

Hazır Fonksiyonlar



Gömülü Sistemler
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



isdigit

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;

    printf("karakter girin: ");
    scanf(" %c", &c1);
    if ( isdigit(c1) )
        printf("%c karakteri rakamdir\n",
c1);
    else
        printf("%c karakter rakam
degildir\n", c1);
    return 0;
}
```



ispunct

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;

    printf("karakter girin: ");
    scanf(" %c", &c1);
    if ( ispunct(c1) )
        printf("%c karakteri noktalama isaretidir\n",
c1);
    else
        printf("%c karakter noktalama isareti
degildir\n", c1);
    return 0;
}
```



isalpha & (isupper, islower)

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;

    printf("karakter girin: ");
    scanf(" %c", &c1);
    if ( isalpha(c1) ) {
        if ( isupper(c1) ) // buyuk harf ise, kucuk harfi
            yaz
            printf("kucuk harf karsiligi:
            %c\n", tolower(c1));
        if ( islower(c1) ) // kucuk harf ise, buyuk harfi
            yaz
            printf("buyuk harf karsiligi:
            %c\n", toupper(c1));
        }
    return 0;
}
```

abcdefghijklmnopqrstuvwxyz

String çevirme fonk.

atoi : string → integer

atol : string → long

atof : string → double

strtod(s, ptr) : s(string) → double

Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

strtol(s, ptr) : s(string) → double

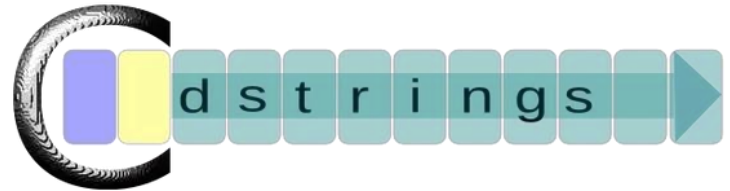
Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

atoi & atol & atof

```
#include <stdio.h>
#include <stdlib.h> // atoi, atof, atol gibi fonksiyonlar stdlib'dedir
int main() {
    char s[50];
    int x;
    long y;
    double z;
    // kucuk sayilarla ve 15-20 basamakli sayilarla test edin
    // sonra kesirli sayilarla test edin
    printf("sayi girin: ");
    gets(s);

    x = atoi(s); // stringin icerigini integer'a cevirir
    y = atol(s); // stringin icerigini long'a cevirir
    z = atof(s); // stringin icerigini double'a cevirir

    printf("int: %d\n", x);
    printf("long: %ld\n", y);
    printf("double: %lf\n", z);
    return 0;
}
```



strtod & strtol

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

```
#include <stdlib.h>
```

```
int main () {
```

```
    char str[30] = "20.30300 Bu kısım yazıdır";
```

```
    char *cevilemeyen_kisim;
```

```
    double cevriken_kisim;
```

```
    cevriken_kisim = strtod(str, &cevilemeyen_kisim);
```

```
    printf("Çevriken kısım: |%f|\n", cevriken_kisim);
```

```
    printf("Çevrilmeyen kısım: |%s|", cevilemeyen_kisim);
```

```
    return(0);
```

```
}
```



String karşılaştırma fonk.

Kullanılışı;

`strcmp(s1, s2)`

Geri dönüş değeri;

`s1` ve `s2` string'lerini karşılaştırır

Eitse 0

Alfabetik olarak `s1` < `s2` ise negatif sayı döndürür

Alfabetik olarak `s1` > `s2` ise negatif sayı döndürür

String karsilastirma fonk.

```
#include <stdio.h>
#include <string.h>
int main() {
    char s1[100] = "aaaa";

    // yanlis kullanim, bu sekilde karsilastirma yapilamaz
    if (s1 == "aaaa")
        printf("esit\n");
    else
        printf("esit degil\n");

    printf("%u\n", s1);
    printf("%u\n", "aaaa");
    printf("\n");

    return 0;
}
```



strcmp

```
#include <stdio.h>
#include <string.h>
int main() {
    char yazi1[10] = "test";
    char yazi2[10] = "test";
    printf("yazi1: %s\n", yazi1);
    printf("yazi2: %s\n", yazi2);
    if (yazi1 == yazi2)
        printf("yazi1 ve yazi2 esit\n");
    else
        printf("yazi1 ve yazi2 esit degil\n");
    printf("yazi1: %u\n", yazi1);
    printf("yazi2: %u\n", yazi2);
    int sonuc = strcmp(yazi1, yazi2);
    if (sonuc == 0)
        printf("yazi1 ve yazi2 esit\n");
    else if (sonuc < 0)
        printf("alfabetik olarak yazi1 < yazi2\n");
    else
        printf("alfabetik olarak yazi1 > yazi2\n");
    return 0;
}
```



strcmp

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

int main() {
    int sonuc;

    sonuc = strcmp("telefon", "tablet");

    if (sonuc == 0)
        printf("a ve b esit\n");
    else if (sonuc < 0)
        printf("alfabetik olarak telefon
tablet\n");
    else
        printf("alfabetik olarak telefon
tablet\n");

    printf("\n");

    return 0;
}
```



String kopyalama fonk.

`strcpy(s2, s1):`

s1 stringini s2'ye kopyalar

`strcat(s2, s1):`

s1 stringini s2'nin sonuna ekler



strcpy

```
#include <stdio.h>
#include <string.h>
int main() {
    char s1[100] = "aa bbb cc";
    char s2[100];

    int karakter_sayisi = strlen(s1);
    printf("strlen(s1) : %d\n", karakter_sayisi)

    // s1'i s2'ye kopyalar
    strcpy(s2, s1);

    printf("s1: %s\n", s1);
    printf("s2: %s\n", s2);

    return 0;
}
```



strcat

```
#include <stdio.h>
#include <string.h>
int main() {
    char s1[100] = "aa bbb cc";
    char s2[100];

    strcpy(s1, "test 1 ");
    strcpy(s2, "deneme 2 ");
    printf("s1: %s\n", s1);
    printf("s2: %s\n", s2);

    strcat(s1, s2);
    printf("strcat(s1, s2) sonrasinda\n");
    printf("s1: %s\n", s1);
    printf("s2: %s\n", s2);

    return 0;
}
```



strcpy

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

    char s1[100] = "aa bbb cc";
    char s2[100];

    strcpy(s1, "test 1 ");
    strcpy(s2, "deneme 2 ");
    strcpy(s1+strlen(s1), s2);

    printf("strcpy(s1+strlen(s1), s2) sonrasinda\n");
    printf("s1: %s\n", s1);
    printf("s2: %s\n", s2);

    return 0;
}
```



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.



Sorular

