

Bilgisayar Programlama I

DERS 4: DÖNGÜLER

DR. HÜSEYİN BAHTİYAR

Çok yönlü Koşullar: Bulmacalar

2

X değerinden bağımsız olarak
hangi koşul asla çalışmaz?

```
if (x < 2) {  
    cout<<'2 den küçük';  
}  
else if (x >= 2) {  
    cout<<'2 veya daha büyük';  
}  
else{  
    cout<<'başka birşey';  
}
```

Switch case komutu

3

- ▶ Klavyeden girilen sayının 1,2,3 olmasına göre ayrı işlemler yapan bir program düşünelim
 - ▶ 1 olunca sonucu -1 versin
 - ▶ 2 olunca karesini alsın.
 - ▶ 3 olduğunda ise 4 ile çarpsın
 - ▶ bu girilenlerden bir sayı olursa o sayının 2 katını versin.
- ▶ Bu şekildeki bir programı öğrendiğimiz if else komutu ile yazabiliriz.

```
#include <iostream>
using namespace std;
int main(){
    int sayi;
    cout <<"sayi gir: "<<endl;
    cin >> sayi;
    if (sayi==1) {
        cout <<-sayi;
    } else if(sayi==2){
        cout <<sayi*sayi;
    }else if(sayi==3){
        cout<<sayi*4;
    }else{
        cout<<sayi*2;
    }
    cout <<"bitti"<<endl;
    return 0;
}
```

Benzer seçici yapıyı switch-case komutu ile oluşturabiliriz.

```
switch(ifade){
    case sabit1:
        //sabit1 olduğunda yapılacaklar
        break;
    case sabit2:
        //sabit2 olduğunda yapılacaklar
        break;
    .
    .
    default:
        //sabit tanımlanan dışında olduğunda yapılacaklar
        break;
}
```



```
#include <iostream>
using namespace std;
int main(){
    int sayi;
    cout <<"sayi gir: "<<endl;
    cin >> sayi;
    if (sayi==1) {
        cout <<-sayi;
    } else if(sayi==2){
        cout <<sayi*sayi;
    }else if(sayi==3){
        cout<<sayi*4;
    }else{
        cout<<sayi*2;
    }
    cout <<"bitti"<<endl;
    return 0;
}
```

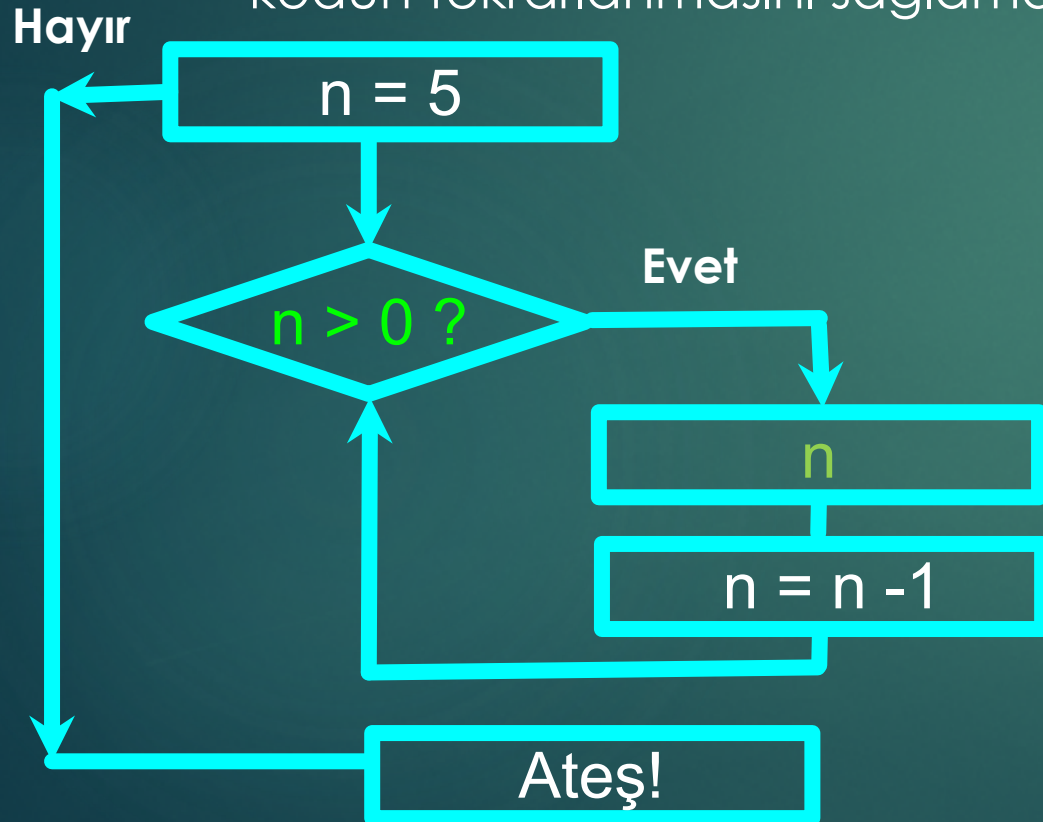


```
#include <iostream>
using namespace std;
int main(){
    int sayi;
    cout <<"sayi gir: "<<endl;
    cin >> sayi;
    switch (sayi){
        case 1:
            cout <<-sayi;
            break;
        case 2:
            cout <<sayi*sayi;
            break;
        case 3:
            cout<<sayi*4;
            break;
        default:
            cout<<sayi*2;
            break;
    }
    cout <<"bitti"<<endl;
    return 0;
}
```

