

Programlamaya Giriş

Diziler & Karakter Katarı (String)

Arş. Gör. Dr. Gülüzar ÇİT

Bilgisayar ve Bilişim Bilimleri Fakültesi

Bilgisayar Mühendisliği Bölümü

gulizar@sakarya.edu.tr

Konu & İçerik

- + Diziler, Karakter Katarları (Strings)
 - + Dizi Tanımlama
 - + Rasgele Sayı Üretimi
 - + Çok Boyutlu Diziler
 - + Dizi Elemanı Olarak Yapılar
 - + Karakter Katarı (String)
 - + Karakter Katarı Kopyalama
 - + İki Değişkenin Değerlerini Takas Etme
 - + Tek Karakter İçin Fonksiyonlar
 - + Sorular
 - + Kaynaklar

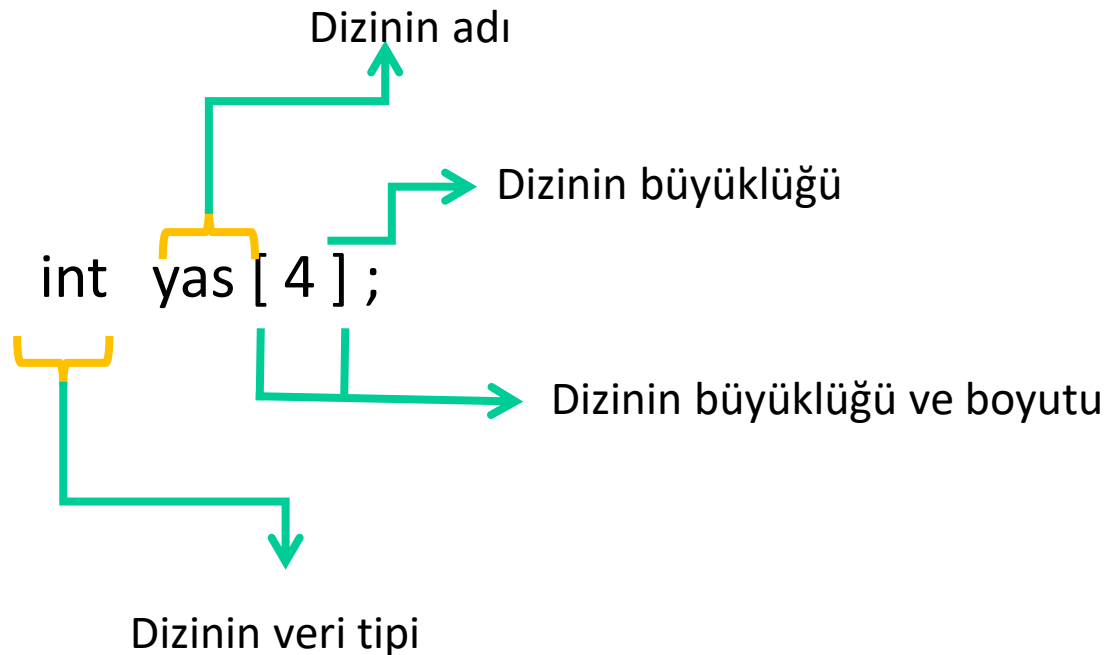


Diziler, Karakter Katarları (Strings)

- ✚ Aynı tipe sahip veri elemanlarının oluşturduğu veri yapılarına (ilişkili veri elemanları topluluğu) **dizi** denir.
- ✚ Yapılar ve sınıflar ilişkili fakat farklı tipe de sahip olabilen verileri tutmak için kullanılırlar.
- ✚ Yapılar, sınıflar ve diziler statik elemanlardır. Programın çalışma süresi boyunca sabit boyuttadırlar.
- ✚ Diziyi oluşturan en küçük birime **eleman** denir.
- ✚ Dizideki her eleman aynı tiptedir.


Diziler, Karakter Katarları (Strings)

Dizi Tanımlama




Diziler, Karakter Katarları (Strings)


Dizi Tanımlama...

-  Eleman erişimi `[]` parantezi arasındaki tamsayı ile belirtilir ve bu sayıya sıra numarası (**index** ya da **indis**) denir.

`yas[2]=3; //dizi elemanına yazma`

`x=yas[3]; //dizi elemanı okuma`

-  Dizi **elemanına** dışarıdan değer atama
`cin >> yas [j];`

-  Dizinin elemanını yazdırma
`cout << j+1 << " ci kisi icin girilen yas " << yas[j];`

-  Dizi, **0** sıra numarası ile başlar

-  Toplam eleman sayısı, maksimum indeksten bir fazladır.

