

Karakter fonksiyonları

isalnum, isalpha, iscntrl, isdigit, isgraph, islower, isprint, ispunct, isspace, isupper, isxdigit, isascii, isblank

```
#include <stdio.h>
#include <ctype.h> // isdigit, isalpha gibi karakter fonksiyonlari icin gerekli

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);

    if ( isalpha(c1) )
        printf("%c karakteri harftir\n", c1);
    else
        printf("%c karakter harf degildir\n", c1);

    if ( ispunct(c1) )
        printf("%c karakteri noktalama isaretidir\n", c1);
    else
        printf("%c karakter noktalama isareti degildir\n", 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;
}
```

String çevirme fonksiyonları

atoi : "string"ın içeriğini "integer"a çevirir

atol : "string"ın içeriğini "long"a çevirir

atof : "string"ın içeriğini "double"a çevirir

strtod(s, ptr) : s "string"ının içeriğini "double"a çevirir. Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

strtol(s, ptr) : s "string"ının içeriğini "long"a çevirir. Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

```
#include <stdio.h>
#include <stdlib.h> // atoi, atof, atol gibi fonksiyonlar stdlib'de

int main() {
    char s[50];
    int x;
    long y;
    double z;

    // küçük sayılarla ve 15-20 basamaklı sayılarla test edin
    // sonra kesirli sayılarla test edin
    printf("sayı girin: ");
    gets(s);

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

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

    /*****/
    double a;

    char *cevilemeyen_kisim;
    a = strtod(s, &cevilemeyen_kisim);
    printf("a: %lf\n", a);
    printf("sayıdan_sonraki: %s\n", cevillemeyen_kisim);

    printf("\n");

    /*****/
    long b;

    // 10 -> onluk sayı tabanından çevirdiğimiz için
    b = strtol(s, &cevilemeyen_kisim, 10);

    printf("b: %ld\n", b);
    printf("cevilemeyen_kisim: %s\n", cevillemeyen_kisim);

    return 0;
}
```

String karşılaştırma örnekleri

strcmp(s1, s2) : s1 ve s2 stringlerini karşılaştırır. Eşitse 0, alfabetik olarak s1 < s2 ise negatif sayı, s1 > s2 ise pozitif sayı döndürür.

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

    // yanlis kullanım, 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");
    /*****/

    char a[10] = "test";
    char b[10] = "test";
    printf("a: test\n");
    printf("b: test\n");

    // yanlis kullanım, bu sekilde karsilastirma yapilamaz
    if (a == b)
        printf("a ve b esit\n");
    else
        printf("a ve b esit degil\n");

    printf("a: %u\n", a);
    printf("b: %u\n", b);
    printf("\n");

    /*****/

    int sonuc = strcmp(a, b);

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

    /*****/

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

    printf("a: telefon\n");
    printf("b: tablet\n");

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

```

else
    printf("alfabetik olarak a > b\n");
printf("\n");

/*****/
return 0;
}

```

String kopyalama fonksiyonları

strcpy(s2, s1) -> s1 stringini s2'ye kopyalar

strcat(s2, s1) -> s1 stringini s2'nin sonuna ekler

```

#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);

    printf("\n");

    /*****/

    // s1'e "test 1 " kopyala (bu sekilde atama yapma yerine kullanilabilir)
    strcpy(s1, "test 1 ");

    // s2'ye "deneme 2 " ata
    strcpy(s2, "deneme 2 ");

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

    // s2'yi s1'in sonuna ekler
    strcat(s1, s2);
    printf("strcat(s1, s2) sonrasinda\n");
    printf("s1: %s\n", s1);
    printf("s2: %s\n", s2);

    printf("\n");

    /*****/

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

```

```
// s2'yi s1'in sonuna eklemenin baska yolu
strcpy(s1+strlen(s1), s2);

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

return 0;
}
```

String arama fonksiyonları

strchr(s, c) : s içerisinde c karakterini arar. Bulunduğu yerin pointerını döndürür. Yoksa NULL (0 adresi) pointer döndürür.

strstr(s1, s2) : s1 içerisinde s2 stringini arar. Bulunduğu yerin pointerını döndürür. Yoksa NULL pointer döndürür.

```
#include <stdio.h>
#include <string.h>
int main() {
    char cumle[100];
    char karakter;
    char aranan_str[20];

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

    printf("aranan karakteri girin:\n");
    scanf("%c", &karakter);

    char *ilk_gectigi_ptr = strchr(cumle, karakter);
    if (ilk_gectigi_ptr == NULL)
        printf("yok\n");
    else
        printf("ilk gectigi index: %d\n", ilk_gectigi_ptr-cumle);

    char *son_gectigi_ptr = strrchr(cumle, karakter);
    if (son_gectigi_ptr == NULL)
        printf("yok\n");
    else
        printf("son gectigi index: %d\n", son_gectigi_ptr-cumle);

    printf("\n");

    /*****/

    printf("aranani girin:");
    scanf("%s", aranan_str);

    ilk_gectigi_ptr = strstr(cumle, aranan_str);
    if (ilk_gectigi_ptr == NULL)
        printf("\nyok\n");
    else
        printf("\nilk gectigi index: %d\n", ilk_gectigi_ptr-cumle);

    return 0;
}
```