Döngüler ve İterasyon

6

- Döngülerin amacı, verilen koşulun gerçekleşmesi süresince belirli bir kodun tekrarlanmasını sağlamaktır.



Çıktısı:

5
4
3
2
1
Ateş!

Belirli / Belirsiz Döngüler

7

- ▶ Döngü yapıları ikiye ayrılır, belirli ve belirsiz döngüler.
 - ▶ Belirsiz döngülerde mantıksal bir **koşul** vardır ve o koşul **Yanlış** olana kadar döngü devam eder.
 - ▶ Eğer döngü sayısını bilmiyorsak bir koşul sonucunda döngüden çıkmasını istiyorsak belirsiz döngüleri kullanmamız gerekir.
 - ▶ **Örnek:** Kullanıcı negatif sayı girene kadar sayı girmesini istemek.
 - ▶ Belirli Döngülerde ise döngü sayısını (sonlu sayıda) bildiğimiz yapılarda kullanırız.
 - ▶ **Örnek:** kullanıcıdan 5 kere üst üste sayı girmesini istiyorsak belirli döngüleri kullanırız.

Belirsiz Döngüler

8

- ▶ Belirsiz döngüler için while veya do while komutlarını kullanırız.
- ▶ While döngüsünde **önce koşul kontrol edilir doğruysa döngü işlemi yapılır.**

```
while (koşul) {
```

```
Yapılmasını istediğimiz komutlar;
```

```
}
```

```
while (n>0) {  
    cout<<"sayi pozitif"<<endl;  
    n--;  
}
```

- ▶ While döngüsünde **önce işlem yapılır son olarak kontrol edilir doğruysa bir sonraki döngü işlemine geçilir.**

```
do {
```

```
Yapılmasını istediğimiz komutlar;
```

