

Programlamaya Giriş

HAFTA 6

Diziler & Karakter Katarı (String)

Prof. Dr. Cemil ÖZ

Doç. Dr. Cüneyt BAYILMIŞ

Dr. Öğretim Üyesi Gülüzar ÇİT

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
 - Çalışma Soruları
 - Kaynaklar

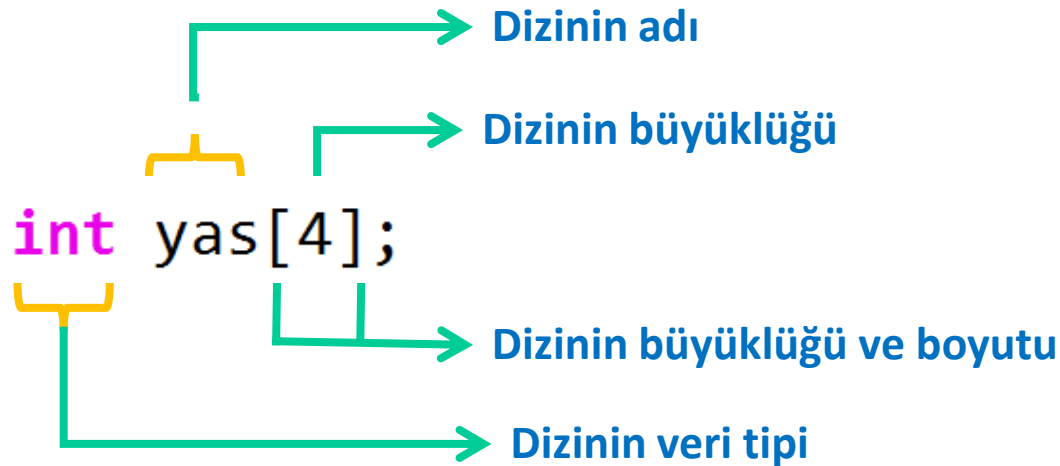


Diziler, Karakter Katarları (Strings)

- 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;  
yas[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];  
//Eleman sayısından az ise diğerleri 0,  
//fazla ise hata!  
int n[5] = { 2,12, 13, 4, 5 };  
int n[5] = { 0 }; //Hepsi 0  
int n[] = { 7, 21, 35, 14, 5 }; // n[5]
```

int c[12];

c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6543
c[11]	78

Diziler, Karakter Katarları (Strings)...

➤ Dizi Tanımlama...

➤ ÖRNEK: ⇒ [1]_dizi.cpp

```
const int ESAY = 20;

int main()
{
    // Sonuç ekranında Türkçe karakterleri kullanabilmek için
    setlocale(LC_ALL, "Turkish");

    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...

➤ ÖRNEK: ⇒ [2]_dizi.cpp

```
const int ESAY = 20;

int main()
{
    int sayi[ESAY];

    for (int i = 0; i < ESAY; i++)
    {
        cin >> sayi[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;
}
```

Diziler, Karakter Katarları (Strings)...

➤ Dizi Tanımlama...

➤ ÖRNEK: ⇒ [3]_dizi_grafik.cpp

```
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)...

➤ Rastgele Sayı Üretimi

➤ ÖRNEK: ⇒ [4]_random.cpp

```
// maksimum rand sayı üretme aralığı [0-32767]
//cout << RAND_MAX << endl;

int sayilar[ESAY];

cout << setw(5) << "İNDİS"
    << setw(10) << "DEĞER" << endl;

// Her çalıştırılmasında farklı random sayılar üretilmesi için
//srand(time(NULL));
for (int i = 0; i < ESAY; i++)
{
    // [0-99] aralığında random sayı üretir
    sayilar[i] = rand() % 100;
    cout << setw(3) << i << setw(10) << sayilar[i] << endl;
}
```

İNDİS	DEĞER
0	41
1	67
2	34
3	0
4	69
5	24
6	78
7	58
8	62
9	64

Press any key to continue . . .

Diziler, Karakter Katarları (Strings)...

➤ Rastgele Sayı Üretimi...

➤ ÖRNEK:

⇒ [5]_random_frekans.cpp

- [0-10] arasında rasgele ürettiği 25000 sayı içinden klavyeden girilen sayının frekansını hesaplayan program



SORU: Sayısal loto tahmin programı geliştiriniz....