`yas[3]=3; //geçerli`

`yas[4]=3; //geçersiz`

Diziler, Karakter Katarları (Strings)

Dizi Tanımlama...


-  Sıra numarası ardışık ve tamsayıdır.

`yas [1,5]=3; //geçersiz`

-  `[]` içerisinde dört işlem yapılabilir, değişken kullanılabilir.

`int a=2;`

`dizi[a+3]=4;`

-  `[]` parantezinin solunda, sağında veya içinde `++` veya `-` operatörleri kullanılabilir.

`++yas [3]` veya `yas [3]++`

`yas [a++]` veya `yas [++a]`

Diziler, Karakter Katarları (Strings)

Dizi Tanımlama...

 `int c[12];`

 `int sayi[5];`

 `int b[100], x[27];`

 `int n[5] = { 2,12, 13, 4, 5 };`

Eleman sayısından az ise diğerleri 0,
fazla ise hata!

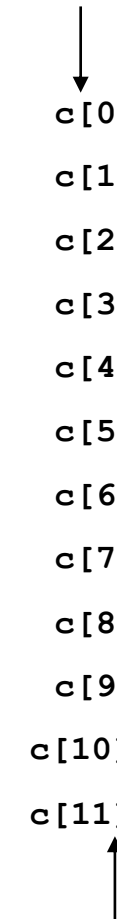
 `int n[5] = { 0 };`

Hepsi 0

 `int n[] = { 7, 21, 35, 14, 5 };`

`n[5]`

`int c[12];`



<code>c[0]</code>	-45
<code>c[1]</code>	6
<code>c[2]</code>	0
<code>c[3]</code>	72
<code>c[4]</code>	1543
<code>c[5]</code>	-89
<code>c[6]</code>	0
<code>c[7]</code>	62
<code>c[8]</code>	-3
<code>c[9]</code>	1
<code>c[10]</code>	6453
<code>c[11]</code>	78

Diziler, Karakter Katarları (Strings)

Dizi Tanımlama...

```
#include <iostream>
#include <iomanip>
//#define ESAY 10

using namespace std;

const int ESAY=20;

int main()
{
    int sayilar[ ESAY ] = {19, 3, 15, 7, 11, 9, 13, 5, 17, 1};

    cout<<setw(5)<<"indis"<<setw(5)<<"Deger"<<endl;

    for ( int i = 0; i < ESAY; i++ )
    {
        cout<<setw(5)<<i<<setw(5)<<sayilar[i]<<endl;
    }
    system("pause");
    return 0;
}
```


Diziler, Karakter Katarları (Strings)

Dizi Tanımlama...

```
#include <iostream>
#include <iomanip>
// #define ESAY 10

using namespace std;

const int ESAY=20;

int main()
{
    int sayilar[ ESAY ] = {19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };

    cout<<setw(5)<<"indis"<<setw(8)<<"Deger"<<setw(20)<<"Grafik"<<endl;

    for ( int i = 0; i < ESAY; i++ )
    {
        cout<<setw(5)<<i<<setw(8)<<sayilar[i]<<setw(15);
        for(int j=0;j<sayilar[i];j++)
            cout<<'*';
        cout<<endl;

    }
    system("pause");
    return 0;
}
```

```
indis  Deger  Grafik
0      19    *****
1      3     ***
2      15    *****
3      7     *****
4      11    *****
5      9     *****
6      13    *****
7      5     *****
8      17    *****
9      1     *
10     0
11     0
12     0
13     0
14     0
15     0
16     0
17     0
18     0
19     0
Devam etmek için bir tuşa basın . . .
```

Diziler, Karakter Katarları (Strings)

Dizi Tanımlama...

```
#include <iostream>
#include <iomanip>
using namespace std;

const int ESAY=100;

int main()
{
    int sayi[ESAY];

    for(int i=0; i<ESAY ; i++)
    {
        sayi[i]=i+1;
    }
    for(int i=0; i<ESAY ; i++)
    {
        cout<<left<<setw(6)<<sayi[i];
        if((i+1)%5==0)
            cout<<endl;
    }
    system("pause");
    return 0;
} // end main
```

Diziler, Karakter Katarları (Strings)

Rasgele Sayı Üretimi

```
#include <iostream>
#include <iomanip>

//#define ESAY 10

using namespace std;

const int ESAY=20;

int main()
{
    cout<<RAND_MAX<<endl;

    int sayilar[ESAY];

    cout<<setw(5)<<"indis"<<setw(5)<<"Deger"<<endl;

    for ( int i = 0; i < ESAY; i++ )
    {
        sayilar[i]=rand()%100;
        cout<<sayilar[i]<<endl;
    }

    system("pause");
    return 0;
}
```

```
32767
indisDeger
41
67
34
0
69
24
78
58
62
64
5
45
81
27
61
91
95
42
27
36
Devam etmek için bir tuşa
```

Diziler, Karakter Katarları (Strings)

Rasgele Sayı Üretimi...

