



# BİLGİSAYAR PROGRAMLAMA 1

## Ders Notu 11 – Sözce(String)

Konya Teknik Üniversitesi  
Elektrik – Elektronik Mühendisliği Bölümü

**16.05.2024**

Konya

---

# Sözce (String)

- Bazı Kaynaklarda Karakter Dizileri ya da Katar isimleriyle de tanımlanmaktadır.
- Sözcelerin normal dizilerden farkı yoktur. Sadece elemanlar char tipinde saklanır.
- Bazı programlama dillerinde sözceleri tutmak için özel veri türleri (string, vb.) bulunmaktadır (Java, C# vs...).
- Ancak C programlama dilinde böyle bir veri türü olmadığı için karakterlerden oluşan bir boyutlu diziler kullanılır.
- Karakter dizilerine özel olarak, karakter dizilerinin sonuna sonlandırıcı karakter olarak adlandırılan bir simge eklenir.
- Sonlandırıcı karakter:
  - Dizin bittiği yeri gösterir.
  - ASCII tablosunun sıfır numaralı ('\0') karakteridir.



# Sözce (String)

- Karakter dizilerine 2 şekilde başlangıç değeri verilebilir:
  - (1) `char selam[6]="Hello";`
  - (2) `char selam[6]={'H', 'e', 'l', 'l', 'o', '\0'};`
- İkinci tanımlamada sonlandırıcı karakter (NULL) programcı tarafından konmalıdır.
- Birinci tanımlamada ise buna gerek yoktur. Çünkü, sonlandırıcı karakter bu atamayla, derleyici tarafından eklenir.

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

# Sözce (String)

- Örnek 1

```
1 #include<stdio.h>
2 main() {
3     char selam [6]="Hello";
4     printf("Merhaba Mesaji: %s\n", selam );
5     return 0; }
6
```

```
Merhaba Mesaji: Hello
-----
Process exited after 0.00986
Press any key to continue .
```

```
1 #include<stdio.h>
2 main() {
3     char selam[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
4     printf("Merhaba Mesaji 2: %s\n", selam );
5     return 0; }
6
```

```
Merhaba Mesaji 2: Hello
-----
Process exited after 0.01056
Press any key to continue .
```



# puts() ve gets() Fonksiyonları

- Bir karakter dizisini klavyeden okumak için C'nin standart gets() fonksiyonu kullanılır.
- puts() fonksiyonu da kendisine argüman olarak verilen karakter dizisini ekrana yazdırır.
- Bu fonksiyonlar herhangi bir indeks tanımlamadan karakter dizilerinin okunmasını sağlar.
- Okuduğu karakter dizisinin sonuna satır sonu işaretini değil, NULL değerini yerleştirir.
- Örnek 2:

```
ders11.2.c
1  #include<stdio.h>
2  main()
3      char kelime[10];
4      puts("Bir kelime giriniz:\n");
5      gets(kelime);
6      puts("\n Asagidaki kelimeyi girdiniz:\n");
7      puts(kelime);
8      return 0;
9
10
```

```
C:\Users\Kemal\Desktop\Google Drive\masaustu\ders
Bir kelime giriniz:
merhaba
    Asagidaki kelimeyi girdiniz:
merhaba
-----
Process exited after 4.957 seconds with
Press any key to continue . . .
```

# Sözce (String)

- Örnek 3:
- scanf sadece tek kelimeyi alır. printf sözcenin tamamını yazar
- puts yazdırmayı bitirdikten sonra bir alt satıra geçer (\n'e gerek yok)

ders11.3.c

```
1  #include<stdio.h>
2  main() {
3      char kelime[20];
4      printf("Bir kelime giriniz:\n");
5      scanf("%s", kelime);
6      printf("\n Asagidaki kelimeyi girdiniz:\n");
7      printf("%s", kelime);
8      return 0;
9  }
```