String arama örnek

```
int main() {
    char cumle[200], 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 ariyoruz
    ara_ptr = strstr(ara_ptr+1, aranan);
}
printf("cumlede %d kere geciyor\n", sayac);

return 0;
}

```

String parçalama fonksiyonları

strtok, strtok_r

```

#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.";
    // printf("cumle girin:\n");
    // gets(cumle);

    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);
    }

    /*****/

    char ornek_2[] = "BLM105:Programlama I:3;BLM212:Sistem Programlama:3;BLM302:Otomata
Teorisi:4";
    printf("\n%s\n\n", ornek_2);

    // ic ice kullanilacaksa strtok_r kullanilmali.
    // bunun icin ic ice kullanildigi kadar kayit pointeri gerekli
    char *kayit1, *kayit2;

    // ayrac -> ;
    char *ders = strtok_r(ornek_2, ";", &kayit1);
    while (ders != NULL) {

        char *kod = strtok_r(ders, ":", &kayit2);
        char *ad = strtok_r(NULL, ":", &kayit2);
        char *akts = strtok_r(NULL, ":", &kayit2);

        printf("kod: %s, ad: %s, akts: %s\n", kod, ad, akts);

        ders = strtok_r(NULL, ";", &kayit1);
    }
}

```

```
    return 0;  
}
```


Dizilerle ilgili fonksiyonlar

memcpy(a, b, size) : b'yi a'ya kopyalar. size dizinin bellekteki boyutu.
memset : diziye char boyutunda sabit bir deger atar.

```
#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("birinci kelimeyi girin: ");
        scanf("%s", string_dizisi[i]);
    }
    printf("girilen kelimeler:\n");
    for (i = 0 ; i < 5 ; i++)
        printf("%s\n", string_dizisi[i]);
    printf("\n");

    /***** 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");

    /***** dizi kopyalama *****/
    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 (i = 0 ; i < 9 ; i++)
        printf("%d ", dizi_3[i]);
    printf("\n\n");

    /***** dizi sifirlama *****/
    // dizi_3'u sifirlar
    memset(dizi_3, 0, sizeof(int)*9);

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

    /***** 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 (i = 0 ; i < 9 ; i++)
    printf("%d ", dizi_3[i]);
printf("\n\n");

return 0;
}

```

Formatlı giriş / çıkış fonksiyonları

```

#include <stdio.h>
int main() {
    int a = 50;
    float b = 99.9558;
    char c[] = "deneme";

    // 10 karaktere sigacak sekilde yazmak
    printf("%10d\n", a);
    printf("%10f\n", b);
    printf("%10s\n", c);
    printf("\n");

    // virgulden sonraki basamak sayisi
    printf("%.4f\n", b); // virgulden sonra 4 basamak yazdir
    printf("%.2f\n", b); // virgulden sonra 2 basamak yazdir
    printf("%10.4f\n", b); // 10 karaktere sigacak sekilde virgulden sonra 4 basamak
    printf("\n");

    // basa 0 koyarak 6 basamak yazdirmek
    printf("%06d\n", 12);
    printf("%06d\n", 123);
    printf("\n");

    printf("0:%s\n", c); // normal yazdirma, boyutu anlamak icin basa ve sona : koyduk
    printf("1:%10s\n", c); // saga dayali 10 karaktere sigacak sekilde yazdir
    printf("2:%.3s\n", c); // ilk 3 karakteri yazdir
    printf("3:%-10s\n", c); // sola dayali 10 karaktere sigacak sekilde yazdir
    printf("4:%10.3s\n", c); // 10 karaktere sigacak sekilde ilk 3 karakteri yazdir
    printf("5:%.3s\n", c+2); // [2,5] arasi karakteri yazdir

    printf("\n");

    /***** stringe yazdirma *****/

    char s1[30];
    sprintf(s1, "sprintf ornek. %d %.2f", a, b);
    printf("%s\n", s1);

    /***** string'den okuma *****/

    char s2[] = "aaa 10 7.5";
    printf("s2: %s\n", s2);

    char x[20];
    int y;
    float z;
    sscanf(s2, "%s %d %f", x, &y, &z);

    printf("x: %s\n", x);
    printf("y: %d\n", y);
    printf("z: %f\n", z);
}

```

```
}    return 0;
```