

# String ve Karakter Dizileri

Yrd. Doç. Dr. Fehim KÖYLÜ  
Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü

# #include <iostream>

- Temel giriş çıkış akımları nesnelerinin tanımlandığı kütüphanedir.
- cin,cout,cerr,clog, wcin,wcout,wcerr,wclog nesneleri içerir.
- using namespace std; //std altındaki bütün nesneler doğrudan kullanılır.
- using std::cout; //temel çıkış birimi
- using std::cin; //temel giriş birimi
- char klavyedenOkunanDeger;
- cout<< "Merhaba";
- cin>>klavyedenOkunanDeger;
- Yukardaki ifadeler verilmeyince aşağıdaki gibi kullanılmalıdır.
- std::cout<< "Merhaba";
- std::cin >> klavyedenOkunanDeger;

# Char

- ‘z’ ile ifade edilen char değeri Ascii kodu ile karşılık bulur. ‘z’ sayısal değeri 122’dir.

# String

- C dili String karakter dizisidir ve \0 ile biter. Değişken bir işaretçidir.
- C++ dili string
  - <string> içinde tanımlı sınıf nesnesidir.
  - değişik özellik ve metotlar içerir.

```
char cs[ ] = "Kelime"; // C-string  
string s = "Kelime"; // C++ - string  
  
cout << s << " " << s.length() << " adet ahrf içerir .\n";  
s.replace(5, 2, "i"); // son harfi i yapar.
```

# String C

- “String ifadesi” çift tırnak arası verilen sabit metin bellekte bir işaretcidir.
- Char dizisi ile String ataması:
  - char dizisi sonunda ‘\0’ yer alır.
  - `char renk[] = “beyaz”;`

	0	1	2	3	4	5
renk[]	b	e	y	a	z	\0

# String C

- Benzer atamalar:
  - `char renk2[] = {'b','e','y','a','z','\0'};`
  - `char *renk3="beyaz";`  
→
  - işaretçi b harfinin adresini tutar  
(`renk3=0xA01`)
  - string dizisinin bittiğini  
nul karakteri (`\0`) belirtir.

	.
	.
	.
0xA00	.
0xA01	b
0xA02	e
0xA03	y
0xA04	a
0xA05	z
0xA06	\0
0xA07	.

# Temel G/Ç ile String Okuma/Yazma C

- `char kelime[20];`
- `cin>>kelime;`
- okunan değeri boşluk veya EOF karakterine kadar kelime değişkenine aktarır.
- Okunacak kelime max 19 harf alabilir. Son harf `\0` olacağı için.

# cin.getline

- `cin.getline(dizi, boyut, ayraç);`
- `char cumle[101];`
- `cin.getline(cumle, 101, '\n');`
- Okunacak satır max 100 harf olabilir. (Dizideki son karakter `\0` olacağı için.)
- Enter tuşu (`\n`) görülünceye kadar karakter okunarak `cumle` değişkenine aktarılır.



# String C++

```
int main(int argc, char const *argv[]) {  
    string s1;  
    string s2("Test");  
    string s3 = "kelime1 kelime2 kelime3";  
    string s4(8, '.');  
    string s5(s3, 8, 7);  
  
    cout<<s2<<" " <<s3<<" " <<s4<<" " <<s5;  
  
    cout<<endl;  
    return(0);  
}
```

- Ekran çıktısı:

```
Test kelime1 kelime2 kelime3 ..... kelime2
```

# String C++

- C++ string nesneleri < > == ile karşılaştırılır.
- if(string1>string2){ ....
- + ve += operatörleri ile metin ekleme yapılabilir.
- string sehir = "Kayseri";
- string ilce = "Melikgazi";
- string adres = ilce + "/" +sehir;
- string ilkUcharf= sehir.substr(0,3);
- s.erase(basl,adet); //basl itibaren adet karakteri sil.
- s.replace(2,4,"123"); //2.indisten itibaren 4 karakteri sil yerine123 ilave et.
- int ayraclndisi = adres.find("/");

