

## 42 C00 Çözümleri

write Fonksiyonu Parametreleri

**İlk Parametre:** Hedefi belirtir:

- 0: Standart giriş (input)
- 1: Standart çıkış (output)
- 2: Standart hata (error)

**Orta Parametre:** Yazdırılacak verinin adresini belirtir. (Bellek adresi: & işareti kullanılır)

**Son Parametre:** Yazdırılacak byte sayısını belirtir.

```
char str[] = "Hello";
write(1, str, 5); // "Hello" yazdırılır
write(1, str, 3); // "Hel" yazdırılır
```

**ex00:**

Write kullanarak ekrana girilen char değeri yazdırmamızı istiyor

```
#include <unistd.h>
void ft_putchar(char c)
{
    write(1, &c, 1);
}

int main(void)
{
    ft_putchar('c');
    return (0);
}
```

**ex01:**

Bizden a'dan z'ye kadar olan değerleri yazdırmamız talep ediliyor.

```
#include <unistd.h>
void ft_print_alphabet(void)
{
    char str = 'a';
    while(str <= 'z')
    {
        write(1, &str, 1);
        str++;
    }
}
```

```
int main ()
{
    ft_print_alphabet();
    return (0);
}
```

**ex02:**

Burada z'den a'ya kadar olan değerleri yazdırmamız talep ediliyor.

```
#include <unistd.h>
void ft_print_reverse_alphabet(void)
{
    char str ='z';
    while(str >= 'a')
    {
        write(1,&str,1);
        str--;
    }
}
```

```
int main ()
{
    ft_print_reverse_alphabet();
    return (0);
}
```

**ex03:**

Burada rakamları küçükten büyüğe yazdırmamız isteniyor.

```
#include <unistd.h>
void ft_print_numbers(void)
{
    int str = '0';
    while(str <= '9')
    {
        write(1,&str,1);
        str++;
    }
}
```

```
int main ()
{
    ft_print_numbers();
    return (0);
}
```

#### **ex04:**

Girilen Sayı Negatif bir sayı ise N Pozitif ise P yazacak kısaca, protitipte bize int tipinde n değişkenini vermiş.

```
#include <unistd.h>
void ft_is_negative(int n)
{
    char negative = 'N';
    char positive = 'P';
    if(n<0)
    {
        write(1,&negative,1);
    }
    else
    {
        write(1,&positive,1);
    }
}
```

```
int main ()
{
    ft_is_negative(5);
    ft_is_negative(-5);
    return (0);
}
```

#### **ex05:**

3 farklı rakamdan oluşan ve sıralı şekilde artan kombinasyonları yazdırmamız isteniyor max sayımız yani 789 olacak ...

İç içe döngülerde işlem sırası içten dışa doğru gider. Yani, dıştaki döngü bir kez çalıştığında, içteki döngüler tamamlanmadan dış döngü bir sonraki iterasyona geçmez.

```

#include <unistd.h>

void ft_print_comb(void)
{
    char a;
    char b;
    char c;

    a = '0';
    while (a <= '7')
    {
        b = a + 1;
        while (b <= '8')
        {
            c = b + 1;
            while (c <= '9')
            {
                write(1, &a, 1);
                write(1, &b, 1);
                write(1, &c, 1);
                if (a != '7')
                    write(1, " ", 2);
                c++;
            }
            b++;
        }
        a++;
    }
}

int main ()
{
    ft_print_comb();
    return (0);
}

```

Üç değişken tanımlanıyor: a, b, ve c. a başlangıçta 0.

- Dış döngü (a): a 7'den küçük olduğu sürece çalışıyor. İlk başta a = 0.
- İç döngü (b): b, a + 1 olarak başlıyor ve 8'den küçük olduğu sürece çalışıyor. İlk başta b = 1.

- En iç döngü (c): c, b + 1 olarak başlıyor ve 9'dan küçük olduğu sürece çalışıyor. İlk başta c = 2.
- Ekran yazdırma: Her döngüde a, b, ve c ekrana yazdırılıyor. Örneğin, ilk durumda: 0 1 2.
- Virgül ve boşluk ekleme: Eğer a henüz 7'ye eşit değilse, yazdırılan sayıların sonuna , ekleniyor.

Döngü akışı:

- c döngüsü: c 9'a ulaşana kadar artıyor (012, 013, ...).
- b döngüsü: c döngüsü tamamlandığında, b bir artıyor ve c = b + 1 oluyor. (023, 024, ...).
- a döngüsü: b döngüsü tamamlandığında, a bir artıyor ve her şey yeniden başlıyor (134, 135, ...).
- Sonuç: Döngüler bu şekilde iç içe ilerleyerek tüm kombinasyonları ekrana yazdırıyor. Örneğin:  
012, 013, 014, ..., 789.

