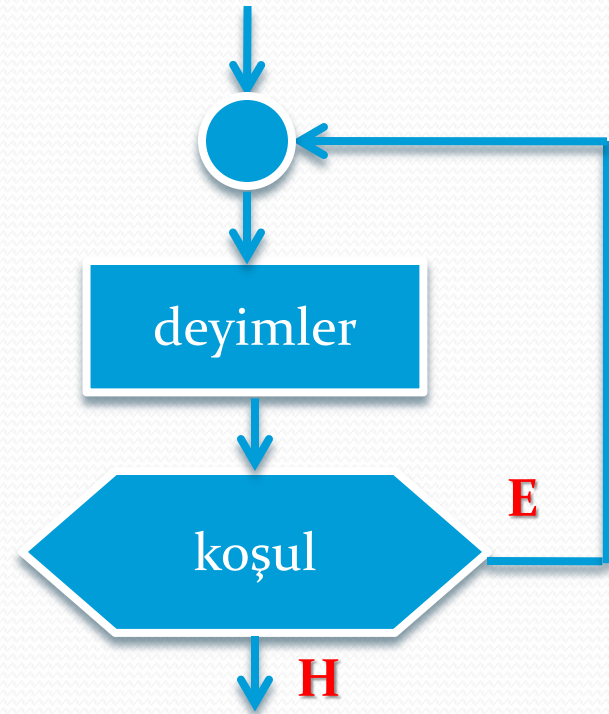


# PROGRAMLAMA TEMELLERİ

Döngüler  
do .. while ..

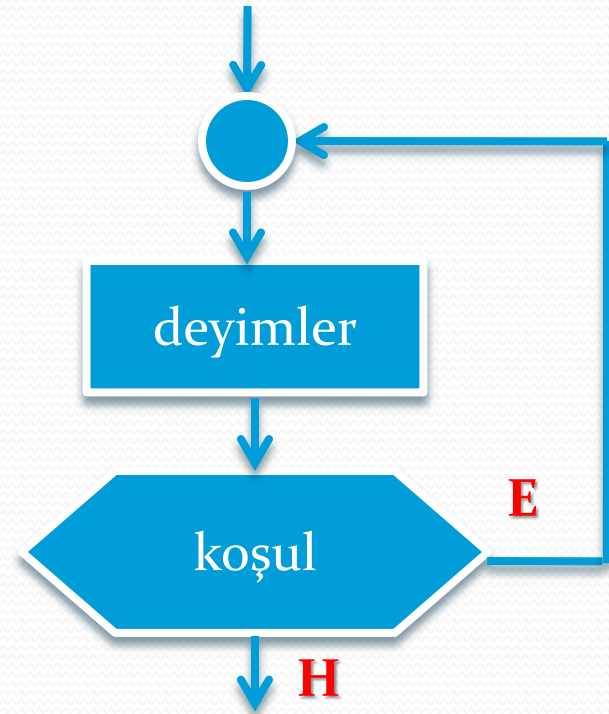
# Döngüler: ..OLMAYANA KADAR

- Bir koşul gerçekleşene kadar **en az bir kez** çalıştırılması gereken komutlar varsa kullanırız.
- Koşula etki eden etmenler – WHILE döngüsünde olduğu gibi- döngü içerisinde değişikliğe uğrayabilir.
- Esnek ve verimli döngüdür.



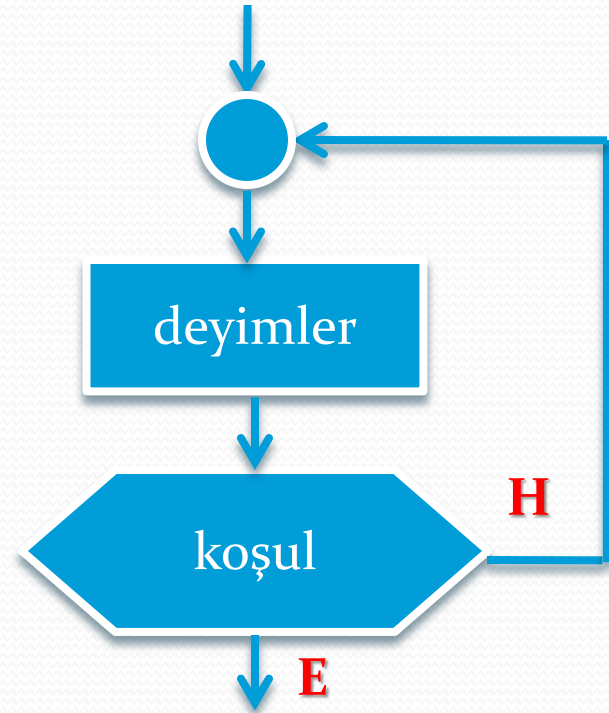
# Döngüler: .. OLMAYANA KADAR

- Koşul döngü sonunda kontrol edilir.
- Koşuldan önce döngü komutları bir kez çalışmıştır.
- Eğer koşul yine sağlanırsa döngü başına gidilir ve komutlar bir kez daha çalıştırılır.
- Döngüden çıkmak için koşul gerçekleşmemelidir.



