

# Hazır Fonksiyonlar



Gömülü Sistem  
Laboratuvarı

# string kutuphaneleri

## Standard C Library



stdio.h



time.h



strings.h



stdlib.h

*not a complete list*

test.c

```
#include <stdio.h>  
printf("...");
```

[https://www.tutorialspoint.com/c\\_standard\\_library/ctype\\_h.htm](https://www.tutorialspoint.com/c_standard_library/ctype_h.htm)



# string arama

**strchr(s, c) :**

s içerisinde c karakterini arar.

Bulundugu yeri tutan isaretçiyi döndürür.

Yoksa **NULL** (0 adresi) pointer döndürür.

**strstr(s1, s2) :**

s1 içerisinde s2 stringini arar.

Bulundugu yeri tutan isaretçiyi döndürür.

Yoksa **NULL** pointer döndürür.





# strstr

```
#include <stdio.h>
#include <string.h>
int main() {
    char cumle[200];
    char aranan[20];
    printf("cumle girin:\n");
    gets(cumle);
    printf("aranani girin:");
    scanf("%s", aranan);

    char *ara_ptr = strstr(cumle, aranan);
    if (ara_ptr == NULL)
        printf("\nyok\n");
    else
        printf("\nilk gectigi index: %d\n", ara_ptr-cumle);
    return 0;
}
```



strstr

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

```
#include <string.h>
```

```
int main() {
```

```
char cumle[200];
```

```
char aranan[20];
```

```
printf("cumle girin:\n");
```

```
gets(cumle);
```

```
printf("aranani girin:");
```

```
scanf("%s", aranan);
```

```
int sayac = 0;
```

```
char *ara_ptr = strstr(cumle, aranan);
```

```
while (ara_ptr != NULL) {
```

```
printf("yer: %d\n", ara_ptr-cumle);
```

sayac++;

// bir sonraki karkaterden itibaren tekrar arıyoruz

```
ara_ptr = strstr(ara_ptr+1, aranan);
```

}

```
printf("cumlede %d kere geciyor\n", sayac);
```

```
return 0;
```

}



# strtok

```
#include <stdio.h>
#include <string.h>
int main() {
```

```
    char cumle[200] = "Bu ornek;noktalama isaretlerini ve bosluklari kullanarak cumleyi  
                        parcalar. Bu cumle,test icin yazilmistir.";
```

```
    const char *ayrac = " .,;"; // bosluk, nokta ve virgul, noktali virgul...
```

```
    char *kelime = strtok(cumle, ayrac);
    while (kelime != NULL) {
        printf("%s\n", kelime);
```

```
        kelime = strtok(NULL, ayrac);
```

```
    }
```

```
    return 0;
```

```
}
```



Bu  
ornek  
noktalama  
isaretlerini  
ve  
bosluklari  
kullanarak  
cumleyi  
parcalar  
Bu  
cumle  
test  
icin  
yazilmistir

# Dizilerde işlem yapmak

`memcpy(a, b, size):`

b'yi a'ya kopyalar. size dizinin bellekteki boyutu.

`memset :`

diziye char boyutunda sabit bir deger atar.



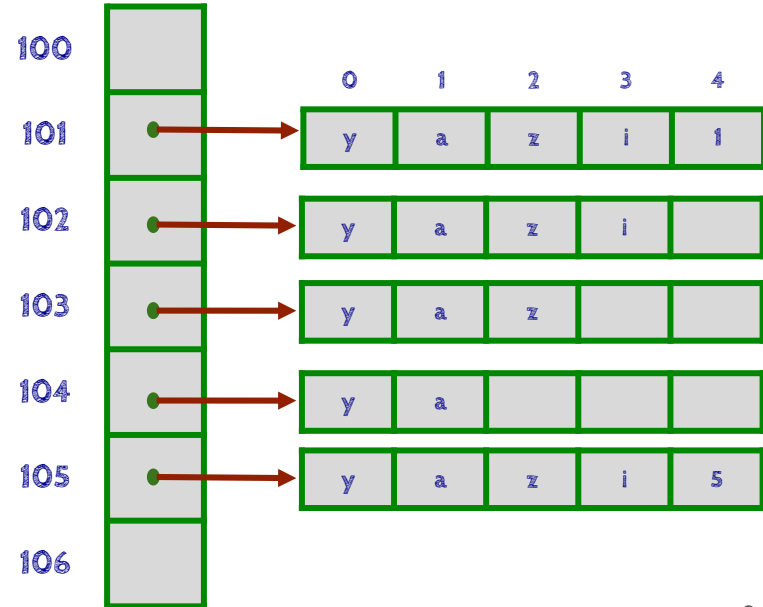
# Dizilerde işlem yapmak

```
#include <stdio.h>
#include <string.h>
int main() {
    int i;
    /****** string dizisi *****/
    // 5 tane 20 karakterlik string dizisi
    char string_dizisi[5][20];

    for (i = 0 ; i < 5 ; i++) {
        printf("%d. kelimeyi girin: ", i+1);
        scanf("%s", string_dizisi[i]);
    }
    printf("girilen kelimeler:\n");
    for (i = 0 ; i < 5 ; i++)
        printf("%s\n", string_dizisi[i]);

    printf("\n");

    return 0;
}
```



# memcmp

```
#include <stdio.h>
#include <string.h>
int main() {
    /***** dizi karsilastirma *****/
    int dizi_1[5] = {1,2,3,4,5};
    int dizi_2[5] = {1,2,3,4,5};

    if (dizi_1 == dizi_2) //HATA!!! bu sekilde kullanilamaz
        printf("esit\n");
    else
        printf("esit degil\n");

    int sonuc = memcmp(dizi_1, dizi_2, sizeof(int)*5);
    if (sonuc == 0)
        printf("esit\n");
    else
        printf("esit degil\n");
    printf("\n");

    return 0;
}
```

# memcpy

```
#include <stdio.h>
#include <string.h>
int main() {

    /****** dizi kopyalama *****/
    int dizi_1[5] = {1,2,3,4,5};
    int dizi_3[9] = {0};

    // dizi_1'yi dizi_3'e kopyalar
    memcpy(dizi_3, dizi_1, sizeof(int)*5);

    // dizi_3'u ekrana yaz
    for (int i = 0 ; i < 9 ; i++)
        printf("%d ", dizi_3[i]);
    printf("\n\n");

    return 0;
}
```

# memset

```
#include <stdio.h>
#include <string.h>
int main() {

    /****** dizi sifirlama *****/
    int dizi_1[5] = {1,2,3,4,5};

    // dizi_3'u sifirlar
    memset(dizi_1, 0, sizeof(int)*5);

    // dizi_3'u ekrana yaz
    for (int i = 0 ; i < 5 ; i++)
        printf("%d ", dizi_1[i]);
    printf("\n\n");

    return 0;
}
```

# memcpy

```
#include <stdio.h>
#include <string.h>
int main() {
    int dizi_1[5] = {1,2,3,4,5};
    int dizi_3[9] = {0};

    /****** diziye ortadan kopyalama *****/
    // dizi_3'un 3. elemanından itibaren dizi_1'i atamak istiyoruz
    // dizi_3'un [2,7] arasına dizi_1'in [0,4] arasını atamak
    memcpy(dizi_3+2, dizi_1, sizeof(int)*5);

    // dizi_3'u ekrana yaz
    for (int i = 0 ; i < 9 ; i++)
        printf("%d ", dizi_3[i]);
    printf("\n\n");

    return 0;
}
```

# Sorular

