

# Strings(Karakter Dizisi)

A graphic showing the word "Strings" enclosed in a rectangular box with double quotes at the corners.

“ Strings ”

**Suhap SAHIN**  
**Onur GÖK**

# Tanımlama ve Kullanım

```
#include <stdio.h>
int main() {
    char ad[20];
    printf("adinizi girin: ");

    scanf("%s", ad);

    printf("\nmerhaba %s\n\n", ad);

    return 0;
}
```

# Tanımlama ve Kullanım

*// Tanımlarken deger atama*

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

```
int main() {
```

```
    char s1[20] = "deneme2";
```

```
    printf("%s\n", s1);
```

```
    return 0;
```

```
}
```

# Tanımlama ve Kullanım

```
// sabit yapma
```

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

```
int main() {
```

```
    const char *s2 = "deneme3";
```

```
    printf("%s\n", s2);
```

```
    return 0;
```

```
}
```

# Tanımlama ve Kullanım

*// dizinin boyutunu atanan degere gore otomatik belirleme*

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

```
int main() {
```

```
    char s3[] = "deneme4";
```

```
    printf("%s\n", s3);
```

```
    return 0;
```

```
}
```

# Sonlandırma Karakteri

```
#include <stdio.h>
int main() {
    char kelime[20];

    printf("bir kelime girin: ");
    scanf("%s", kelime);

    int i;

    // tamamini karakter karakter yazdirirsak, girilmemis kisim rastgele olur
    for (i = 0 ; i < 20 ; i++)
        printf("%c", kelime[i]);

    printf("\n\n");

    return 0;
}
```

# Sonlandırma Karakteri

```
#include <stdio.h>
int main() {
    char kelime[20];

    printf("bir kelime girin: ");
    scanf("%s", kelime);

    int i;

    // stringin bittigini anlamak icin '\0' karakteri kullaniliyor
    // printf %s asagidaki dongu gibi calisiyor
    for (i = 0 ; kelime[i] != '\0' ; i++)
        printf("%c", kelime[i]);

    printf("\n\n");

    return 0;
}
```

# Karakterleri Say Tersten Yazdır

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

```
int main() {
```

```
    char kelime[20];
```

```
    int karakter_sayisi;
```

```
    int i;
```

```
    printf("bir kelime girin: ");
```

```
    // NOT: birden fazla kelime girilirse ilkini okur. bosluk, tab ve enter
```

```
    //kelime sonunu belirler
```

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

```
    printf("girilen kelime: %s\n", kelime);
```

```
    .....
```

```
    .....
```

```
    return 0;
```

```
}
```



# Karakterleri Say Tersten Yazdır

```
// karakter sayisini sayma islemi
```

```
karakter_sayisi = 0;
```

```
while (kelime[karakter_sayisi] != '\0')
```

```
    karakter_sayisi++;
```

```
printf("karakter sayisi: %d\n", karakter_sayisi);
```

```
// karakter sayisini sayma islemi
```

```
for (karakter_sayisi = 0 ; kelime[karakter_sayisi] != '\0' ; karakter_sayisi++) {
```

```
    /* bos dongu. sadece harf sayisini arttiriyor */
```

```
}
```

```
printf("karakter sayisi: %d\n", karakter_sayisi);
```

# Karakterleri Say Tersten Yazdır

```
// tersten yazdırma  
printf("tersten yazilisi: ");  
for (i = karakter_sayisi-1 ; i >= 0; i--)  
    printf("%c", kelime[i]);  
printf("\n");
```

# gets ve puts fonksiyonları

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

```
int main() {
```

```
    char s[100];
```

```
    // gets fonksiyonu ile cumle okunabilir (bosluklar dahil).
```

```
    // scanf %s ile kelime okunabiliyor. scanf %s bosluklari almiyor.
```

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

```
    gets(s);
```

```
    printf("cumle: ");
```

```
    puts(s);
```

```
    return 0;
```

```
}
```

# gets fonksiyonunun hatası

```
#include <stdio.h>
int main() {
    int a = 10;
    char yazi[8];
    int b = 20;

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

    printf("programi 8 veya daha fazla karakter girerek test edin\n");
    printf("yazi girin: ");
    gets(yazi);

    // scanf'de de ayni problem var
    // scanf("%s", yazi);

    printf("girilen kelime: ");
    puts(yazi);

    printf("\na: %d ve b: %d\n", a, b);

    return 0;
}
```

# gets fonksiyonu!!!

ISO C11 (2011) standartında gets fonksiyonu kaldırıldı. Derste kullanılan derleyici C99 (1999) standartını kullanmaktadır.

Gets fonksiyonu güvenlik açığı (buffer overflow) oluşturmaktadır.

Büyük bir string girerek programın akısını bozulabilmektedir.

string yerine makine kodu girilirse, programın içindeki verilere hatta bilgisayarın dosya sistemine erişime bile sebep olabilir.