#### ex06:

Bu projede amaç, artan düzende sıralanan 00 ile 99 arasındaki tüm farklı iki basamaklı sayı kombinasyonlarını ekrana yazdırmak...

```
#include <unistd.h>
```

```
void ft_putchar(char c)
{
    write(1, &c, 1);
}
```

```
void ft_print_comb2(void)
{
    int a;
    int b;

    a = 0;
    while (a <= 98)
    {
        b = a + 1;
        while (b <= 99)
        {
            ft_putchar(a / 10 + '0');
            ft_putchar(a % 10 + '0');
            ft_putchar(' ');
```

```

        ft_putchar(b / 10 + '0');
        ft_putchar(b % 10 + '0');

        if (a != 98 || b != 99)
            write(1, " ", 2);

        b++;
    }
    a++;
}
}
int main()
{
    ft_print_comb2();
    return (0);
}

```

**ft\_putchar** fonksiyonu ile ekrana çıktı verebilmek için write fonksiyonunu kullanıyoruz. Bu fonksiyonu oluşturduk.

- **a** ve **b** adında iki integer değişkeni tanımlıyoruz. Başlangıç değeri olarak **a = 0** veriyoruz.
- **a <= 98** şartıyla bir while döngüsü başlatıyoruz. Bu döngü, **a** 98'e ulaşana kadar çalışacak.
- **b = a + 1** diyerek **b**'yi **a**'nın bir fazlası yapıyoruz. İç içe başka bir while döngüsüyle **b <= 99** şartına kadar devam ediyoruz.
- **ft\_putchar** fonksiyonunu kullanarak **a** ve **b**'nin her iki basamağını ekrana yazdırıyoruz. İlk önce onlar basamağı, sonra birler basamağı yazdırılır.
- **if** koşuluyla **a** ve **b** son değerlerine ulaşmadıkça virgül ve boşluk koyuyoruz.
- İçteki döngü **b++** ile artarken, dıştaki döngü de **a++** ile artıyor.

**NOT:** Bir sayının 10'a bölümü onlar basamağını 10'a olan modunu almak ise birler basamağını bulmamıza yarar.

**Ex07:**

Parametre olarak girilen sayıyı ekrana yazdıran bir fonksiyon yazmamız isteniyor bu alınan sayı tüm integer sayı tipini kapsamalı. Bu kod, bir tam sayıyı ekrana yazdırmak için kullanılır.

- Pozitif sayılar, basamaklarına ayrılarak sırayla yazdırılır.

- Negatif sayılar için önce eksi işareti yazdırılır, ardından pozitif kısmı yazdırılır.
- Kodun içinde, tek tek karakterleri yazdıran bir fonksiyon (ft\_putchar) ve sayının tamamını işleyen bir başka fonksiyon (ft\_putnbr) bulunur.

```
#include <unistd.h>
```

```
void ft_putchar(char c)
{
    write(1, &c, 1);
}
```

- '-' değerini yazdırmamız gerek. Onun için bunu kullanmamız gerekiyor.

```
void ft_putnbr(int nb)
{

if (nb == -2147483648)
{
write(1, "-2147483648", 11);
return;
}
```

- Eğer sayı -2147483648 ise, direkt olarak "write" fonksiyonuyla bu sayı yazdırılır.
- Bu özel bir durum çünkü bu sayı negatife çevrilemez (taşma hatası olur)

```
if(nb<0)
{
    ft_putchar('-');
    nb = -nb;
}
```

Eğer sayı negatifse:

İlk olarak eksi işareti ekrana yazdırılır.  
Ardından, sayı pozitif yapılır (nb= -nb;)

```
if(nb>9)
{
    ft_putnbr(nb / 10);
    ft_putnbr(nb % 10);
}
```

Eğer sayı 9'dan büyükse:

- Sayı onlar basamağına ve birler basamağına ayrılır.
- Rekürsiyon (kendini çağırma) ile önce büyük basamaklar, sonra küçük basamaklar yazdırılır.

Örnek ;

```
ft_putnbr(123) çağrısı;
```

- $123 > 9$  olduğu için  $123 / 10 = 12$  ve  $123 \% 10 = 3$  hesaplanır.
- `ft_putnbr(12)` ve `ft_putnbr(3)` çağrılır.
- Bu işlemler devam ederek 1, 2, ve 3 sırayla yazdırılır.

```
else
```

```
{  
    ft_putchar(nb + 48);  
}
```

Eğer sayı 9 veya daha küçükse (örneğin 0 ile 9 arasındaysa):

- Bu sayı direkt bir karaktere dönüştürülerek yazdırılır.
- +48 Neden?
  - ASCII tablosunda, 0 karakterinin karşılığı 48'dir.
  - Bu yüzden, sayıya 48 ekleyerek doğru karakter bulunur.

```
ft_putnbr(7):
```

- $7 + 48 = 55$  olur, bu da '7' karakterine karşılık gelir.