C:\Users\Kema\Desktop\Google Drive\masaust

```
Bir kelime giriniz:
Merhaba Dünya

Asagidaki kelimeyi girdiniz:
Merhaba

-----
Process exited after 9.443 seconds
Press any key to continue . . .
```

ders11.2.c

```
1  #include<stdio.h>
2  main() {
3      char kelime[20];
4      puts("Bir kelime giriniz:\n");
5      gets(kelime);
6      puts("\n Asagidaki kelimeyi girdiniz:\n");
7      puts(kelime);
8      return 0;
9  }
```

C:\Users\Kema\Desktop\Google Drive\masaust

```
Bir kelime giriniz:
Merhaba Dünya

Asagidaki kelimeyi girdiniz:
Merhaba Dünya

-----
Process exited after 8.634 seconds
Press any key to continue . . .
```



# Sözce (String) Özel Fonksiyonları

- `strlen()` fonksiyonu: Sözcenin uzunluğunu döndürür.
- Dönüş tipi `unsigned long` (işaretsiz tamsayı)'dır.
- Sözce sonunda bulunan `NULL` karakterini hesaba katmaz.
- Örneğin, karakter dizisi `"abc"` değerlerini içeriyorsa, `strlen()` fonksiyonu bu uzunluk olarak `"3"` değerini döndürür.
- Bu tip özel fonksiyonları kullanmak için `<string.h>` kütüphanesi programa eklenmelidir.

# strlen() Örnek

- Örnek 4:

ders11.4.c

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char dizi[50];
5      int dizi_uzunluk,i=0;
6      printf("Bir metin giriniz:\n");
7      gets(dizi);
8      printf("\n Tek tek karakterler yazdiriliyor:\n");
9      dizi_uzunluk=strlen(dizi);
10     for(i=0;i<dizi_uzunluk;i++)
11     {printf("%c\n",dizi[i]);
12     }
13     printf("Tum Karakterler Pespese Yazdiriliyor:\n");
14     printf("%s", dizi);
15     return 0;
16 }
17
```

C:\Users\Kema\Desktop\Google Drive\masaustu\ders icin ornek

Bir metin giriniz:

Merhaba Dunyali

Tek tek karakterler yazdiriliyor:

M  
e  
r  
h  
a  
b  
a  
  
D  
u  
n  
y  
a  
l  
i

Tum Karakterler Pespese Yazdiriliyor:

Merhaba Dunyali

-----  
Process exited after 9.078 seconds with return



# strcpy() fonksiyonu

- strcpy() fonksiyonu: Bir sözceyi başka bir sözceye kopyalamaya yarar.
- Örnek 5:

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char dizi1[50]="Merhaba";
5      char dizi2[50];
6
7      strcpy(dizi2, dizi1);
8
9      printf("dizi1: %s\n",dizi1);
10     printf("dizi2: %s\n",dizi2);
11
12     return 0;
13 }
14
```

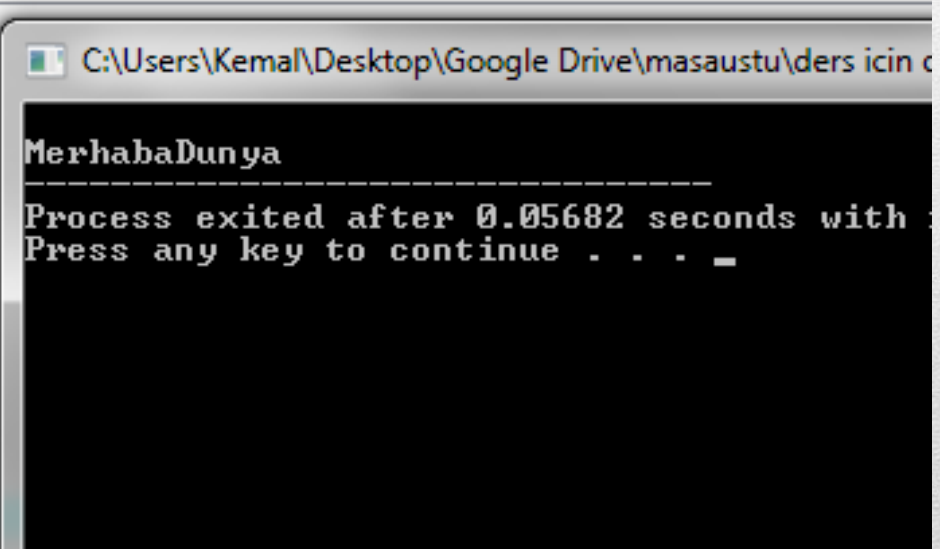
```
dizi1: Merhaba
dizi2: Merhaba
```

```
-----
Process exited after 0.2757 seconds with return va
Press any key to continue . . . _
```