Buffer overflow problemi scanf %s ile de ortaya çıkabilir. Dolayısıyla eleman sayısına göre okuma yapılmalıdır.

Örneğin string 50 elemanlı ise scanf("%49s", str); şeklinde kullanılmalıdır.

# gets yerine fgets

## Kullanılışı;

```
char *fgets(char *str, int n, FILE *stream)
```

## Parametreler;

**\*str:** Okunan string'in saklandığı yeri tutan isaretçi

**n:** Okunacak maksimum karakter sayısı (son bos karakter de dahil)

**\*stream:** String'in okundugu yeri tutan isaretçi

# gets yerine fgets

## Kullanılışı;

`char *fgets(char *str, int n, FILE *stream)`

## Geri dönüş değeri;

islem basarılı ise, str isaretçisi döner

Dosya Sonu ile karşılaşıldığında veya hiçbir karakter okunmazsa, str isaretçisinin içeriği değişmeden kalır ve bos isaretçisi (Null Pointer) döndürülür.

Bir hata olursa, bos isaretçi(Null Pointer) döner.

# gets fonksiyonunun alternatifi

*Örnek:*

```
fgets(s, 8, stdin);  
s[strlen(-1)]='\0'; // sondaki \n yerine \0 ata
```

*Uyarı: fgets fonksiyonu kelimenin sonunda enter ( \n ) karakteri bırakır. Bundan dolayı fgets'den hemen sonra sondaki \n karakterinin yerine \0'a atanmalıdır.*



# gets fonksiyonunun alternatifi

```
#include <stdio.h>
int main() {
    int a = 10;
    char s[8]; // 7 karakter + 1 sonlandirma karakteri sigabilir
    int b = 20;
    printf("a: %d ve b: %d\n\n", a, b);
    printf("programi 8 veya daha fazla karakter girerek test edin\n");
    printf("kelime girin: ");
    fgets(s, 8, stdin); // gets yerine bu sekilde kullanabiliriz
    // scanf ile kelime okumak istersek
    // scanf("%7s", s);
    printf("girilen kelime: ");
    puts(s);
    printf("\na: %d ve b: %d\n", a, b);
    return 0;
}
```

# string içinde karakter arama

*Uyarı: Bu işlemi yapan hazır fonksiyon C kütüphanesinde vardır.*

```
#include <stdio.h>
int main() {
    char cumle[100];
    printf("bir cumle girin: ");
    fgets(cumle, 100, stdin); // gets yerine bu şekilde kullanabiliriz
    // gets(cumle);
    printf("cumlenin uzunlugu: %d\n", karakter_sayisi(cumle));
    int yer = karakter_ara(cumle, 'a');
    if (yer == -1)
        printf("cumlede a karakteri yok\n");
    else
        printf("cumlede a karakteri %d. indexte var\n", yer);
    return 0;
}
```

# string içinde karakter arama

```
int karakter_sayisi(const char *s) {  
    int i;  
    for (i = 0 ; s[i] != '\0' ; i++) {  
        /* islem yapmadan say */  
    }  
    return i;  
}  
  
int karakter_ara(const char *s, char c) {  
    int i;  
    for (i = 0 ; s[i] != '\0' ; i++) {  
        if (s[i] == c)  
            return i;  
    }  
    return -1;  
}
```

# string birlestirme

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

    char cumle_1[100];
    char cumle_2[50];

    printf("bir cumle girin: ");
    fgets(cumle_1, 50, stdin);

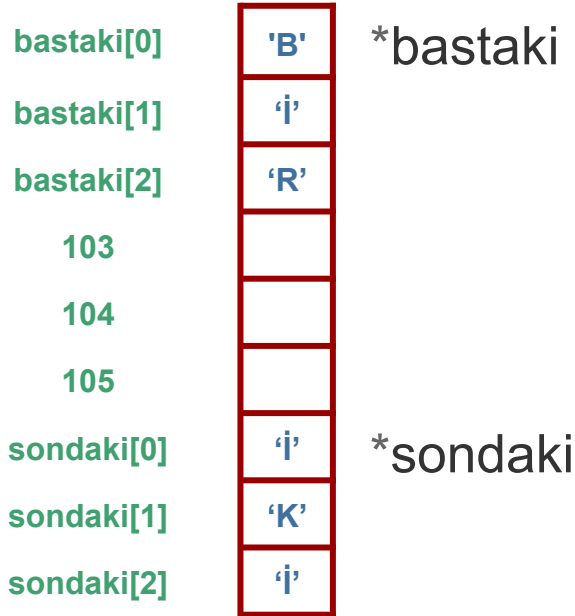
    printf("eklenecek cumle girin: ");
    fgets(cumle_2, 50, stdin);
    // cumle_1'in sonuna cumle_2'yi ekle

    string_ekle(cumle_1, cumle_2);
    printf("ikinci cumle, birinci cumleye eklendi.\n\n");
    puts(cumle_1);
    return 0;
}
```

# string birleştirme

