

```
#include <stdio.h>
```

```
#include <math.h>
```

```
//
```

```
//#ODEV 11 Cozumler
```

```
//
```

```
//
```

```
//Soru1
```

```
//Recursive fonksiyon 1-n arasi rakam ve sayilari ekrana yazdirir.
```

```
//n degerini Global olarak tanitirsak isimiz kolaylasir.
```

```
//diger turlu rec1N(int x, int n) seklinde yazip n'i sabit tutmaliyiz.
```

```
//Sorunun birden cok cozumu vardır. Tek cozum bu diye dusunmeyiniz.
```

```
int n;
```

```
int rec1N(int x){
```

```
    //Eger sayi x n'den buyuk olmusse yazdirmaya veya yeniden cagirmaya gerek yok
```

```
    //return'u bos yapip fonksiyonu durdurabiliriz.
```

```
    if (x == n+1)
```

```
        return;
```

```
    printf("%d\n", x++); //burda x yazdirilir ve daha sonra 1 arttirilir.Bu sebeple return rec1N(x)  
yeterli olur. Aksi halde rec1N(x+1) yapmamiz yeterli olacakti.
```

```
    return rec1N(x);
```

```
}
```

//Soru2:

//sayi 0 olana kadar bolunur ve her bolunmede count degeri 1 arttirilir.

//count degerini global almamiz isimizi kolaylastiracaktir.

//sayinin 0'dan farkli oldugu varsayimmi ile yapilmistir.

int count = 0;

int kacBasamak(int x){

 //Eger sayi artik bolunemezse tum basamaklar hesaplanmistir demektir.

 if (x == 0)

 return;

 //sayi 0dan farkli ise demekki daha basamak sayisi vardır.

 count++;

 return kacBasamak(x/10);

}

//soru4:

//Bu defa yukaridaki islemdede recursive cagirim yapmadan once son basamagi yazdirmamiz gerekmektedir.

int tersYazdirma(int x){

 //Eger sayi artik bolunemezse tum basamaklar hesaplanmistir demektir.

 if (x == 0)

 return;

 //sayi 0dan farkli ise demekki daha basamak sayisi vardır.

 printf("%d", (x% 10));

 return tersYazdirma(x/10);

}

//Soru3:

//EKOK: En küçük ortak kat

//EBOB: En büyük ortak Bölen

```
int EBOB(int x, int as, int bs){
```

```
    if (as % x == 0 && bs % x == 0)
```

```
        return x;
```

```
    return EBOB(x-1, as, bs);
```

```
}
```

```
int EKOK(int x, int as, int bs){
```

```
    if ( x %as == 0 && x % bs == 0)
```

```
        return x;
```

```
    return EKOK(x+1, as, bs);
```

```
}
```

```
int main(){
```

```
    printf("n:");
```

```
    scanf("%d", &n);
```

```
    //1den baslatiriz ki n'e kadar yazdirsın.
```

```
rec1N(1);
```

```
int sayi = 0;
```

```
printf("Basamak sayisi bulunacak sayiyi gir:");
```

```
scanf("%d", &sayi);
```

```
kacBasamak(sayi);
```

```
printf("%d sayisi %d basamaklidir...\n", sayi, count);
```

```
tersYazdirma(sayi);
```

```
int a,b;
```

```
printf("\nEbob ve Ekok icin a v b sayilarini giriniz:\n");
```

```
printf("a:");
```

```
scanf("%d", &a);
```

```
printf("b:");
```

```
scanf("%d", &b);
```

```
int max = a;
```

```
if (max < b)
```

```
    max = b;
```

```
int ebo = EBOB(max, a, b);
```

```
int eko = EKOK(max, a, b);
```

```
printf("\nEKOK:%d", eko);  
printf("\nEBOB:%d", ebo);  
}
```