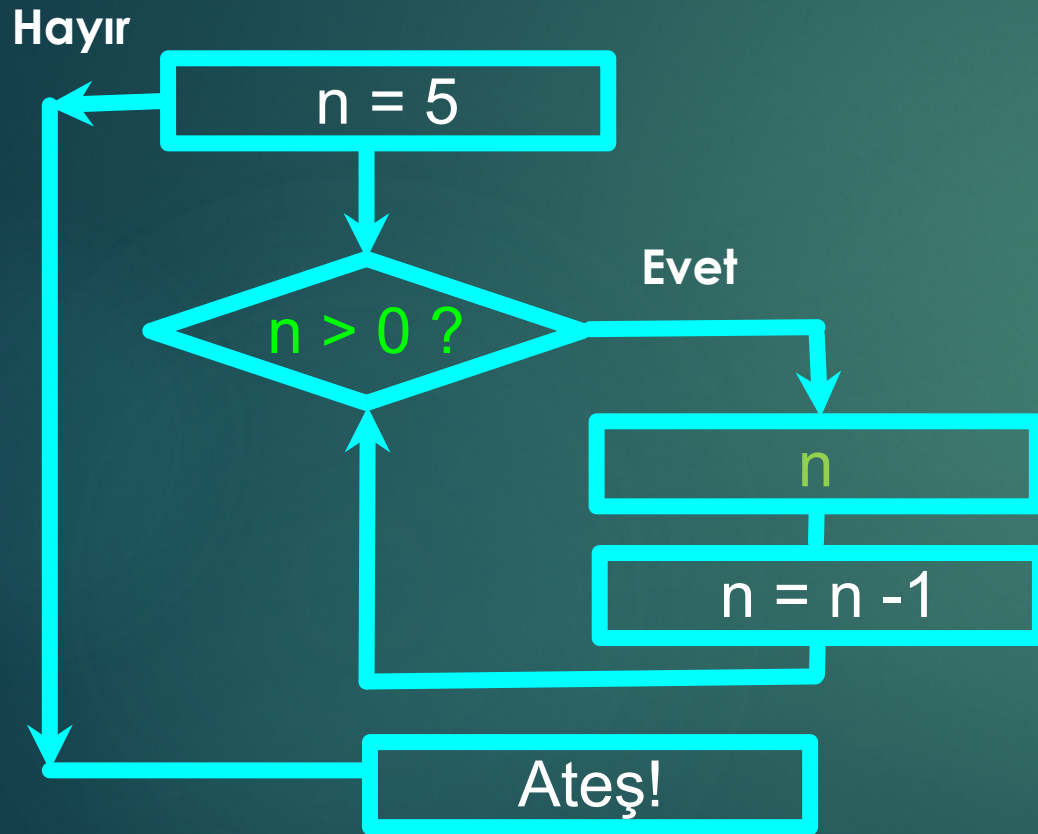
```
} while (koşul)
```

```
do {  
    cout<<"sayi pozitif"<<endl;  
    n--;  
}while (n>0);
```


Belirsiz Döngüler

Örnekler:

9



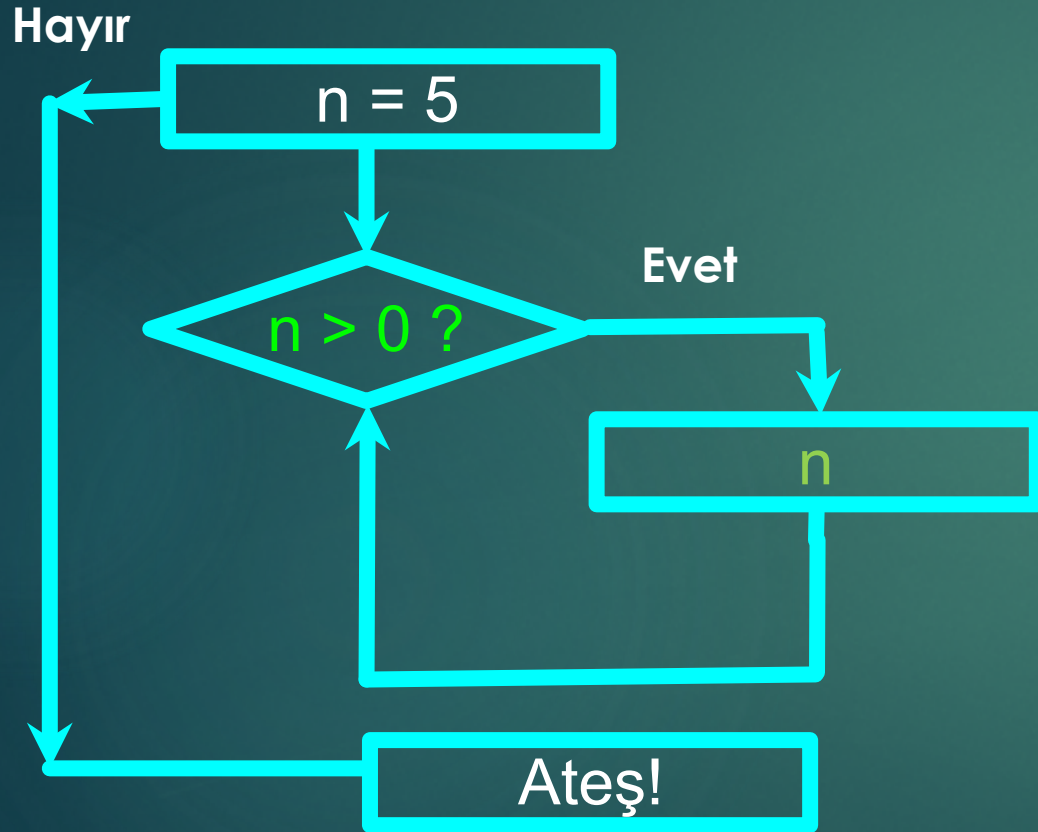
```
#include <iostream>
using namespace std;
int main(){
    int n = 5;
    while(n > 0){
        cout << n << endl;
        n = n - 1;
    }
    cout << "Ateş" << endl;
    return 0;
}
```