```
int karakter_sayisi(const char *s) {  
    int i;  
    for (i = 0 ; s[i] != '\0' ; i++) {  
        /* islem yapmadan say */  
    }  
    return i;  
}  
  
// bastaki isimli stringe, sondaki isimli stringi ekler  
void string_ekle(char *bastaki, const char *sondaki) {  
    int bastaki_N = karakter_sayisi(bastaki);  
    int sondaki_N = karakter_sayisi(sondaki);  
  
}
```

# string birleştirme



```
char cumle_1[3] = {'B', 'İ', 'R'}
```

```
char cumle_2[3] = {'İ', 'K', 'İ'}
```

```
int bastaki_N = karakter_sayisi(bastaki);
```

```
int sondaki_N = karakter_sayisi(sondaki);
```

# string birleştirme

bastaki[0]

bastaki[1]

bastaki[2]

103

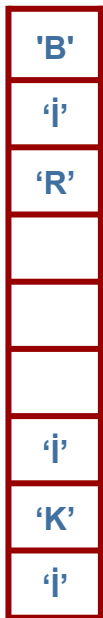
104

105

sondaki[0]

sondaki[1]

sondaki[2]



\*bastaki

\*sondaki

```
int bastaki_N = 3
```

```
int sondaki_N = 3
```

```
for (i = 0 ; i < sondaki_N ; i++)  
    bastaki[i+bastaki_N] = sondaki[i];
```

```
for (i = 0 ; i < 3 ; i++)  
    bastaki[i+3] = sondaki[i];
```

# string birleştirme

<code>bastaki[0]</code>	'B'	*bastaki
<code>bastaki[1]</code>	'İ'	
<code>bastaki[2]</code>	'R'	
<code>bastaki[3]</code>	'İ'	
<code>bastaki[4]</code>	'K'	
<code>bastaki[5]</code>	'İ'	
<code>bastaki[6]</code>	'\0'	
107		
108		

`bastaki[bastaki_N+sondaki_N] = '\0';`

`bastaki[3+3] = '\0';`



# string birleştirme

```
int karakter_sayisi(const char *s) {  
    int i;  
    for (i = 0 ; s[i] != '\0' ; i++) {  
        /* islem yapmadan say */  
    }  
    return i;  
}  
  
// bastaki isimli stringe, sondaki isimli stringi ekler  
void string_ekle(char *bastaki, const char *sondaki) {  
    int bastaki_N = karakter_sayisi(bastaki);  
    int sondaki_N = karakter_sayisi(sondaki);  
  
    int i;  
    for (i = 0 ; i < sondaki_N ; i++)  
        bastaki[i+bastaki_N] = sondaki[i];  
    bastaki[bastaki_N+sondaki_N] = '\0';  
}
```

# string içerisinde string arama

```
#include <stdio.h>
int main() {
    char cumle[100];
    char aranan[20];
    printf("bir cumle girin: ");
    fgets(cumle, 100, stdin); // gets yerine bu şekilde kullanabiliriz
    // gets(cumle);
    printf("aranan kelime: ");
    scanf("%20s", aranan);
    printf("cumlenin uzunlugu: %d\n", karakter_sayisi(cumle));
    printf("aranan kelimenin uzunlugu: %d\n", karakter_sayisi(aranan));
    int var = string_ara(cumle, aranan);
    if (var == 0) {
        printf("aranan kelime cumlede yok\n");
    } else {
        printf("aranan kelime cumlede var\n");
    }
    return 0;
}
```

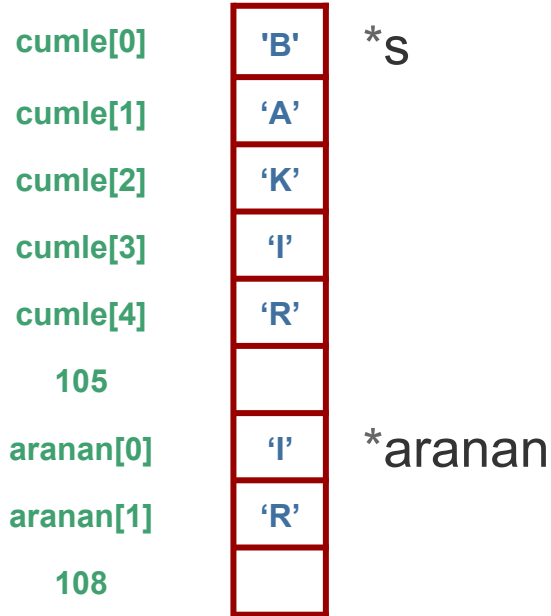
# string içerisinde string arama