```
int dizi[ESAY];
int sayi, tekrar = 0;
char devam;

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

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

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

    cout << "\n" << tekrar << endl;

    do
    {
        cout << "Devam Etmek İstiyor musunuz? [E / H]";
        cin >> devam;
    } while (!(devam == 'e' || devam == 'E' || devam == 'h' || devam == 'H'));
} while (devam == 'e' || devam == 'E');
```

Diziler, Karakter Katarları (Strings)...

➤ Çok Boyutlu Diziler

➤ ÖRNEK: ⇒

[6]_cokboyutlu_dizi.cpp

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

Soru: Klavyeden girilen öğrenci not bilgilerine bağlı olarak bir istatistik programı yaz

- Vize ortalaması
- final ortalaması
- ortalama ortalaması
- grafik gösterimi (100=30 yıldızla ifade edilsin)



```
int matris[SATIR][SUTUN];

srand(time(0));

// Rasgele Değer Üretme
for (int i = 0; i < SATIR; i++)
    for (int j = 0; j < SUTUN; j++)
        matris[i][j] = rand() % 10 + 1;

// Ekrana Yazdırma
for (int i = 0; i < SATIR; i++)
{
    for (int j = 0; j < SUTUN; j++)
        cout << setw(5) << matris[i][j];
    cout << endl;
}

cout << matris[0][0] + matris[5][4];
```

Diziler, Karakter Katarları (Strings)...

➤ Çok Boyutlu Diziler...

➤ Matris Toplama

```
for (int i = 0; i < SATIR; i++)  
    for (int j = 0; j < SUTUN; j++)  
        C[i][j] = A[i][j] + B[i][j];
```

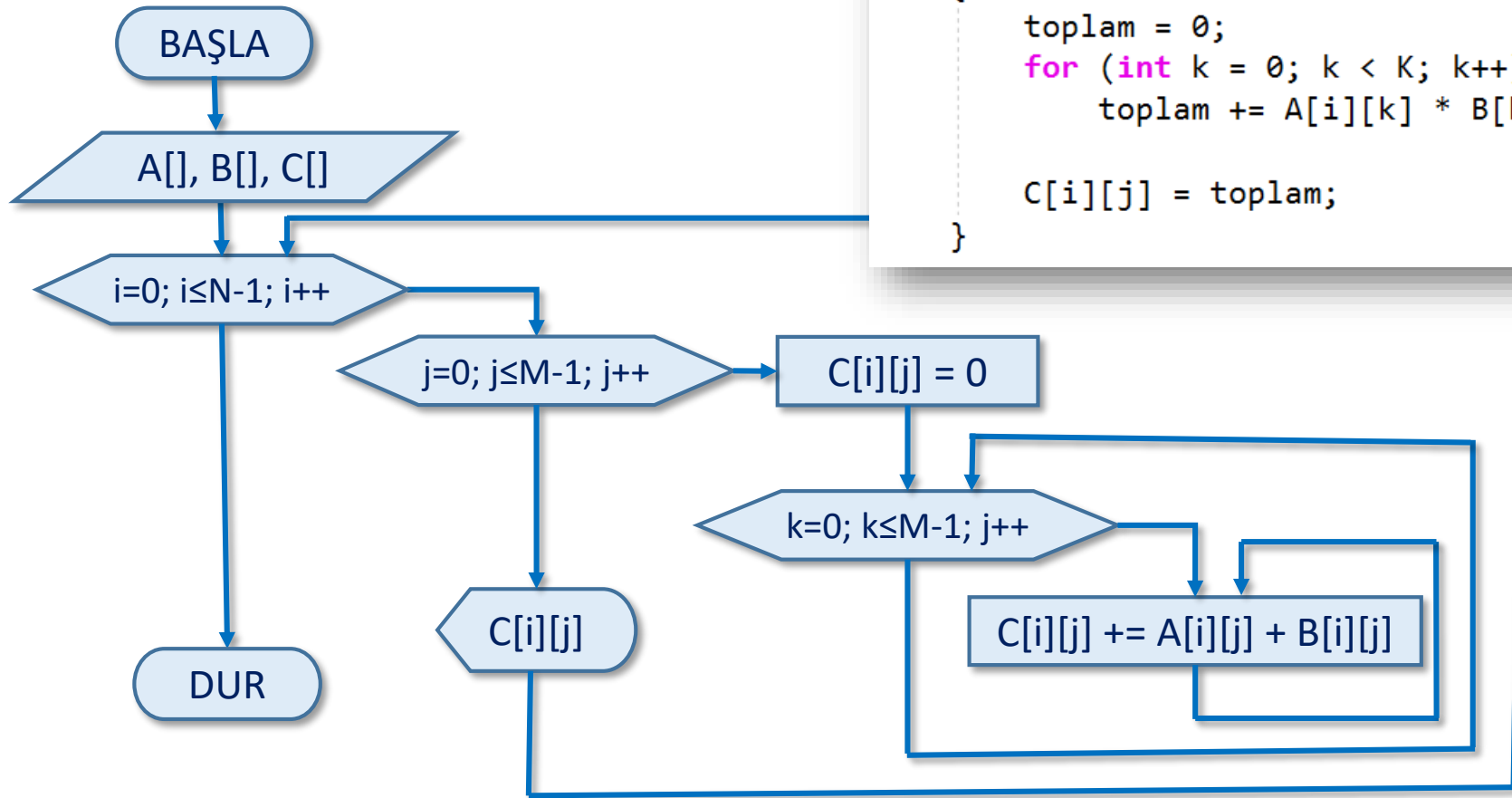
➤ Matrisin Transpozisini Alma