-  Klavyeden girilen sayının frekansını bulan program

SORU: Sayısal loto tahmin programı geliştiriniz....

```
#include <iostream>
#include <iomanip>
#include <ctime> //time()

using namespace std;

int main()
{
    const int ESAY = 25000;
    int dizi[ESAY];
    int sayi,tekrar=0;

    do
    {
        srand( time( 0 ) );
        for ( int i = 0; i < ESAY ; i++ )
        {
            dizi[i] = rand()%11;
            cout<<dizi[i]<<" ";
        }

        do
        {
            cout<< "\nsayi giriniz(0-10 arasi)";
            cin>>sayi;
        }while(!(sayi>0&&sayi<=10));


        for ( int j = 0; j < ESAY ; j++ )
        {
            if(sayi==dizi[j])
                tekrar++;
        }

        cout<<"\n"<<tekrar<<endl;
    }while(sayi>0);

    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Çok Boyutlu Diziler

-  İki boyutlu diziye [0-10] arasında rasgele değer atayan ve bu diziye ekrana yazdıran program

Soru :Öğrenci not hesaplama programı yaz... Eklentiler (Vize ortalaması, final ortalaması, ortalama ortalaması, grafik (100=30 yıldızla ifade edilsin) gösterimi)

```
#include <iostream>
#include <iomanip>
#include <ctime> //time()

using namespace std;

const int SATIR = 6;
const int SUTUN = 5;

int main()
{
    int matris[SATIR][SUTUN];

    srand( time( 0 ) );

    for ( int i = 0; i < SATIR ; i++ )
    {
        for(int j=0;j<SUTUN;j++)
        {
            matris[i][j] = rand()%10+1;
        }
    }

    for ( int i = 0; i < SATIR ; i++ )
    {
        for(int j=0;j<SUTUN;j++)
        {
            cout<<setw(5)<<matris[i][j];
        }
        cout<<endl;
    }
    cout<<endl<<matris[0][0]+matris[5][4];
    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Çok Boyutlu Diziler...

Matris Toplama




```
void matrisTopla(int a[][], int b[][], int c[][])
{
    int satir, sutun;
    for (satir = 0; satir<10; satir++)
        for (sutun = 0; sutun<10; sutun++)
            c[satir][sutun] = a[satir][sutun] + b[satir][sutun];
}
```

Matrisin Transpozmesini Alma

```
void matrisTrnspoz(int a[][10], int b[][10])
{
    int satir, sutun;
    for (satir = 0; satir<10; satir++)
        for (sutun = 0; sutun<10; sutun++)
            b[satir][sutun] = a[sutun][satir];
}
```

Diziler, Karakter Katarları (Strings)

Dizi Elemanı Olarak Yapılar












```
 struct Olcu{  
    int metre;  
    int cmetre;  
};  
  
 Olcu uzunluk[10];  
  
 cin >> uzunluk[2].metre;
```

Soru 1: Klavyeden 0 metre 0 cmetre girilene kadar girilen tüm ölçüleri dizide saklayarak ekrana listeleyen, ardından girilen ölçülerin aritmetik ortalamasını hesaplayan programı yazınız.

Soru 2: Öğrenci bilgilerini saklamak üzere bir tip (struct) tanımlayınız. 5 öğrenciye ait bilgiler için bir dizi tanımlayınız. Bilgi girişi, yazdırma v.s. işlemler yaptırınız...






