

Temel Programlama II

Furkan : İSİM
Kaya : SOYİSİM

Armstrong Sayıları

191216002 : NUMARA

Merhaba,

Bu raporu 18/04/2020 tarihinde BIL102 kodlu Temel Programlama 2 dersinde verilmiş olan armstrong numaralarını bulma ödevi için hazırladım. İyi okumalar dilerim.

Kodumu yazmaya armstrong numarasını bulan fonksiyonda lazım olacak fonksiyonları yazmakla başladım. Us alma, sayı uzunluğunu bulma ve istediğim basamaktaki sayıyı bana getirecek fonksiyon.

Us(int sayi, int us):

```
int Us(int sayi, int us){  
    if(us==0) return 1;  
    int sonuc=1;  
    for(int i=1;i<=us;i++){  
        sonuc=sonuc*sayi;  
    }  
    return sonuc;  
}
```

Üst alma fonksiyonumu solda görebilirsiniz. Verilen sayıyı istenildiği kadar kendisiyle çarpıp sonucu döndürüyor.

SayiDegeri(int sayi, int basamak):

```
int SayiDegeri(int sayi, int basamak){  
    int istenilenSonda = sayi/Us(10,basamak-1);  
    return istenilenSonda % 10;  
}
```

İstenilen basamaktaki sayıyı döndüren fonksiyon. Şu mantıkla

çalışıyor; İçine aldığı sayıyı 10'nun katlarına bölerek istenilen basamağın birler basamağına dönüşmesini sağlıyor ve 10'a modunu alıp birler basamağındaki sayıyı geri döndürüyor.

SayiUzunlugu(int sayi):

```
int SayiUzunlugu(int sayi){  
    int l=1;  
    while(sayi>9){  
        l++;  
        sayi/=10;  
    }  
    return l;  
}
```

Sayı uzunluğu fonksiyonum. Şu mantıkla çalışıyor. Sayı 10'a bölünüp küçülebiliyorsa basamak vardır demektir. Her bölümde basamak uzunluğunu 1 artırıp en sonunda sonucu bana döndürüyor.

ArmstrongMu(int sayi):

```
int ArmstrongMu(int sayi){  
    int toplam = 0;  
    int sayiUzunlugu=SayiUzunlugu(sayi);  
    for (int i=1; i<=sayiUzunlugu; i++)  
    {  
        int rakam = SayiDegeri(sayi, i);  
        toplam+=Us(rakam, sayiUzunlugu);  
    }  
    return toplam==sayi;  
}
```

İçine aldığı sayının basamaklarında gezip her basamağındaki rakamı sayının toplam uzunluğu kadar kendisiyle çarpıp toplam değişkenine ekliyor. Fonksiyonun sonunda da bu toplam sayıya eşit mi değil mi onu döndürüyor. Eşitse 1 değilse 0.

Main():

```
int main(void) {
    int bulunanlar[1000];
    int bulunanlari=0;
    int alt,ust;
    printf("Lütfen aralığın alt limitini giriniz: ");
    scanf("%d", &alt);
    printf("Lütfen aralığın üst limitini giriniz: ");
    scanf("%d", &ust);
    for(int i=alt; i<=ust; i++){
        if(ArmstrongMu(i)){
            bulunanlar[bulunanlari++]=i;
        }
    }
    if(bulunanlari==0){
        printf("Girdiğiniz aralıkta hiç armstrong sayısı yok!");
    }else{
        printf("%d ile %d arasındaki armstrong sayıları: ", alt, ust);
        for(int i=0;i<bulunanlari;i++){
            printf("%d ", bulunanlar[i]);
        }
    }
    return 0;
}
```

Programı oluştururken main fonksiyonumu yazmayı sona bıraktım.

Önce kullanıcıdan alt ve üst aralığı alıyor ardından for döngüsüyle bu aralıktaki her bir rakamı armstrong mu değil mi kontrol ediyor eğer armstrong numarasıysa fonksiyonun başında tanımladığım int arrayin içine alıyor ve bulunanlar sayacımı 1 arttırıyor.

Tüm rakamları dolaşıp armstrongları topladıktan sonra sayacın artıp artmadığını kontrol ediyoruz eğer artmamışsa sayı bulamadık demektir ekrana uyarı verip programı bitiriyoruz. Eğer sayacımız artıyorsa for döngüsüyle bulunanlar dizimizi dolaşıp her birini ekrana bastırıyoruz.