Çıktısı:

5
4
3
2
1
Ateş!

Belirsiz Döngüler

Örnek: Sonsuz Döngü

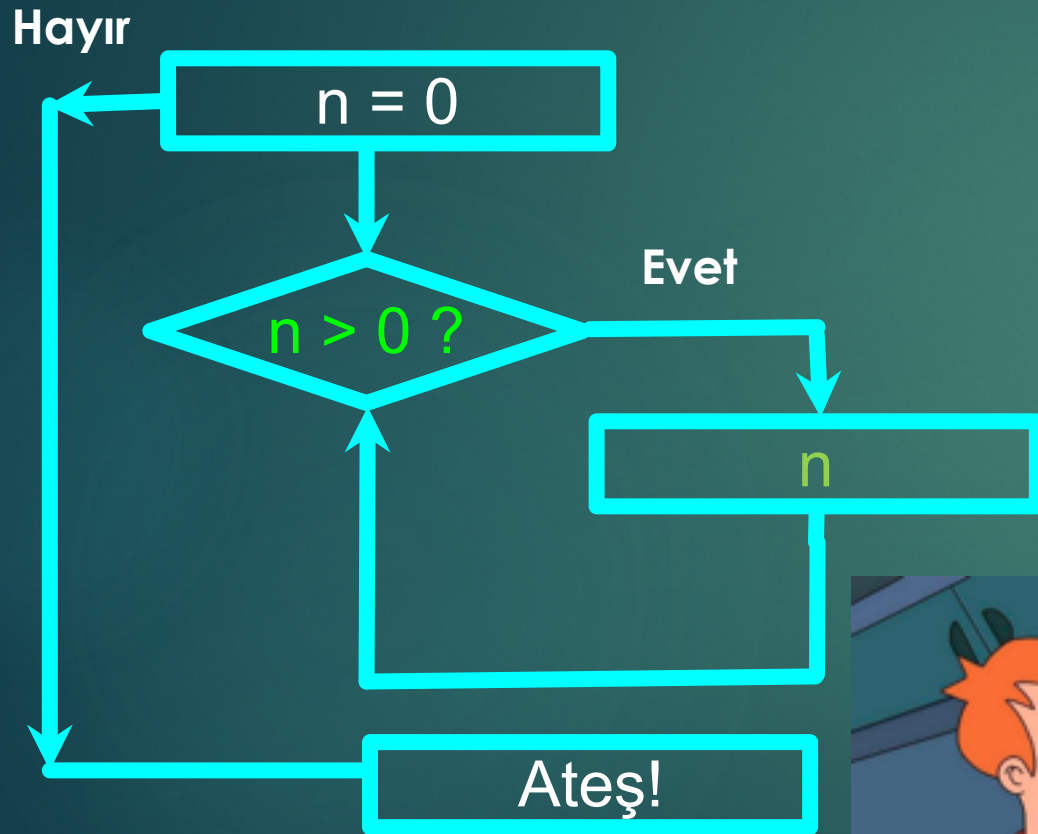


```
#include <iostream>
using namespace std;
int main(){
    int n =5;
    while(n>0){
        cout<<n<<endl;
    }
    cout<<"Ateş"<<endl;
    return 0;
}
```

Bu döngüdeki sorun nedir?

Belirsiz Döngüler

Örnek: Başka bir döngü



```
#include <iostream>
using namespace std;
int main(){
    int n = 0;
    while(n > 0){
        cout << n << endl;
    }
    cout << "Ateş" << endl;
    return 0;
}
```

Peki bu döngü ne yapıyor??