Diziler, Karakter Katarları (Strings)

Dizilerde Sıralama

-  Bazı uygulamalarda ister sayısal ister alfasayısal bilgilere sahip olsun dizilerdeki bilgilerin belirli bir sırada olması istenir. Örneğin: Küçükten büyüğe, A'dan Z'ye sıralama
-  Sıralama belirli bir anahtar sözcüğe göre verilere sıralı erişimin sağlanmasıdır.
-  Sıralı veriler üzerinde işlem yapmak çok daha hızlı ve kolaydır.
-  Literatürde çok sayıda sıralama algoritması vardır. Bunlardan bazıları :
 -  Basit Sıralama
 -  Seçmeli Sıralama (Selection Sort)
 -  Kabarcık Sıralama (Bubble Sort)
 -  Araya Yerleştirerek Sıralama (Insertion Sort)
 -  Birleşmeli Sıralama (Merge Sort)
 -  Kümelemeli Sıralama (Heap Sort)
 -  Hızlı Sıralama (Quick Sort)

Diziler, Karakter Katarları (Strings)

Seçmeli Sıralama

-  Selection sort algoritması en basit sıralama algoritmalarındandır.
-  Sıralamaya dizideki ilk ya da son elemandan başlanır.
-  Seçilen dizi elemanı ile bu elemandan sonra gelen dizi elemanları sırası ile sıralama isteğine göre (büyüklük/küçüklük) karşılaştırılır.
-  Eğer büyüklük/küçüklük durumu var ise yerleri değiştirilir.
-  Bu karşılaştırma işlemi sondan bir önceki eleman ile son elemanın karşılaştırılmasına kadar devam eder ve aynı şekilde eğer büyüklük/küçüklük durumu var ise yerleri değiştirilir.

Diziler, Karakter Katarları (Strings)

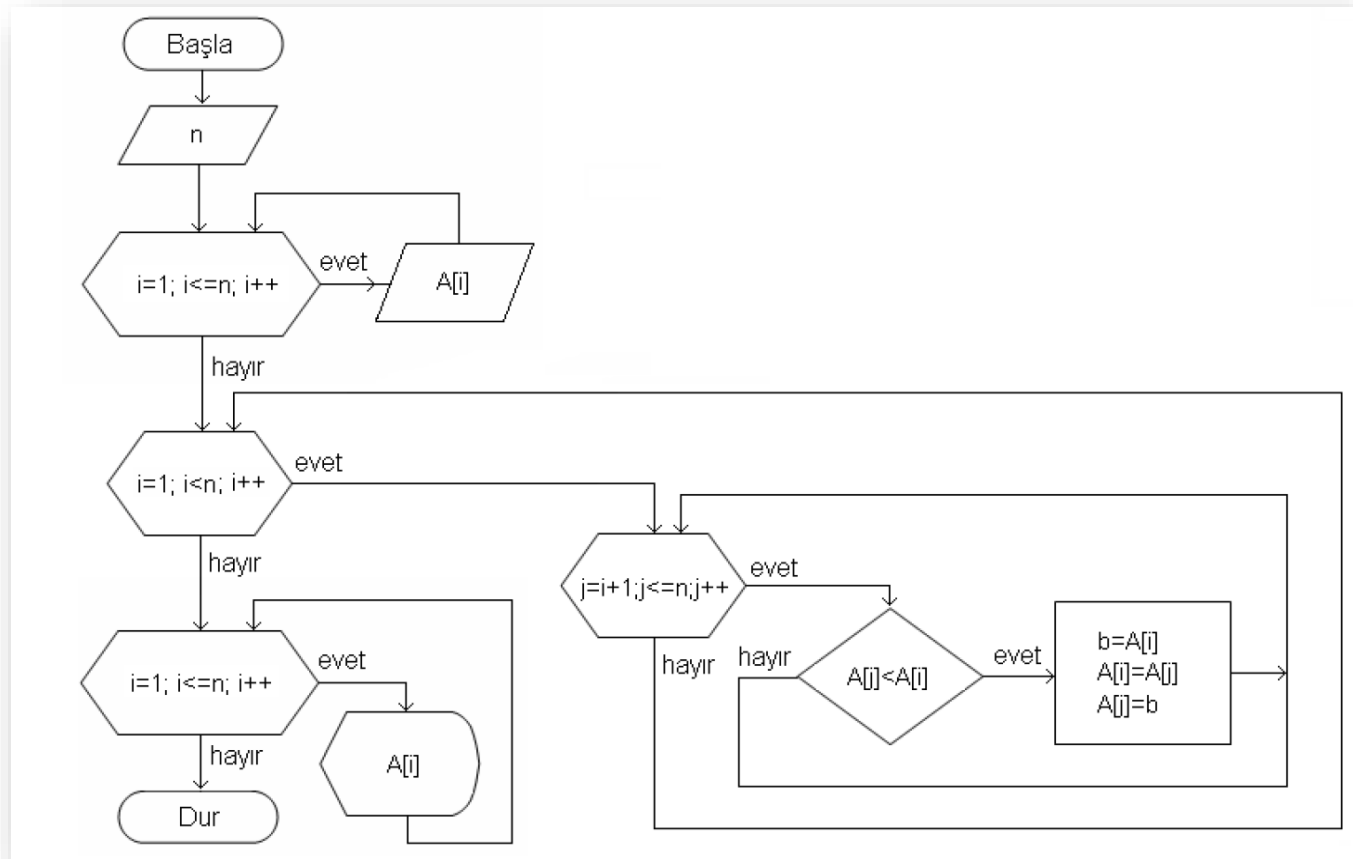
Seçmeli Sıralama...