# String C++

- atama işlemi
- `string s1 = "Erciyes";`
- `string s2 = s1;`
- `string s3.assign(s1);`
- s2 ve s3 ayrı birer değişkendir.
- `string s4.assign(s1,4,3); //yes`
- `s2[0]=s4[2]; //tek değer erişimi ve ataması`

# String C++

- ilave işlemi
- `string s1 = "Erciyes";`
- `s1.append(" Üniversitesi");`
- ya da
- `s1 += " Üniversitesi";`
- `s1.append(baskaStr,baslangic, adet);`
- `s1.substr(start, N);` //N karakter başlangıçtan itibaren
- `s1.swap(s2);` //değişken değerlerini değiştirir.

# String C++

- işlemler == != < > <= >=
- karşılaştırma işlemleri için işaretler haricinde compare metotları kullanılır.
- s1.compare(s2);
- s1.compare(start,length,s2,start,length);
- s1.compare(start, length, s2)

# String C++

- `s1.size()` veya `s1.length()`; // karakter sayısı
- `s1.capacity()`; // yeniden yerleştirme yapmadan tutulabilecek karakter sayısı
- `s1.max_size()`; // olabilecek en büyük boyut
- `s1.resize(yeniUzunluk)`; // verilen uzunluğa ayarlar.

# String C++

- `s1.size()` veya `s1.length()`; // karakter sayısı
- `s1.capacity()`; // yeniden yerleştirme yapmadan tutulabilecek karakter sayısı
- `s1.max_size()`; // olabilecek en büyük boyut
- `s1.resize(yeniUzunluk)`; // verilen uzunluğa ayarlar.

# String C++

- arama metotları
- sonuç olarak indisi döndürür.
- `s1.find(s2);`
- `s1.rfind(s2);` //sağdan sola arama
- `s1.find_first_of(s2);` //s2 içindeki karakterden herhangi birinin ilk karşılaştığı indis döndürür.
- `s1.find_last_of(s2);` //s2 karakterlerinden herhangi birinin s1 içinde son geçtiği indis
- `s1.find_first_not_of(s2);` // s2 içinde yer almayan s1 stringinin ilk karakter indisi
- `s1.find_last_not_of(s2);` // s2 içinde yer almayan s1 in son karakteri indisi



# String C++

- `s1.erase(baslangic); //sona kadar siler.`
- `s1.replace (baslangic, N, s2);`
- `s1.replace(baslangic, N, s2, index ,adet);`
- `s1.insert(indis, s2);`
- `s1.insert(indis, s2,indis, N);`

# C++ String C Char\* Dönüşümü

- `string s = "Bu bir cümledir.";`
- `const char * cstringDegiskeni = s.c_str();`
- `cout<< s.length()<<endl; // length() metodu ile harf sayısı`

# cstring kütüphanesi

```
#include <cstring>
```

C dili char dizisi için kullanılan metotlar :

1. string işleme fonksiyonları
2. string değişkenleri karşılaştırma
3. karakter veya string arama
4. birim string değerlere parçalama

# cstring metotları

Dönüş tipi	metot adı	parametreler	açıklama
char*	strtok	(char *s1, const char* s2)	s1 metnini parçalara ayırır. s2 metninin içindeki karakterleri ayraç olarak kullanır. Her parçalama sonrası kelimeyi işaretçi olarak gönderir. Aynı komut sonraki çağrılarda ilk parametre null verilir. geri dönüşü null olduğunda parçalama bitmiştir.
int	strncmp	(const char *s1, const char* s2, size_t n)	s1 'in n karakterini ve s2 içindeki metni karşılaştırır. s1=s2 ise 0, s1<s2 ise negatif s1>s2 ise pozitif döndürür.
int	strlen	(const char *s)	s metninin karakter sayısını verir.

