



Amac

Temel C programlama

Soru

Aşağıda header'ları bulunan fonksiyonları yazını ve test dosyasında testlerini gerçekleştiriniz:

```
// karakterin harf olup olmadini kontrol eden fonksiyon
int isLetter(char c);
// karakterin rakam olup olmadigini kontrol eden fonksiyon
int isDigit(char c);
// sayinin cift sayi olup olmadigini kontrol eden fonksiyon
int isEven(int value);
// sayinin asal sayi olup olmadigini kontrol eden fonksiyon
int isPrime(int value);
// sayinin basamaklari toplamini hesaplayan fonksiyon
int digitSum(int value);
// sayinin tersini hesaplayan fonksiyon
int reverseNumber(int value);
// kucuk harfi buyuk harfe ceviren fonksiyon
void upperCase(char c);
// buyuk harfi kucuk harfe ceviren fonksiyon
void downcase(char c);
// karakterin buyuk harf olup olmadini kontrol eden fonksiyon
int isUpperCase(char c);
// faktoriyel hesabi yapan fonksiyon
int factoriel(int value);
// ust hesabi yapan fonksiyon
int power(int base, int degree);
```

```

// toplama hesabi yapan fonksiyon
int add(int value1, int value2);
// ikinci dereceden bir bilinmeyenli denklemin koklerini hesaplayan fonksiyon
void findRoots(int a, int b, int c);
// sayinin palindrome olup olmadigini hesaplayan fonksiyon
int isPalindromeNumber(int value);
// metnin palindrome olup olmadigini hesaplayan fonksiyon
int isPalindromeText(char *c, int size);
// // karakterin sesli harf olup olmadini kontrol eden fonksiyon
int isVowel(char c);
// dizinin icinde o karakterin olup olmadigini hesaplayan fonksiyon
int exist(int *arr, int size, int value);
// dizinin icinde aranan karakterin varsa ilk nerde oldugunu hesaplayan fonksiyon
int getIndex(int *arr, int size, int value);
// dizinin icinde aranan karakterin kac defa gectigini hesaplayan fonksiyon
int count(int *arr, int size, int value);
// dizinin max sayisini hesaplayan fonksiyon
int findMax(int *arr, int size);
// dizinin 2. max sayisini hesaplayan fonksiyon
int findMax2(int *arr, int size);
// dizinin min sayisini hesaplayan fonksiyon
int findMin(int *arr, int size);
// dizinin 2. min sayisini hesaplayan fonksiyon
int findMin2(int *arr, int size);
// dizinin elemanlari toplamini hesaplayan fonksiyon
int sumArray(int *arr, int size);
// dizinin ortalamasini hesaplayan fonksiyon
double averageArr(int *arr, int size);
// dizinin standart sapmasini hesaplayan fonksiyon
double deviationArr(int *arr, int size);
// ekrana matrisi yazdiran fonksiyon
void printMatrix(int **matrix, int row, int column);
// matrisi transpose eden fonksiyon
int** transposeMatrix(int **matrix, int row, int column);
// matrisin elemanlarini 50-100 arasinda dolduran fonksiyon
int** fillMatrix(int **matrix, int row, int column);

```

Onemli Notlar:

1. TP calismalarini “header file – source file – test file” seklinde gerceklestirmeniz gerekiyor: Yazdiginiz fonksiyonların headerlarını header dosyasina, fonksiyonların kodlarını source dosyasina, test kodlarını da test dosyasina yaziniz.
2. Dosyalarinizi zipleyip “OgrenciNo_IsimSoyisim_TPX.tar.gz” seklinde isimlendirerek sisteme yukleyiniz.
3. TP calismalarini linux ortaminda, herhangi ek bir IDE’ye gerek duymadan (kate, gedit vs) yapılması tercihimizdir. Farkli platformlarda hazirlanan odevlerin kontrolu sirasinda dosya uzantisi vb. sebeplerden calismamasi ogrencinin sorumlulugundadir.