 Selection sort ile artan bir dizi sıralaması için;

- ① Sıralanmamış dizi elemanları içindeki en küçük değerli elemanı bul
- ② Bulunan bu elemanı (en küçük değerli), sıralanmamış dizinin ilk elemanı ile yer değiştir (swap).
- ③ Sıraya konulmamış ilk elemanın yeri dizinin sonu oluncaya kadar bu işlemi tekrarla. (Kalan elemanlar arasında tekrarla)




Diziler, Karakter Katarları (Strings)

Seçmeli Sıralama...



Diziler, Karakter Katarları (Strings)

Kabarcık Sıralama

-  Bubble sort algoritması yer değiştirmeli sıralama (exchange sort) olarak da bilinir.
-  Kabarcık sıralaması olarak adlandırılmasının nedeni dizi içindeki büyük elemanların algoritmanın her adımında dizi sonuna doğru lineer olarak ilerlemesidir.
-  Kabarcık sıralama da baştan sona doğru yapılan taramalar ile arka arkaya gelen (ardışık) elemanlar kıyaslanır. Eğer küçüklük/büüklük durumu var ise yerleri değiştirilir.

Diziler, Karakter Katarları (Strings)

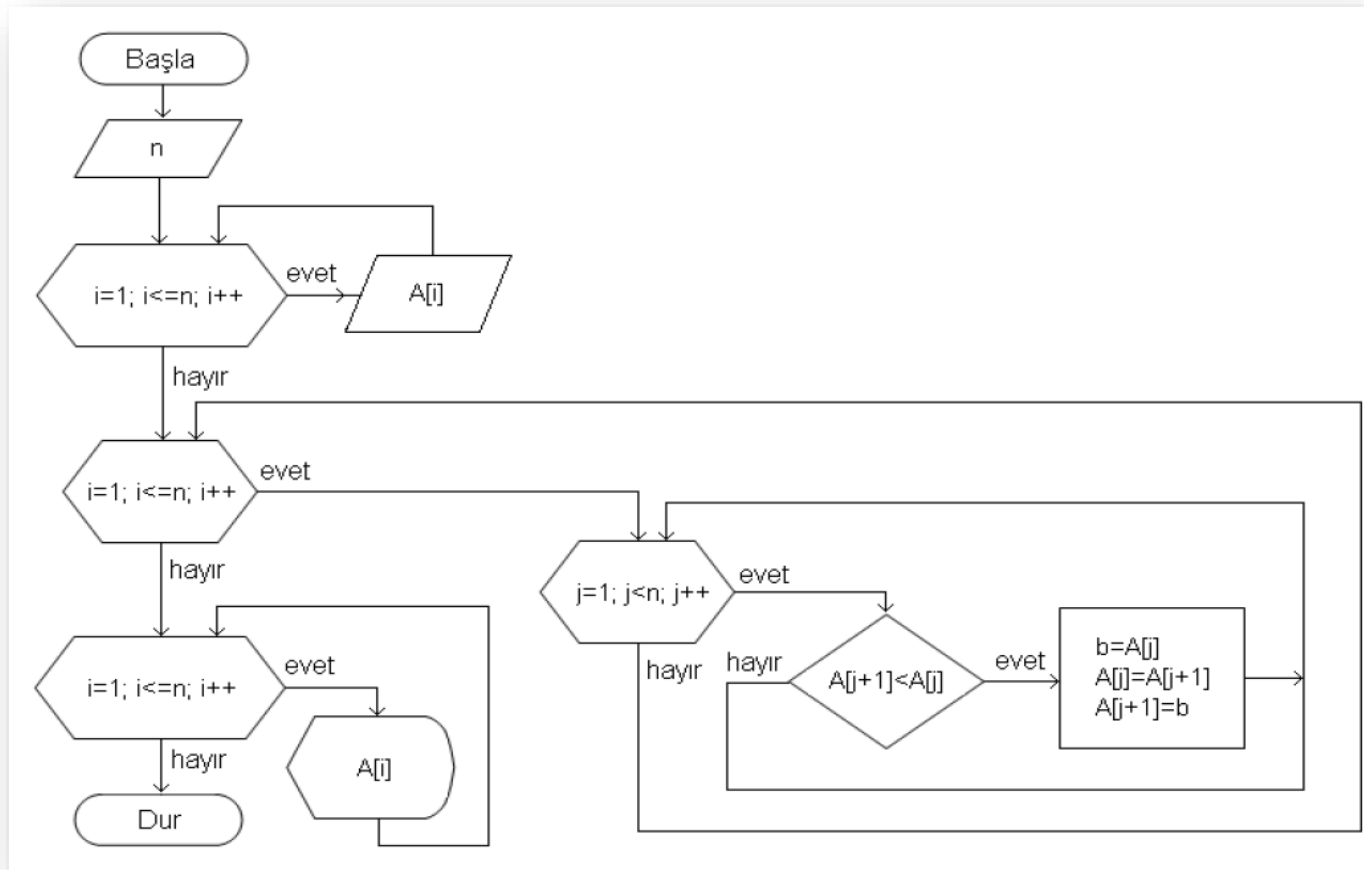
Kabarcık Sıralama...