```
for (int i = 0; i < SATIR; i++)  
    for (int j = 0; j < SUTUN; j++)  
        T[j][i] = A[i][j] ;
```

Diziler, Karakter Katarları (Strings)...

➤ Çok Boyutlu Diziler...

➤ Matris Çarpma



```
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
    {
        toplam = 0;
        for (int k = 0; k < K; k++)
            toplam += A[i][k] * B[k][j];

        C[i][j] = toplam;
    }
```

Diziler, Karakter Katarları (Strings)...

➤ Dizi Elemanı Olarak Yapılar

```
struct Olcu {  
    int metre;  
    int cmetre;  
};
```

```
Olcu uzunluk[10];  
  
uzunluk[2].metre = 4;
```

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 yapı (struct) tanımlayınız. 5 öğrenciye ait bilgiler için bir dizi tanımlayınız. Bilgi girişi, yazdırma v.s. işlemleri 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.

Diziler, Karakter Katarları (Strings)...

➤ Dizilerde Sıralama...

- 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)...

➤ Dizilerde Sıralama...

➤ Seçmeli Sıralama (Selection Sort)

- 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)...

➤ Dizilerde Sıralama...

➤ Seçmeli Sıralama (Selection Sort)...

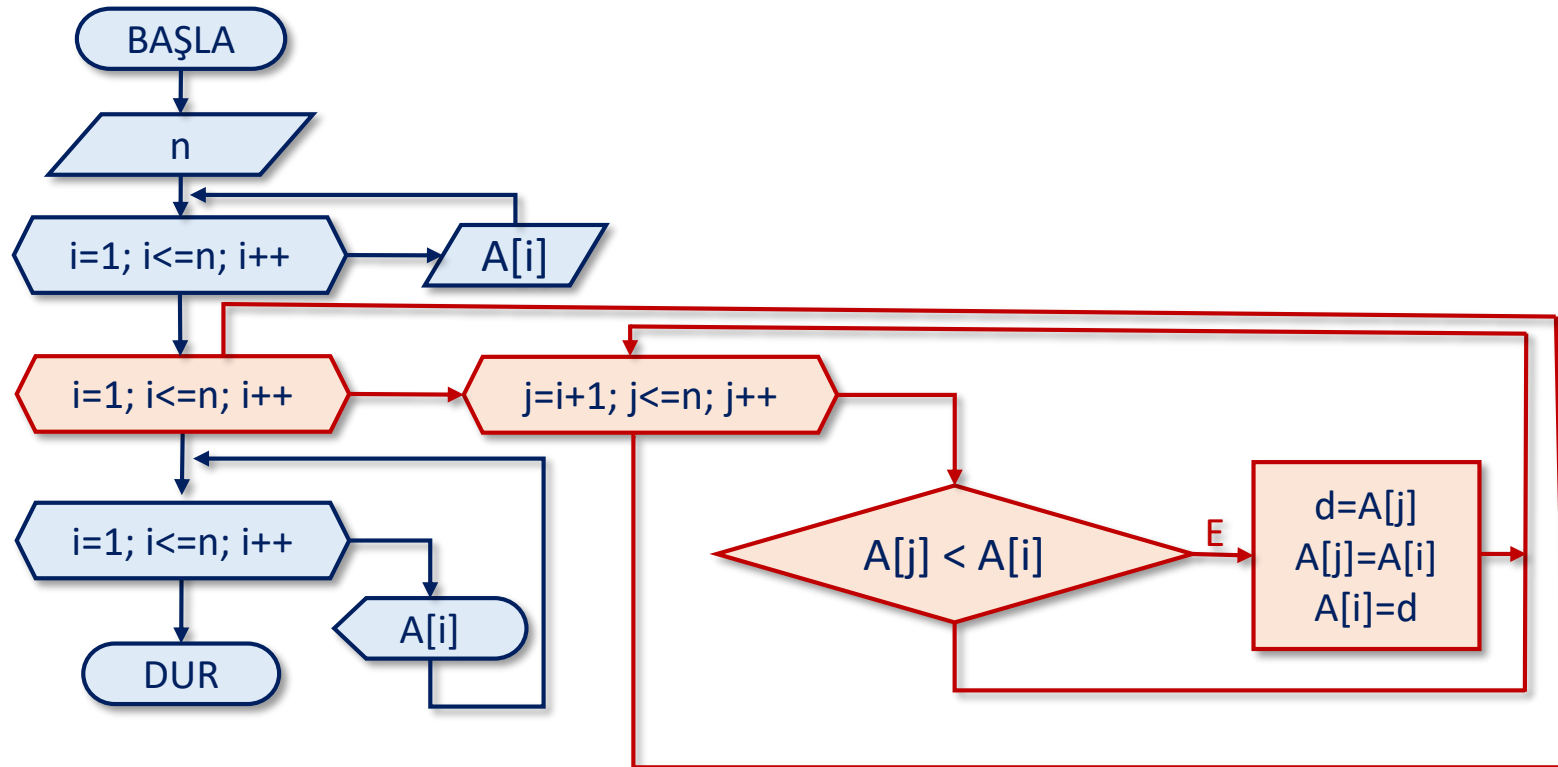
➤ Selection sort ile artan bir dizi sıralaması için;

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

Diziler, Karakter Katarları (Strings)...

➤ Dizilerde Sıralama...

