

## 42 C04 Çözümleri

Not : ex00'dan ex02' ye kadar daha önceden yaptım. O yüzden tekrardan açıklamayacağım.

### ex00: ft\_strlen

```
#include <stdio.h>
int      ft_strlen(char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
        i++;
    }
    return (i);
}

int main ()
{
    printf("Karakter Uzunluğu : %d \n",ft_strlen("Herkes Merhaba"));
    return (0);
}
```

-----

### ex01 : ft\_putstr

```
#include <unistd.h>
void      ft_putstr(char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
        write(1,&str[i],1);
        i++;
    }
}

int main()
{
    ft_putstr("Merhaba");
    return (0);
}
```

## ex02 : ft\_putnbr

```
#include <unistd.h>

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

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

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

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

    if (nb <= 9)
    {
        ft_putchar(48 + nb);
    }
}

int main()
{
    ft_putnbr(2000);
    ft_putchar('\n');
    ft_putnbr(9);
    ft_putchar('\n');
    ft_putnbr(-2000);
    ft_putchar('\n');
    return (0);
}
```

### ex03:

#### Amaç:

ft\_atoi fonksiyonu, bir karakter dizisini (string) tam sayıya (int) dönüştürmek için kullanılır. Özellikle kullanıcıdan alınan veya dosyadan okunan metinsel sayıların matematiksel işlemlerde kullanılabilmesini sağlar.

#### Adım Adım Açıklama:

##### 1. Değişkenler:

```
int i = 0;
int sign = 1;
int result = 0;
```

- i: Dizideki karakterleri tek tek kontrol etmek için kullanılan indeks değişkeni. Başlangıçta sıfır (0) olarak tanımlanır.
- sign: Sayının pozitif (+) veya negatif (-) olup olmadığını belirleyen değişken. Başlangıçta 1 olarak tanımlanır (pozitif).
- result: Sayının tam sayı değerini tutan değişken. Başlangıçta sıfır (0) olarak tanımlanır.

##### 2. Boşluk Karakterlerini Atlama:

```
while (str[i] == ' ' || (str[i] >= 9 && str[i] <= 13))
    i++;
```

- Bu döngü, başta yer alan boşlukları (' ') ve ASCII kontrol karakterlerini (tab, newline, vb.) atlar.
- str[i] == ' ' ifadesi, boşluk karakterini kontrol eder.
- (str[i] >= 9 && str[i] <= 13) ifadesi, ASCII değerleri 9 ile 13 arasında olan boşluk karakterlerini kontrol eder. Bunlar:
  - 9: Tab (\t)
  - 10: Yeni satır (\n)
  - 11: Dikey tab (\v)
  - 12: Form feed (\f)
  - 13: Satır başı (\r)

##### 3. İşaret (+ veya -) Kontrolü:

```
while (str[i] == '+' || str[i] == '-')
{
    if (str[i] == '-')
        sign *= -1;
    i++;
}
```

- Eğer karakter dizisinde + veya - işareti varsa, bu işaretin sayının pozitif mi yoksa negatif mi olduğunu belirlemek için kontrol edilir.
- Eğer - (eksi) işareti varsa, sign değişkeni -1 ile çarpılır ve sayıyı negatif hale getirir.
- Eğer + işareti varsa, sign değişkeni değişmeden kalır ve sayı pozitif olarak kabul edilir.

#### 4. Sayısal Karakterleri Dönüştürme:

```
while (str[i] >= '0' && str[i] <= '9')
{
    result = result * 10 + (str[i] - '0');
    i++;
}
```

1) Bu döngü, sayısal karakterleri (0'dan 9'a kadar) işler.

2)  $str[i] - '0'$  ifadesi, bir karakterin ASCII değerinden '0' karakterinin ASCII değeri çıkarılarak, karakterin tam sayı karşılığı elde edilir.

- Örneğin:  $'5' - '0' = 5$

3)  $result = result * 10 + (str[i] - '0')$  ifadesi, her yeni rakamı mevcut result değerine ekler. İlk başta result sıfırdır, sonra her yeni rakam eklenir.

Örneğin, "123" stringini işlerken:

- İlk turda:  $result = 0 * 10 + 1 = 1$
- İkinci turda:  $result = 1 * 10 + 2 = 12$
- Üçüncü turda:  $result = 12 * 10 + 3 = 123$

4) Eğer  $str[i]$  bir rakam değilse (örneğin "42abc" gibi), döngü durur ve result şu ana kadar hesaplanan değeri döndürür (42).

#### 5. Sonucu Döndürme:

```
return result * sign;
```

- Son olarak, result değişkeni işaret (sign) ile çarpılır ve döndürülür.
  - Eğer sayının işareti negatifse ( $sign = -1$ ), result negatif olur.
  - Eğer sayı pozitifse ( $sign = 1$ ), result olduğu gibi pozitif döner.

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

```
int ft_atoi(char *str)
{
    int i = 0;
    int sign = 1;
    int result = 0;

    while(str[i] == ' ' || (str[i] >= 9 && str[i] <= 13))
        i++;

    while(str[i] == '-' || str[i] == '+')
    {
        if(str[i] == '-')
        {
            sign *= -1;
        }
        i++;
    }
}
```

```
while(str[i] >= '0' && str[i] <= '9')
{
    result = result * 10 + (str[i] - '0');
    i++;
}

return result * sign;
}

int main(void)
{
    char str1[] = "    ---+---+1234ab567";
    char str2[] = "+456";

    printf("Sonuç 1 : %d\n", ft_atoi(str1));
    printf("Sonuç 2 : %d\n", ft_atoi(str2));

    return 0;
}
```