# Döngüler: .. OLMAYANA KADAR

- Genellikle
  - klavyeden girişlerin kontrolünde
  - Bir tuşa basmayı beklemede
  - Menülerde çıkış kontrolünde
  - Vs kullanılır.
- FOR döngüsü yerine kullanılmaz.



# C++'ta do .. while ..

- Döngü sonundaki mantıksal ifade gerçekleşmeyene kadar çalışacak deyimler içerir.
- Yazım

```
do {  
    <deyim/ler>;  
}while (<koşul>;
```
- Döngü komutlarının başında **do**, sonunda da **while** deyimi bulunur. Birden fazla deyim içerdiği durumlarda **do** sonrasında «{« ve **while** öncesinde «}» kullanılır.
- **while** komutundan sonra gelen koşul **true** olduğu sürece bu deyimler çalıştırılır, **false** olursa döngü biter.
- Koşul döngü deyimlerinden sonra değerlendirilir.

# Örnekler

- Eğer özel bir tuşa basılmasını beklersek:

**do**

```
cout<<"CIKMAK İSTİYOR MUSUNUZ? ";
```

```
while (getchar() != 'e');
```

- Eğer girdi yapılırken kullanıcının devam edip etmeyeceği sorulacaksa;

**do{**

**:**

```
cout << "Devam edecek misiniz? [e/h]:";
```

```
cin>>cevap;
```

```
}while (cevap=='e');
```

# Örnekler

- Kullanıcının doğru şifreyi girmesini beklersek:

```
long pin=9876, upin;  
do{  
    cout <<"Pin: ";  
    cin >> upin;  
}while (upin!=pin) ;
```