# strcat() fonksiyonu

- strcat() fonksiyonu: Bir sözcüğü başka bir sözcenin sonuna ekleyip bunları birleştirmeye yarar.
- Örnek 6:

```
ders11.6.c
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char dizi1[50]="Merhaba";
5      char dizi2[50]="Dunya";
6
7      strcat(dizi1, dizi2);
8
9      printf("\n%s",dizi1);
10
11     return 0;
12 }
13
```





# strcmp() fonksiyonu

- strcmp() fonksiyonu: İki sözcüyü alfabetik olarak karşılaştırmaya yarar.
- Eğer kendisine gelen ilk argüman alfabetik olarak 2. argümandan önce geliyorsa, 0'dan küçük bir değer döndürür.
- Eğer ilk argüman alfabetik olarak diğerinden sonra geliyorsa 0'dan büyük bir değer döndürür.
- Eğer iki sözcedeki elemanlar eşitse, 0 değerini döndürür
- Örnek 7:

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char cevap1[50]="Ankara";
5      char cevap2[50];
6      int sonuc;
7
8      printf("\nTürkiyenin başkenti neresidir?\n");
9      gets(cevap2);
10     sonuc=strcmp(cevap1,cevap2);
11     if (sonuc==0)
12         printf("Tebrikler! Doğru Cevap...\n");
13     else
14         printf("Malesef Yanlış Cevap...\n");
15
16     return 0;
17 }
18
```

```
Türkiyenin başkenti neresidir?
Ankara
Tebrikler! Doğru Cevap...
```

```
-----
Process exited after 5.607 seconds with
Press any key to continue . . . _
```

# strcmp() fonksiyonu

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char cevap1[50]="Ankara";
5      char cevap2[50];
6      int sonuc;
7
8      printf("\nTürkiyenin başkenti neresidir?\n");
9      gets(cevap2);
10     sonuc=strcmp(cevap1,cevap2);
11     if (sonuc==0)
12         printf("Tebrikler! Doğru Cevap...\n");
13     else
14         printf("Malesef Yanlış Cevap...\n");
15
16     printf ("%d",sonuc);
17     return 0;
18 }
19
```

```
Türkiyenin başkenti neresidir?
Konya
Malesef Yanlış Cevap...
-1
-----
Process exited after 10.9 seconds
Press any key to continue . . .
```

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char cevap1[50]="Ankara";
5      char cevap2[50];
6      int sonuc;
7
8      printf("\nTürkiyenin başkenti neresidir?\n");
9      gets(cevap2);
10     sonuc=strcmp(cevap1,cevap2);
11     if (sonuc==0)
12         printf("Tebrikler! Doğru Cevap...\n");
13     else
14         printf("Malesef Yanlış Cevap...\n");
15
16     printf ("%d",sonuc);
17     return 0;
18 }
```

```
Türkiyenin başkenti neresidir?
Ahhh
Malesef Yanlış Cevap...
1
-----
Process exited after 9.026 seconds
Press any key to continue . . .
```



# strncpy() fonksiyonu

- strncpy() fonksiyonu: İkinci sözcenin ilk N sayıdaki karakterini Birinci sözcenin ilk N sayıdaki karakterinin yerine kopyalamaya yarar.
- Fonksiyon içindeki argümanlar yazılırken sözce1, sözce2 ve son olarak da kaçınıcı karaktere kadar kopyalamanın yapılacağı sayı girilmelidir.
- Örnek 8:

ders11.8.c