Döngüden Çıkmak

12



- **break** komutu ile döngünün içindeki yerden zıplayarak döngü dışındaki komuta **ışınlanabiliriz** !

```
#include <iostream>
using namespace std;
int main(){
    int n = 5;
    while(n > 0){
        cout << n << endl;
        if(n == 2){
            break;
        }
        n--;
    }
    cout << "Ateş" << endl;
    return 0;
}
```

Çıktısı:

5
4
3
2

Ateş!

Bir yinelemeyi **continue** ile bitirmek

13

- **continue** komutu ile **geçerli olan döngünün** geriye kalan kısmı işleme konulmaz bir sonraki döngüye sıçar.

```
#include <iostream>
using namespace std;
int main(){
    int n =6;
    while(n>0){
        n--;
        if(n==2){
            continue;
        }
        cout<<n<<endl;
    }
    cout<<"Ateş"<<endl;
    return 0;
}
```

Çıktısı:

5

4

3

1

Ateş!

Exit ile programdan çıkmak

14

- ▶ exit komutu programdan çıkmak için kullanılır, cstdlib kitaplığında tanımlanmıştır. Kullanımındaki amaç özel bir çıkış kodu ile programı sonlandırmaktır.
- ▶ Programın devam etmesini istemediğimiz durumlarda kullanabiliriz.

Exit ile programdan çıkmak

15

- ▶ exit komutu programdan çıkmak için kullanılır, cstdlib kitaplığında tanımlanmıştır. Kullanımındaki amaç özel bir çıkış kodu ile programı sonlandırmaktır.
- ▶ Programın devam etmesini istemediğimiz durumlarda kullanabiliriz.

exit komutu için
kütüphane ekledik!

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
    int n = 6;
    while(n > 0){
        n--;
        if(n == 2){
            exit(1);
        }
        cout << n << endl;
    }
    cout << "Ateş" << endl;
    return 0;
}
```

Çıktısı:

5
4
3

Belirli Döngüler

16

- Yineleme sayısı belli olan döngülerdir. Başlangıç değeri, koşul ve artırma miktarı şeklinde 3 adımlıdır. **Kullanımı:**

```
for( başlangıç; koşul; artım) {
```

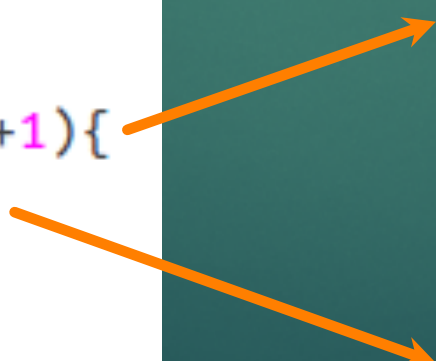
Döngüde yapılacaklar

```
}
```

Çıktısı:

```
#include <iostream>
using namespace std;
int main(){
for (int i =0; i<5; i=i+1){
    cout <<i<<endl;
}
return 0;
}
```

0
1
2
3
4



Belirli Döngüler

17

- Yineleme sayısı belli olan döngülerdir. Başlangıç değeri, koşul ve artırma miktarı şeklinde 3 adımlıdır.

```
for( başlangıç; koşul; artım) {
```

Döngüde yapılacaklar

```
}
```

1. Öncelikle döngü değişkeninin başlangıç koşulu yazılır.
2. Çalışma koşulu yazılır.
3. Artım belirlenir.

Belirli Döngüler

Örnek:

18

- ▶ 5 ten geriye sayıp ateşleme yapan programı for döngüsü ile yazalım.

Belirli Döngüler

Örnek:

- 5 ten geriye sayıp 1 den sonra Ateş yazan programı for döngüsü ile yazalım.

```
#include <iostream>
using namespace std;
int main(){
    for (int i =5; i>0; i=i-1){
        cout <<i<<endl;
    }
    cout<<"Ateş"<<endl;
    return 0;
}
```

Örnek:

Klavyeden girilen sayının karekökü

- ▶ Klavyeden **pozitif sayı girilene kadar sayı girmemizi isteyen**, pozitif sayı girilince de **karekökünü alan** program yazalım.

Örnek:

21

Klavyeden girilen sayının karekökü

Matematik Kütüphanesi
Karekök fonksiyonunu
Kullanmak için çağırdık.

Önce girdiyi alıp sonra
kontrol sağlasın diye
do – while kullandık.

Sayının karekökünü aldık

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
double n;
do{
cout<<"Karekoku alincak sayi girin: "<<endl;
cin >>n;
}while(n<0);
double karekok;
karekok = sqrt(n);
cout << n <<" sayisinin karekoku "<<karekok;
return 0;
}
```

Örnek:

22

Klavyeden girilen sayının karekökü 2

- ▶ Klavyeden **pozitif sayı girilince karekökünü alıp tekrar sayı girmemizi** isteyen, **negatif sayı girilince de programdan çıkan** program yazalım.

