדול מאורך המספר num אז הפונקציה תחזיר -1.

דוגמאות:

lastN (12345,2) → 45

lastN (12345,4) → 2345

lastN (12345,6) → -1

תרגיל 4

כתבן פונקציה רקורסיבית:

```
void ChangeFrame(int mat[N][N],int rowCol,int size);
```

הפונקציה מקבלת מערך דו ממדי של מספרים (ה N מוגדר כ define), מספר המציין שורה ועמודה (rowcol) ומספר size. על הפונקציה להפוך שורות עם עמודות של תת המטריצה בגודל sizeXsize.

דוגמה:

עבור המטריצה mat הבאה כאשר size=4 וכי rowcol=0 (מתחילים תמיד מאיבר בשורה אפס ועמודה אפס)

1	6	11	16	5
2	7	12	17	10
3	8	13	18	15
4	9	14	19	20
21	22	23	24	25

לאחר הפעלת הפונקציה, mat תראה כך:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

שימו לב:

מותר להשתמש בלולאות ובפונקציות עזר אך עיקר הפתרון אמור להתבצע בצורה רקורסיבית. במידה ועיקר הפתרון לא יתבצע בצורה רקורסיבית הניקוד על שאלה זו יהיה אפס.

תרגיל 5:

כתוב פונקציה רקורסיבית:

```
int concatFirst (int num, int start);
```

הפונקציה מקבלת מספר שלם חיובי num ומספר שלם חיובי start. הפונקציה מחזירה מספר חדש אשר הינו שרשור של המספר start עם המספר num. (יש להניח כי num ו- start גדולים מ-0).

דוגמאות:

concatFirst (45, 123) => 12345

concatFirst (21, 12) =>1221

concatFirst (12345, 6) => 612345

concatFirst (2,6) => 62

תרגיל 6:

כתוב פונקציה רקורסיבית:

int MergeNumbers(int n1, int n2);

הפונקציה מקבלת שני מספרים n_1 , n_2 חיוביים גדולים מ-0. הפונקציה מחזירה מספר חדש המהווה מיזוג של שני המספרים מימין לסירוגין באופן הבא:
(עבור n_1 מספר באורך 5 ו- n_2 מספר באורך 2)

$n_1 = a_5a_4a_3a_2a_1$

$n_2 = b_2b_1$

יחזיר את המספר הבא:

$a_5a_4a_3b_2a_2b_1a_1$

דוגמאות:

MergeNumbers (45, 123) => 12435

MergeNumbers (12345, 67) => 1236475

MergeNumbers (24, 13) => 1234

תרגיל 7

נתונה הסדרה: 1, 2, 3, 6, 4, 13, 7, 24, 11, 42, ...

- שלושת האיברים הראשונים בסדרה הם: 1, 2, 3
- כל איבר שנמצא במקום זוגי בסדרה, החל מהמקום ה-4, הוא סכום שלושת האיברים הקודמים לו.
- כל איבר הנמצא במקום אי-זוגי בסדרה, החל מהמקום ה-5, הוא ההפרש בערך המוחלט שבין שני האיברים במקומות הזוגיים שלפניו.

כתוב אלגוריתם רקורסיבי לחישוב האיבר ה- n בסדרה זו
(הקלט לפעולה יהיה המקום ה- n [החל מ-1], והערך שיוחזר יהיה ערכו של האיבר במקום זה). ניתן להניח כי הקלט יהיה מספר שלם חיובי בלבד.

int Series(int n);

תרגיל 8:

בתרגיל זה ניתן להניח כי המספרים המועברים כפרמטר חיוביים, גדולים מאפס, ומורכבים מספרות ממוינות בסדר עולה משמאל לימין.

ממשו את הפונקציה הרקורסיבית

void FindCommonDigit(unsigned long n1, unsigned long n2, int arr[], int* sumCommon)

המקבלת שני מספרים שלמים חיוביים מסוג long, מערך של מספרים שלמים (ניתן להניח שמאותחל האפסים וגודלו כמספר הספרות במספר הקצר מבין n_1 ו- n_2 , ומצביע sumCommon). המצביע מצביע למשתנה המכיל בתוכו את מס' הספרות הזהות (ללא כפילויות) בשני המספרים. על

הפונקציה לעדכן את המערך arr שיכיל את הספרות המשותפות לשני המספרים ואת
*sumCommon למס' הספרות הזרות. ניתן להניח ש
*sumCommon מאותחל לפני הזימון לפונקציה ב 0.
דוגמת הרצה עבור תוכנית main לדוגמה:
עבור :

FindCommonDigit (113355799, 222446688),arr,&sum)

נקבל:

sum=0 | Arr[]={0,0,0,0,0,0,0,0}

עבור :

FindCommonDigit (111355777, 2234466899),arr,&sum)

נקבל:

sum=1 | Arr[]={3,0,0,0,0,0,0,0}

עבור :

FindCommonDigit (112233445, 112233445),arr,&sum)

נקבל:

sum=5 | Arr[]={5,4,3,2,1,0,0,0}