```
1  #include<stdio.h>
2  #include<string.h>
3  main() {
4      char dizi1[50]="Kutuphane";
5      char dizi2[50]="Kitapci";
6
7      strncpy(dizi1, dizi2, 5);
8
9      printf("dizi1: %s\n",dizi1);
10
11     return 0;
12 }
13
```

C:\Users\Kemal\Desktop\Go

dizi1: Kitaphane

Process exited after 0  
Press any key to conti

# Bazı Özel Fonksiyonlar

- `strncat()` fonksiyonu:
  - Bu fonksiyon argüman olarak gelen ikinci sözcenin, argüman olarak gelen ilk N sayıdaki karakterini kopyalayıp, argüman olarak gelen birinci sözcenin sonuna ekler.
- `strncmp()` fonksiyonu:
  - Argüman olarak aldığı iki sözcenin tamamını karşılaştırmak yerine sadece ilk N karakterini karşılaştırır.
  - Sonuç özellikleri `strcmp()` fonksiyonu ile aynıdır.



# Sözce Kullanımında Yapılan Bazı Hatalar

- Dizi notasyonu kullanarak tanımladığınız bir karakter dizisine sadece tanımlama esnasında değer atanabilir. Örneğin aşağıdaki gibi bir kullanım hatalıdır:

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

```
kelime="Ankara"; //Yanlış kullanım
```

- Önce sözce tanımlanmış ve sonraki satırda değer ataması yapılmaya çalışılmıştır.
- Bir C dizisi tanımlandığında belli bir bellek alanı ayrılır ve dizi adı bu bellek alanının başlangıç adresini tutar. Dizinin adıyla ulaşılan bellek adresi sabittir ve başka bir bellek adresiyle değiştirilemez. 2. satır yürütülürken "Ankara" değeri bellekte ayrı bir alanda oluşturulur ve bu atama komutuyla birlikte, bu değerın adresinin, sözcenin adresiyle değiştirilmesi söylenmiş olur fakat bu sözcenin adresi başta belirlendiği için değiştirilmesi mümkün değildir.
- Doğru olan:

```
char kelime[20]="Ankara"; //Doğru
```

# Sözce Kullanımında Yapılan Bazı Hatalar

- ```
char sehir[20] = "Ankara";  
char kelime[20];  
kelime=sehir;    //Yanlış kullanım
```
- C programlama dilinde değeri atanmış bir sözcenin değerini değiştirmek için = operatörünü kullanamazsınız.
- Bunun için strcpy fonksiyonu kullanılmalıdır.
- Doğru olan:

```
char sehir[20] = "Ankara";  
char kelime[20];  
strcpy (kelime, "Ankara");    //Doğru
```



# ÖDEV:

Bir sözcedeki boşluklar haricindeki tüm farklı karakterleri bulan ve bunları sıralı bir şekilde ekrana yazdıran bir fonksiyon hazırlayınız.

Örnek Girdi: "merhaba dünya"

Çıktı: a, b, d, e, h, m, n, r, u, y,

# Ö D E V:

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

void farkli_karakterleri_yazdir(const char *metin) {
    int kullanim[256] = {0};

    for (int i = 0; metin[i] != '\0'; i++) {
        if (metin[i] != ' ')
            kullanim[(unsigned char)metin[i]] = 1;
    }

    for (int i = 0, ilk = 1; i < 256; i++) {
        if (kullanim[i]) {
            if (!ilk) printf(", ");
            printf("%c", i);
            ilk = 0;
        }
    }

    printf("\n");
}

int main() {
    farkli_karakterleri_yazdir("merhaba dunya");
    return 0;
}
```