 Bubble sort ile artan bir dizi sıralaması için;

- ❶ Sıralanmamış dizi elemanlarının her birinin değeri, sağındaki komşusu ile karşılaştırılır.
- ❷ Eğer değeri komşusundan büyük bir elemana raslanırsa, komşusu ile yer değiştirilir (swap). Böylece sıralı olmayan elemanlar her tarandığında, sadece yan yana bulunan iki eleman arasında sıralama yapılmış olur. Ayrıca içlerinden en büyük olanı da en sağa ilerlemiştir ve artık sıraya girmiştir. (sağındaki tüm komşularından büyük olduğu için, sürekli komşusuyla yer değiştirir.)
- ❸ Sıraya konulmamış eleman kalmayana kadar, sıralanmamış elemanların taranma işlemi tekrarlanır

Diziler, Karakter Katarları (Strings)

Kabarcık Sıralama...



Diziler, Karakter Katarları (Strings)

Kabarcık Sıralama...

```
void bubblesort(int b[], int max)
{
    int kez, i, gecici;
    for (kez = 1; kez <= max - 1; kez++)
        for (i = 0; i <= max - 2; i++)
            if (b[i] > b[i + 1])
            {
                gecici = b[i];
                b[i] = b[i + 1];
                b[i + 1] = gecici;
            }
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String) – C String


 `char kdizi[] = { 'H', 'e', 'l', 'l', 'o', '\0' };`

 `char kdizi[] = "Hello";`

 `kdizi[0] -> 'H'`

`cin>>kdizi;`

`cout<<kdizi;`

 Karakter dizilerinin string (katar) olabilmesi için sonunda mutlaka **null** karakteri olmalı. Aksi taktirde mantıksal hatalara neden olabilir.

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String) – C String

```
#include <iostream>
using namespace std;

int main()
{
    const int MAX = 80;
    char str[MAX];

    cout << "Karakter dizisi gir";
    cin >> str; |

    cout << "Girilen karakter dizisi" << str << endl;
    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String) – C String

```
#include <iostream>
using namespace std;

int main()
{
    const int MAX = 80;
    char str[MAX], str2[MAX];

    cout << "Karakter dizisi gir";
    //cin >> str;
    cin.get(str, MAX); // boşlukları da alır

    cout << "Girilen karakter dizisi" << str << endl;

    cout << "Karakter dizisi (çok satır) gir";
    cin.get(str, MAX, '.'); // Çoklu satır
    cout << "Girilen karakter dizisi" << str << endl;

    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Tek Karakter İçin Fonksiyonlar

Fonksiyon	Görevi
<code>isdigit (int c)</code>	<code>c</code> rakamsa TRUE, değilse FALSE döndürür
<code>isalpha (int c)</code>	<code>c</code> harfse TRUE, değilse FALSE döndürür
<code>isalnum (int c)</code>	<code>c</code> harf veya rakamsa TRUE, değilse FALSE döndürür
<code>isxdigit (int c)</code>	<code>c</code> hexadecimal ise TRUE, değilse FALSE döndürür
<code>islower (int c)</code>	<code>c</code> küçük harf ise TRUE, değilse FALSE döndürür
<code>isupper (int c)</code>	<code>c</code> büyük harf ise TRUE, değilse FALSE döndürür
<code>tolower (int c)</code>	Küçük harfe çevir
<code>toupper (int c)</code>	Büyük harfe çevir
<code>isspace (int c)</code>	<code>c</code> boşluk karakteri ise TRUE, değilse FALSE döndürür
<code>isctrl (int c)</code>	<code>c</code> kontrol karakteri ise TRUE, değilse FALSE döndürür