➤ Seçmeli Sıralama (Selection Sort)...



Diziler, Karakter Katarları (Strings)...

➤ Dizilerde Sıralama...

➤ Kabarcık Sıralama (Bubble Sort)...

- 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/büyüklük durumu var ise yerleri değiştirilir.

Diziler, Karakter Katarları (Strings)...

➤ Dizilerde Sıralama...

➤ Kabarcık Sıralama (Bubble Sort)...

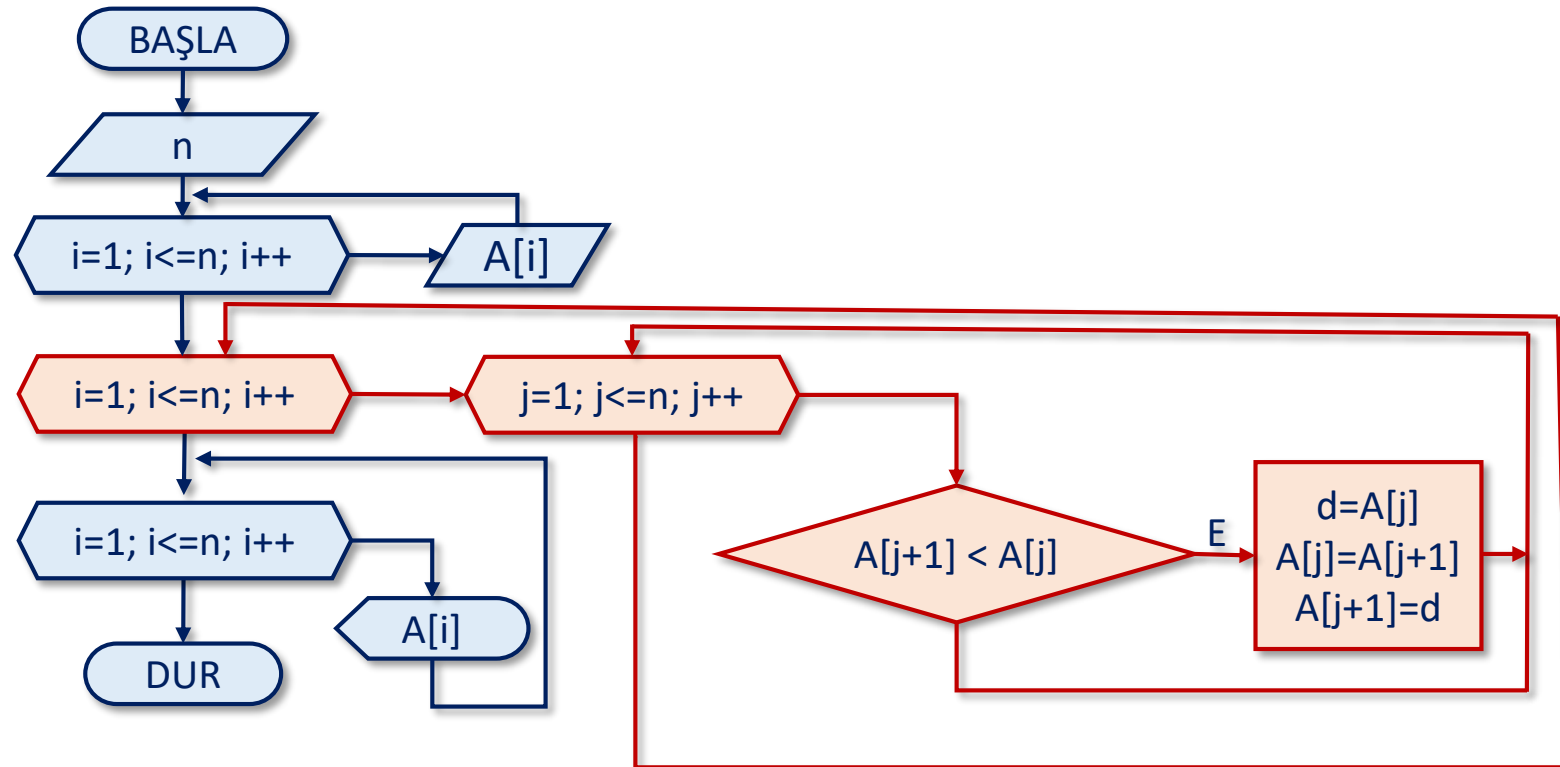
➤ Bubble sort ile artan bir dizi sıralaması için;

- 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)...

➤ Dizilerde Sıralama...

➤ Kabarcık Sıralama (Bubble Sort)...



Diziler, Karakter Katarları (Strings)...

➤ Dizilerde Sıralama...

➤ Kabarcık Sıralama (Bubble Sort)...