Örnek:

23

Klavyeden girilen sayının karekökü 2

- ▶ Klavyeden **pozitif sayı girilince karekökünü alıp tekrar sayı girmemizi** isteyen, **negatif sayı girilince de döngüden çıkan** program yazalım.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
double n;
do{
cout<<"Karekoku alincak sayi girin: "<<endl;
cin >>n;
if(n<0){
break;
}
double karekok;
karekok = sqrt(n);
cout << n <<" sayisinin karekoku "<<karekok<<endl;
}while(n>0);
cout<<"negatif sayı girdiniz!"<<endl;
return 0;
}
```

Sıfırdan küçükse

Döngüden çık

Örnek:

Belirli döngüler

24

- ▶ Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.
- ▶ **İpucu:**
 - ▶ For döngüsü kullanacağız.

Örnek:

Belirli döngüler

25

- Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int n;
    cout<<"Sayi girin: "<<endl;
    cin >>n;
    int toplam = 0;
    for(int i =0; i<n; i++){
        toplam = toplam +i;
    }
    cout<<"0 - "<<n<<" arası sayıların toplamı |"<< toplam<<endl;
    return 0;
}
```

Toplam değeri tutacak değişken.
İlk değeri 0 atadık.

Örnek:

Belirli döngüler

26

- Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int n;
    cout<<"Sayi girin: "<<endl;
    cin >>n;
    int toplam = 0;
    for(int i =0; i<n; i++){
        toplam = toplam +i;
    }
    cout<<"0 - "<<n<<" arası sayıların toplamı |"<< toplam<<endl;
    return 0;
}
```

Toplam değeri tutacak değişken.
İlk değeri 0 atadık.

Döngü öncesi
0

Döngü sonrası
 $0+1=1$

Örnek:

Belirli döngüler

27

- Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int n;
    cout<<"Sayi girin: "<<endl;
    cin >>n;
    int toplam = 0;
    for(int i =0; i<n; i++){
        toplam = toplam +i;
    }
    cout<<"0 - "<<n<<" arası sayıların toplamı |"<< toplam<<endl;
    return 0;
}
```

Toplam değeri tutacak değişken.
İlk değeri 0 atadık.

Döngü öncesi

0
1

Döngü sonrası

0+1=1
1+2=3

Örnek:

Belirli döngüler

28

- Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int n;
    cout<<"Sayi girin: "<<endl;
    cin >>n;
    int toplam = 0;
    for(int i =0; i<n; i++){
        toplam = toplam +i;
    }
    cout<<"0 - "<<n<<" arası sayıların toplamı |"<< toplam<<endl;
    return 0;
}
```

Toplam değeri tutacak değişken.
İlk değeri 0 atadık.

Döngü öncesi	Döngü sonrası
0	$0+1=1$
1	$1+2=3$
2	$2+3=5$

Örnek:

Belirli döngüler

29

- Sıfırdan başlayıp klavyeden girilen sayıya kadar olan sayıların toplamını yazdıran program yazın.

```
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int n;
    cout<<"Sayi girin: "<<endl;
    cin >>n;
    int toplam = 0;
    for(int i =0; i<n; i++){
        toplam = toplam +i;
    }
    cout<<"0 - "<<n<<" arası sayıların toplamı |"<< toplam<<endl;
    return 0;
}
```

Toplam değeri tutacak değişken.
İlk değeri 0 atadık.

Döngü öncesi	Döngü sonrası
0	$0+1=1$
1	$1+2=3$
2	$2+3=5$
.....	
36	$36+9=45$

Kaynaklar

30

- ▶ <http://www.cplusplus.com/doc/tutorial/>
- ▶ www.py4e.com
- ▶ C++ How to program (Deitel)
- ▶ Ferhat Özok Bilgisayar Programlama I ders notları