```
//isdigit fonk. Tanımı:
//rakamsa 1, değilse 0
int isdigit(int c)
{
    if ((c >= '0') && (c <= '9'))
        return 1;
    return 0;
}

//tolower fonk. Tanımı:
int tolower(int c)
{
    if (isupper(c))
        c += 32;
    return c;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı Dönüştürme Fonksiyonları

Fonksiyon	Görevi
atof	Dizgiyi double'a dönüştürür
atoi	Dizgiyi int'e dönüştürür
atol	Dizgiyi long int'e dönüştürür
strtod	Dizgiyi double'a dönüştürür
strtol	Dizgiyi long int'e dönüştürür
strtoul	Dizgiyi unsigned int'e dönüştürür

```
int atoi(const char *nptr)
{
    int x = 0, j = 0;
    while (nptr[j])
    {
        x = x * 10 + (nptr[j] - '0');
        j++;
    }
    return x;
}
```

```
cout<<atoi("12")*2;
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı Kopyalama

```
#include <iostream>
#include <cstring> //strlen()
using namespace std;

int main()
{
    char str1[] = "Aynasi istir kisinin lafa bakilmaz!";

    const int MAX = 80;
    char str2[MAX];

    |
    cout<<strlen(str1);
    int j;
    for(j=0; j<strlen(str1); j++)
        str2[j] = str1[j];
    str2[j] = NULL;
    cout << str2 << endl;
    system("pause");

    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı İşleme Fonksiyonları

Fonksiyon	Görevi
<code>strcpy</code>	Bir dizgiyi başka bir dizgiye kopyalar
<code>strncpy</code>	Bir dizginin belirli sayıdaki karakterini başka bir dizgiye kopyalar
<code>strcat</code>	Başka bir dizgiye ekler
<code>strncat</code>	Bir dizginin belirli sayıdaki karakterini başka bir dizgiye ekler

```
void strcpy(char *s1, const char *s2)
{
    int i = 0;
    while (s2[i])
    {
        s1[i] = s2[i];
        i++;
    }
    s1[i] = '\0';
}
```

`strcpy(hedef, kaynak);`

Diziler, Karakter Katarları (Strings)

Karakter Katarı İşleme Fonksiyonları

```
#include <iostream>
//#include <cstring> //strlen()
using namespace std;

int main()
{
    char str1[] = "Aynasi istir kisinin lafa bakilmaz!";

    const int MAX = 80;
    char str2[MAX];

    cout<<strlen(str1);
    int j;
    strcpy(str2, str1);
    cout << str2 << endl;
    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı

```
#include <iostream>
#include <string>

using namespace std;
////////////////////////////////////

int main()
{
    const int DAYS = 7;
    const int MAX = 10;

    char star[DAYS][MAX] = { "Pts", "Sali", "Cars",
                             "Pers", "Cuma", "Cumartesi", "Pazar" };

    //string star[2]={"Pazartesi", "Sali"};
    for(int j=0; j<DAYS; j++)
        cout << star[j] << endl;

    system("pause");

    return 0;
}
```


Diziler, Karakter Katarları (Strings)

Karakter Katarı Karşılaştırma Fonksiyonları

Fonksiyon	Görevi
<code>strcmp(...)</code>	İki karakter dizgisini karşılaştırır. Birincisi büyükse pozitif, ikincisi büyükse negatif, eşitse 0 döndürür
<code>strncmp(...)</code>	İki karakter dizgisini belirli sayıdaki karakterlerini karşılaştırır. Birincisi büyükse pozitif, ikincisi büyükse negatif, eşitse 0 döndürür

```
int strcmp(const char* s1, const char* s2)
{
    int i;
    for (i = 0; s1[i] == s2[i]; i++)
        if (!s1[i])
            return 0;
    return s1[i] - s2[i];
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String) – C String