```
int gecici;  
  
for (int i = 0; i < ESAY; i++)  
    for (int j = 0; j < ESAY - 1; j++)  
        if (sayilar[j] > sayilar[j + 1])  
        {  
            gecici = sayilar[j];  
            sayilar[j] = sayilar[j + 1];  
            sayilar[j + 1] = gecici;  
        }
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı (String) – C String

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

```
char kdizi[] = { 'H', 'e', 'l', 'l', 'o', '\0' };  
//char kdizi[] = "Hello";  
//kdizi[0]->'H'  
cout << "Metin Girin:";  
cin >> kdizi;  
cout << kdizi;
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı (String) – C String...

➤ ÖRNEK: ⇒ [7]_karakter_dizisi.cpp

```
const int MAX = 80;  
char str[MAX];  
  
cout << "Karakter dizisi gir: ";  
cin >> str;  
  
cout << "Girilen karakter dizisi: " << str << endl;
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı (String) – C String...

➤ ÖRNEK: ⇒ [8]_karakter_dizisi2.cpp

```
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: ";  
// '.' - sınırlandırma karakteri  
cin.get(str, MAX, '.'); // Çoklu satır  
cout << "Girilen karakter dizisi: " << str << endl;
```

Diziler, Karakter Katarları (Strings)...

➤ Tek Karakter İçin Fonksiyonlar

Fonksiyon	Görevi
isdigit (int c)	c rakamsa TRUE, değilse FALSE döndürür
isalpha (int c)	c harfse TRUE, değilse FALSE döndürür
isalnum (int c)	c harf veya rakamsa TRUE, değilse FALSE döndürür
isxdigit (int c)	c hexadecimal ise TRUE, değilse FALSE döndürür
islower (int c)	c küçük harf ise TRUE, değilse FALSE döndürür
isupper (int c)	c büyük harf ise TRUE, değilse FALSE döndürür
tolower (int c)	Küçük harfe çevir
toupper (int c)	Büyük harfe çevir
isspace (int c)	c boşluk karakteri ise TRUE, değilse FALSE döndürür
isctrl (int c)	c 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	Dizgideki sayıyı double'a dönüştürür
strtol	Dizgideki sayıyı long int'e dönüştürür
strtout	Dizgideki sayıyı 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

➤ ÖRNEK: ⇒ [9]_karakter_dizisi3.cpp

```
char str1[] = "Aynası iştir kişinin lafa bakılmaz!...";

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

int uzunluk = strlen(str1);

for (int i = 0; i < uzunluk; i++)
    str2[i] = str1[i];

str2[uzunluk] = NULL;           // '\0'

cout << str2 << endl;
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı İşleme Fonksiyonları

Fonksiyon	Görevi
strcpy	Bir dizgiyi başka bir dizgiye kopyalar
strncpy	Bir dizginin belirli sayıdaki karakterini başka bir dizgiye kopyalar
strcat	Başka bir dizgiye ekler
strncat	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...

➤ ÖRNEK: ⇒ [10]_karakter_dizisi4.cpp

```
char str1[] = "Aynası iştir kişinin lafa bakılmaz!...";

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

// strlen() - karakter uzunluğu
cout << strlen(str1);

// str1'deki metni str2'ye kopyala
strcpy_s(str2, str1);
cout << str2 << endl;
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı

➤ ÖRNEK: ⇒ [11]_karakter_katari.cpp

```
const int DAYS = 7;
const int MAX = 10;

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

for (int i = 0; i < DAYS; i++)
    cout << gunler[i] << endl;

string gunler2[DAYS] = { "Pazartesi", "Salı", "Çarşamba",
                        "Perşembe", "Cuma", "Cumartesi", "Pazar" };

for (int i = 0; i < DAYS; i++)
    cout << gunler2[i] << endl;
```

```
#include <string>
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı Karşılaştırma Fonksiyonları

Fonksiyon	Görevi
strcmp(...)	İki karakter dizgisini karşılaştırır. Birincisi büyükse pozitif, ikincisi büyükse negatif, eşitse 0 döndürür.
strncmp(...)	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

➤ ÖRNEK: ⇒ [12]_karakter_katari2.cpp