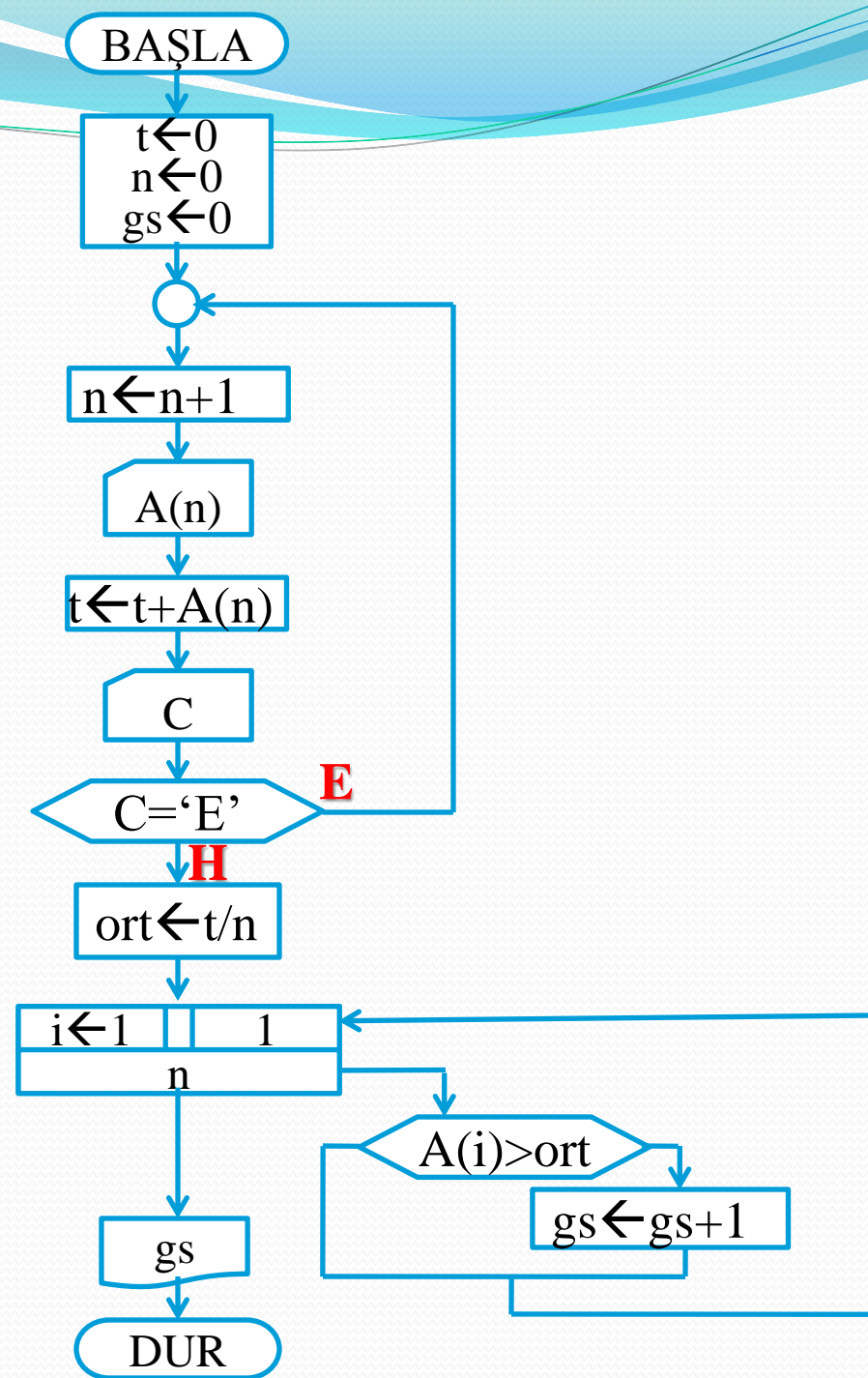
# Örnek: BDS2

- Kullanıcının klavyeden girdiği notları okuyan, notların ortalamasını hesaplayan, ortalamanın üzerindeki notları sayan ve geçenlerin sayısı olarak ekrana yazan program.
- **Algoritma**
- Not toplamını, not sayısını ve geçen sayısını sıfırla
- TEKRARLA
  - Not sayısını 1 artır, Notu oku, not toplamına ekle
  - Kullanıcıya yeni not girip girmeyeceğini sor
- Yeni not girmek istemeyene KADAR
- Not ortalamasını hesapla
- 1'den başla, 1er ileri sayarak not sayısına kadar
  - Eğer not ortalamadan yüksekse geçen sayısını 1 artır
- Geçen sayısını yazdır



# Örnek: BDS2

## Akış Diyagramı



# Örnek: BDS2 Pascal Kodu

```
// Klavyeden vize notu girildikçe notları toplayan
// ve sonunda ortalamayı hesaplayıp ortalamamanın
// üzerindeki sayısını yazan program
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    long t=0;
    int A[100], n=0, gs=0;
    char c;
    double ort;
    do{
        cout << "Notu giriniz: ";
        cin >> A[n];
        t+=A[n];
        n++;
        cout << "Devam edecek misiniz? [e/h] : ";
        cin >> c;
    }while(c=='e');
    ort=t/n;
    for(int i=0; i<n; i++)
        if(A[i]>ort) gs++;
    cout <<"Ortalama: " << ort << endl;
    cout << "GECEN SAYISI: " << gs << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Örnek: BDS2 Pascal Kodu

```
// Klavyeden vize notu girildikçe notları toplayan
// ve sonunda ortalamayı hesaplayıp ortalamamanın
// üzerindeki sayısını yazan program
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    long t=0;
    int A[100], n=-1, gs=0;
    char c;
    double ort;
    do{
        n++;
        cout << "Notu giriniz: ";
        cin >> A[n];
        t+=A[n];
        cout << "Devam edecek misiniz? [e/h] : ";
        cin>>cevap;
    }while(cevap=='e');
    ort=t/n;
    for(int i=0; i<=n; i++)
        if(A[i]>ort) gs++;
    cout <<"Ortalama: " << ort << endl;
    cout << "GECEN SAYISI: " << gs << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Break (kes)

- Program akışını Blok dışına yönlendirir.
- Genelde döngü (for, while, do..while), switch..case yapılarında kullanılır.

# Örnek: break

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int i=0;
    for(;;)
    {
        i++;
        cout<< i <<endl;
        if(i==25) break;
    }
    /*
    for(i=0;i<=25;i++)
        cout << i << endl;
    */
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Continue (devam et)

- Döngülerde kendisinden sonraki döngü komutlarının çalışmasını engelleyerek program akışını döngü kontrolüne yönlendirir.

# Örnek: continue

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

using namespace std;

int main(int argc, char *argv[])
{
    for(int i=0;i<10000;i++)
    {
        cout << setw(8);
        if(i%2==0)
            continue;
        cout << i;
    }
    /*
    for(int i=1;i<10000;i+=2)
        cout << setw(8) << i;
    */
    system("PAUSE");
    return EXIT_SUCCESS;
}turn EXIT_SUCCESS;
}
```

# Ödevler: (6.12.2011)

- Klavyeden kullanıcı pozitif tamsayı girecek, program bu sayının bütün asal çarpanlarını bulup, diziyeye atayacak  
 $144 \rightarrow 2$  (2 tane),  $3$  (4 tane)
- Kullanıcı klavyeden n tane sayı girecek, program en fazla tekrar eden sayıyı bulacak yazacak  
 $\{1,3,2,2,4,4,5,4,3,4,4,2,3,4,2,2,5\} \rightarrow 4$  (6 kez tekrar etmiş)
- Kullanıcı klavyeden n tane sayı girecek, program en büyük sayıyı, en küçük sayıyı bulacak yazacak  
 $\{6,8,5,7,3,9,4\} \rightarrow 9$  (en büyük),  $3$  (en küçük)