# cstring metotları -2

Dönüş tipi	metot adı	parametreler	açıklama
char*	strcpy	(char *s1, const char* s2)	s2 sabiti içindeki karakterleri s1 içine kopyalar. s1 işaretçisini geri döndürür.
	strncpy	(char *s1, const char* s2, size_t n )	s2 sabiti içindeki n adet karakterleri s1 içine kopyalar. s1 işaretçisini geri döndürür.
	strcat	(char *s1, const char* s2)	s2 sabiti içindeki karakterleri s1 içindeki metnin sonuna ilave eder. s1 işaretçisini geri döndürür. s1 içindeki null (\0) karakteri ötelenir, ilave edilenlerin sonuna konulur.
	strncat	(char *s1, const char* s2, size_t n )	s2 sabiti içindeki n adet karakterleri s1 içindeki metnin sonuna ilave eder. s1 işaretçisini geri döndürür.
int	strcmp	(const char *s1, const char* s2)	s1 ve s2 içindeki metni karşılaştırır. s1=s2 ise 0 s1<s2 ise negatif s1>s2 ise pozitif döndürür.

# cstring metotları -3

Dönüş tipi	metot adı	parametreler	açıklama
const char*	strchr	(const char *s1, int karakter)	verilen karakterin ilk bulunduğu pozisyonun işaretçisini döndürür.
	strchr	(char *s1, int karakter)	
size_t	strcspn	(const char* str1, const char* keys);	str1 içinde karakterleri tarayarak keys ile verilen harflerden ilk rastladığı pozisyonun indisini geri döndürür.
const char*	strpbrk	(const char *s1, const char *s2)	s2 içinde geçen harflerden birinin s1 içinde ilk karşılaştığı işaretçisini döndürür.
char*	strpbrk	(char *s1, const char *s2)	

# Örnekler

```
int main ()
{
    char str[] = "fcba73";
    char keys[] = "1234567890";
    int i;
    i = strcspn (str,keys);
    printf ("The first number in str is at position %d.\n",i+1);
    return 0;
}
```

cümle içindeki ilk rakamı  
bulan strcspn örneği

```
int main ()
{
    char key[] = "apple";
    char buffer[80];
    do {
        printf ("Guess my favorite fruit? ");
        fflush (stdout);
        scanf ("%79s",buffer);
    } while (strcmp (key,buffer) != 0);
    puts ("Correct answer!");
    return 0;
}
```

string eşleştirmesi  
strcmp örneği

```
int main ()
{
    char str[] = "This is a sample string";
    char key[] = "aeiou";
    char * pch;
    printf ("Vowels in '%s': ",str);
    pch = strpbrk (str, key);
    while (pch != NULL)
    {
        printf ("%c " , *pch);
        pch = strpbrk (pch+1,key);
    }
    printf ("\n");
    return 0;
}
```

sesli harfleri bulan  
strpbrk örneği

\*<http://www.cplusplus.com/reference/cstring/strpbrk/>

# String stream işlemleri

- `#include <sstream>`
- `#include <iostream>`
- Bellek üzerinden string G/Ç işlemleri
- `istringstream` //string üzerinden giriş sınıfı
- `ostringstream` // string'e çıkış sınıfı
- `stringstream(stringDegisken);`
- `istringstream(stringDegisken);`
- `ostringstream(stringDegisken);`



# String stream işlemleri

```
int main(int argc, char const *argv[]) {  
    ostringstream oss;  
    int n = 44;  
    float x = 3.14;  
    oss << "Merhaba!\t" << n << "\t" << x;  
    string s = oss.str();  
    cout << endl << s << endl;  
    return(0);  
}
```

# String stream işlemleri

```
const string buffer = oss.str();  
istringstream iss(buffer);
```

```
string kelime;  
int m;  
float y;
```

```
iss >> kelime >> m >> y;
```

```
s = iss.str();  
cout << endl << s << endl;
```

```
cout << "kelime = " << kelime << endl;  
cout << "m = " << m << endl;  
cout << "y = " << y << endl;
```