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    char kopya1[50], kopya2[50], kelime[50];
    int s;
    cout << "Metin Giriniz: ";
    cin >> kelime;
    strcpy(kopya1, kelime);    // kelime katarını kopya1'e kopyala
    cout << kopya1 << endl;
    strncpy(kopya2, kelime, 4); //kelime nin ilk 4 karakterini kopyala
    cout << kopya2 << endl;
    if (strcmp(kopya1, kelime) == 0) // kopya1 ve kelime karsilastir
        cout << "kopyalama tamamdır";
    strcat(kopya1, kelime);    // kelimeyi kopya1'e ekle
    cout << kopya1 << endl;
    system("pause");
    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String) – String Sınıfı

```
string ad2 = "Eray";  
string soyad = "Tamer";  
cin >> ad1; //Boşluktan sonrasını almaz  
getline(cin, ad2); //Boşluğu da okur.  
ad1 = ad2;  
ad1.assign(ad2);    //ad1=ad2  
ad1.append("Mert");  //ad1 sonuna Mert ifadesini ekler  
ad1.compare(ad2);    //ad1 ile ad2 yi karşılaştırır. Aynı ise 0 değerini döndürür.  
cout << ad1.substr(5, 4); //ad1 in 5. karakterinden sonraki 4 karakter  
cout << ad1.find("Me"); //Me ifadesinin ad1 içerisindeki konumu  
ad1.insert(10, soyad); //ad1 in 10. karakterinden sonra soyad nesnesinin değerini ekler.  
cout << ad1.length(); //ad1 in kaç karakter olduğu bilgisini yazdırır.
```

Diziler, Karakter Katarları (Strings)

Tek Karakter İçin Fonksiyonlar

```
#include <iostream>
#include <cctype>

using namespace std;

int main()
{
    string str1= "" ;

    const int MAX = 80;
    char str2[MAX];
    cin>>str1;
    cout<<str1.length()<<" karakter"<<endl;
    int j;

    for(j=0; j<str1.length(); j++)
        str2[j] = toupper(str1[j]);
    str2[j] = NULL;

    cout << str2 << endl;
    system("pause");

    return 0;
}
```

Diziler, Karakter Katarları (Strings)

Karakter Katarı (String)

```
//Klavyeden girilen karakter dizisini tersten yazdırma

#include <iostream>
#include <string>

using namespace std;

int main()
{
    string ad1;

    getline(cin,ad1); //Boşluğu da okur...
    cout<<ad1.length();
    system("pause");
    for(int i=ad1.length()-1;i>=0;i--)
        cout<<ad1[i];
    system("pause");
    return 0;
}
```

Sorular

- ✚ Klavyeden girilen karakter katarını ters çevirerek ekrana yazan (Ali -> ilA gibi) programı yazınız...
- ✚ Klavyeden girilen karakter katarının her bir karakterini:
 - ✚ Tamamen büyük
 - ✚ Tamamen küçük
 - ✚ Büyükse küçük, küçükse büyük (Ali->aLi gibi)
 - ✚ Kelimelerin ilk harfini büyüğe çevir.... (boşluk karakteri kontrol edilmeli...) yazan programı geliştiriniz (switch-case ile menu oluşturun...).
- ✚ Bir önceki sorudaki çevirme işlemlerini ayrı fonksiyonlarla gerçekleştiriniz...
- ✚ Kullanıcı ESC tuşuna basmadığı sürece klavyeden girilen sayıların ortalamasını alan program.. (ESC tuşunun ASCII karşılığı=27, Enter=13...)

Sorular...

- ✚ Kullanıcı noktalama işaretlerine basmadığı sürece rasgele olarak üretilen sayının asal olup olmadığını bulan program...
- ✚ Girilen karakter katarının uzunluğunu döndüren fonksiyonu yazınız... (hazır fonksiyon kullanmayınız...)
- ✚ A ile başlayan en fazla 10 karakter uzunluğunda (uzunluğu rastgele belirlenecek) 20 adet rastgele kelime oluşturunuz... (Kelimelerin anlamlı olması gerekmiyor) rand, append, insert v.b. fonksiyonlardan faydalanabilirsiniz
- ✚ ASCII tablosunu tarayarak sayı karşılığı asal olan harflerden 3 karakterli kelimeler oluşturunuz...

Kaynaklar

- ✚ Robert Lafore, Object Oriented Programming in C++, Macmillan Computer Publishing
- ✚ Deitel, C++ How To Program, Prentice Hall
- ✚ Prof. Dr. Cemil ÖZ, Programlamaya Giriş Ders Notları
- ✚ Prof. Dr. Celal ÇEKEN, Programlamaya Giriş Ders Notları