```
char kopya1[50], kopya2[50], kelime[50];

cout << "Metin Giriniz: ";
cin >> kelime;

strcpy_s(kopya1, kelime);           // kelime katarını kopya1'e kopyala
cout << kopya1 << endl;

strncpy_s(kopya2, kelime, 4);       // kelime nin ilk 4 karakterini kopyala
cout << kopya2 << endl;

if (strcmp(kopya1, kelime) == 0)    // kopya1 ve kelime yi karsilastir
    cout << "kopyalama tamamdır";

strcat_s(kopya1, kelime);           // kelimeyi kopya1'e ekle

cout << kopya1 << endl;
```

```
#include <cstring>
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı (String) – String Sınıfı

➤ ÖRNEK: ⇒ [13]_karakter_katari3.cpp

```
string ad = "cenk";
string ad2 = "Mehmet";
string soyad = "Yalcin";

//cin >> ad1;           // Boşluktan sonrasını almaz
//cout << ad1;
getline(cin, ad2);       // Boşluğu da okur.
cout << ad2;

//ad1=ad2;
ad1.assign(ad2);         // ad1=ad2

cout << ad1;
ad1.append(" Mert");     // ad1 sonuna Mert ifadesini ekler

cout << ad1.compare(ad2); // ad1 ile ad2 yi karşılaştırır.
                        // Aynı ise 0 değerini döndürür
cout << ad1 << ad2 << endl;
cout << ad1.substr(5, 4); // ad

cout << endl << ad1.find("me"); // "me" ifadesinin ad1 içerisindeki konumu

ad1.insert(10, soyad);

cout << ad1.length();
```

```
#include <string>
```

Diziler, Karakter Katarları (Strings)...

➤ Tek Karakter İçin Fonksiyonlar

➤ ÖRNEK: ⇒ [14]_karakter_katari4.cpp

```
string str1 = "";
#include <string>

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;
```

Diziler, Karakter Katarları (Strings)...

➤ Karakter Katarı - String

➤ ÖRNEK: ⇒ [15]_karakter_katari5.cpp

```
string ad1;
```

```
#include <string>
```

```
getline(cin, ad1); //Boşluğu da okur...  
cout << ad1.length();
```

```
//Klavyeden girilen karakter dizisini tersten yazdırır  
for (int i = ad1.length() - 1; i >= 0; i--)  
    cout << ad1[i];
```

Çalışma Soruları

- Klavyeden girilen karakter katarını ters çevirerek ekrana yazan programı yazınız
 - "Ali" -> "ilA" gibi
- Klavyeden girilen karakter katarının her bir karakterini:
 - Tamamen büyük
 - Tamamen küçük
 - Büyükse küçük, küçükse büyük (Ali->aLi gibi)
 - Kelimelerin ilk harfini büyüğe çevirerek yazan programı geliştiriniz
 - Boşluk karakteri kontrol edilmeli
 - switch-case kullanarak seçmeli menü 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...)

Çalışma Soruları...

- Kullanıcı noktalama işaretlerine basmadığı sürece rasgele olarak üretilen sayının asal olup olmadığını bulan programı yazınız.
- Girilen karakter katarının uzunluğunu döndüren fonksiyonu hazır fonksiyon kullanmadan yazınız
- En fazla 10 karakter uzunluğunda (uzunluğu rastgele belirlenecek) A ile başlayan 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

- Deitel, C++ How To Program, Prentice Hall
- Horstmann, C., Budd, T., Big C++, Jhon Wiley & Sons, Inc.
- Robert Lafore, Object Oriented Programming in C++, Macmillan Computer Publishing
- Prof. Dr. Celal ÇEKEN, Programlamaya Giriş Ders Notları
- Prof. Dr. Cemil ÖZ, Programlamaya Giriş Ders Notları