```
int karakter_sayisi(const char *s) {  
    int i;  
    for (i = 0 ; s[i] != '\0' ; i++) {  
        /* islem yapmadan say */  
    }  
    return i;  
}
```

# string içerisinde string arama

```
int string_ara(const char *s, const char * aranan) {  
    int i, j;  
    int s_n = karakter_sayisi(s);  
    int aranan_N = karakter_sayisi(aranan);  
  
    return 0;  
}
```

# string içerisinde string arama



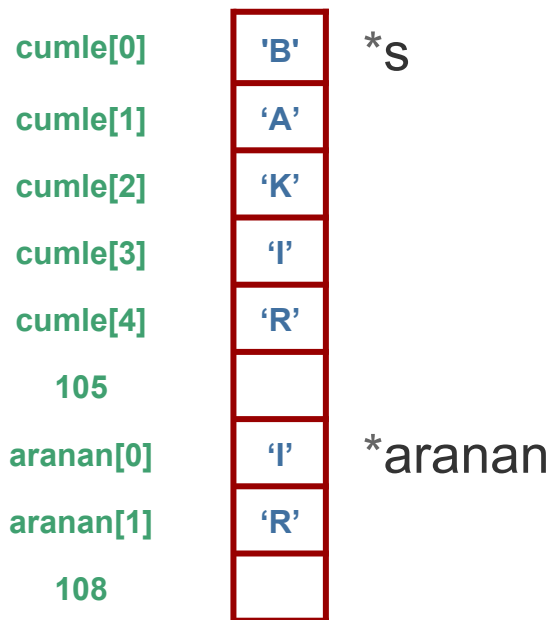
```
char cumle[5] = {'B', 'A', 'K', 'I', 'R'}
```

```
char aranan[2] = {'I', 'R'}
```

```
int s_n = karakter_sayisi(s);
```

```
int aranan_N = karakter_sayisi(aranan);
```

# string içerisinde string arama



```
char cumle[5] = {'B', 'A', 'K', 'I', 'R'}
```

```
char aranan[2] = {'I', 'R'}
```

```
int s_n = 5
```

```
int aranan_N = 2
```

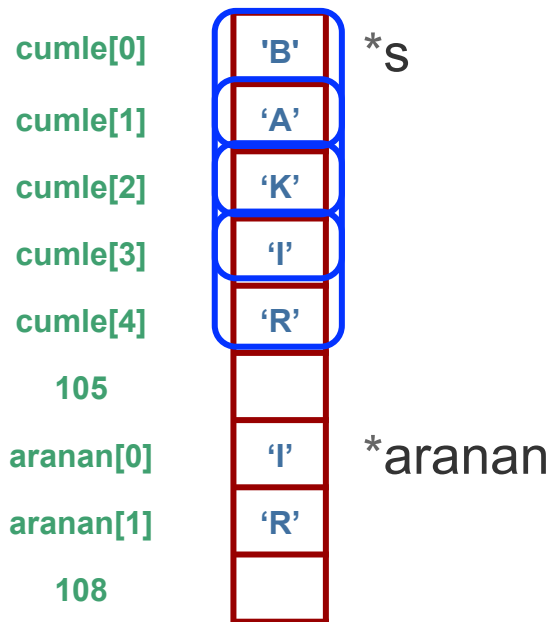
```
for (i = 0 ; i <= s_n-aranan_N ; i++)
```

```
for (i = 0 ; i <= (5-2) ; i++)
```

# string içerisinde string arama

```
int string_ara(const char *s, const char * aranan) {  
    int i, j;  
    int s_n = karakter_sayisi(s);  
    int aranan_N = karakter_sayisi(aranan);  
  
    for (i = 0 ; i <= s_n-aranan_N ; i++) {  
        // i'den basla, aranan_N tanesi ayni mi bak  
  
  
    }  
    return 0;  
}
```

# string içerisinde string arama



```
char cumle[5] = {'B', 'A', 'K', 'I', 'R'}
```

```
char aranan[2] = {'I', 'R'}
```

```
for (i = 0 ; i <= 3 ; i++) {  
    for (j = 0 ; j < 2 ; j++) {  
        if (s[i+j] != aranan[j])  
            break;  
    }  
}
```



# string içerisinde string arama

```
int string_ara(const char *s, const char * aranan) {
    int i, j;
    int s_n = karakter_sayisi(s);
    int aranan_N = karakter_sayisi(aranan);

    for (i = 0 ; i <= s_n-aranan_N ; i++) {

        for (j = 0 ; j < aranan_N ; j++) {
            if (s[i+j] != aranan[j])
                break;
        }
        // eger j == aranan_N olduysa: j dongusu bitmis ve aranan_N tanesi aynidir
        if (j == aranan_N)
            return 1; // return var
    }
    return 0;
}
```

# Sorular

