

# DÖNGÜLER

BMÜ-111 ALGORİTMA VE  
PROGRAMLAMA-I

YRD. DOÇ. DR. İLHAN AYDIN

# GİRİŞ

Ekrana «Merhaba Java» şeklinde bir mesajı 100 defa yazmak istediğinizi düşünelim.

Aşağıdaki komut satırını 100 kez yazmak sıkıcı olacaktır.

```
System.out.println("Merhaba Java!");
```

Bu problemi nasıl çözebiliriz?

# AÇIK PROBLEM

Problem:

100  
kez

```
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");
```

...

...

...

```
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");  
System.out.println("Merhaba Java!");
```

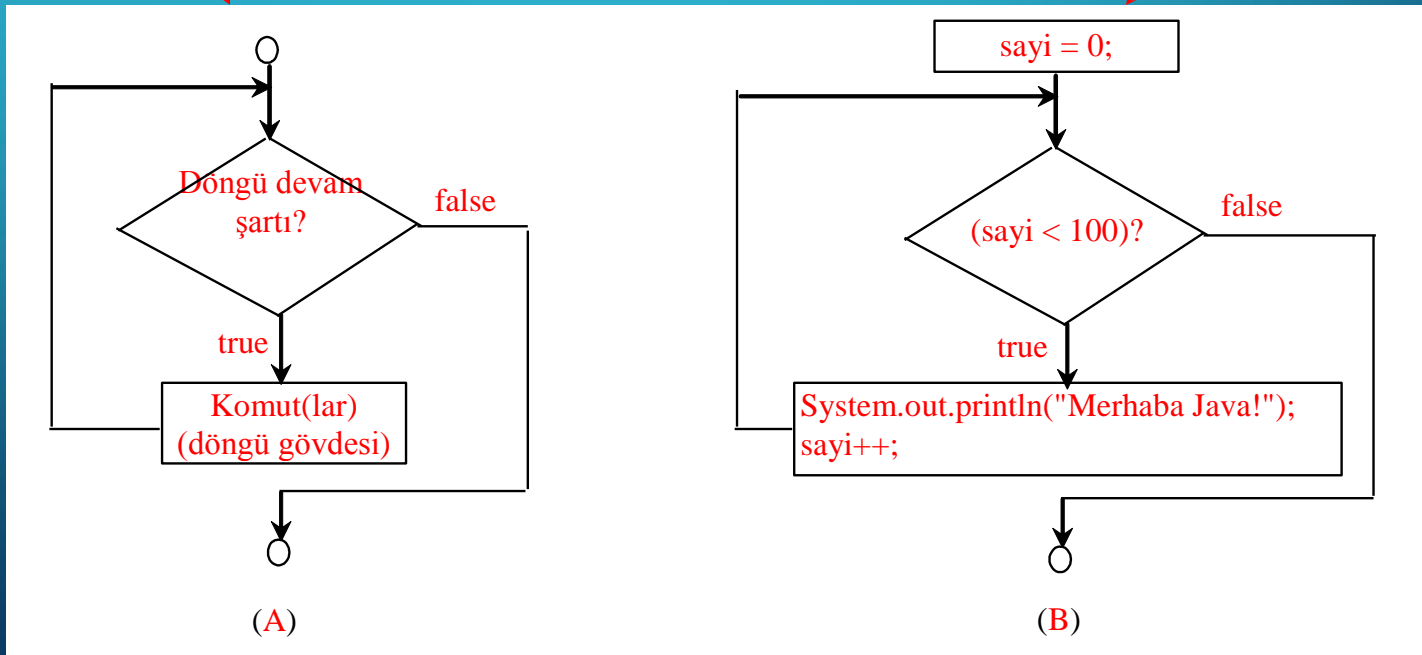
# WHILE DÖNGÜLERİNİN TANIMLANMASI

```
int sayi = 0;  
while (sayi < 100) {  
    System.out.println("Merhaba Java");  
    sayi++;  
}
```

# WHILE DÖNGÜSÜ AKIŞ ŞEMASI

```
while (Döngü devam şartı) {  
    // döngü-gövdesi;  
    Komutlar;  
}
```

```
int sayi, = 0;  
while (sayi < 100) {  
    System.out.println("Merhaba Java!");  
    sayi++;  
}
```



# WHILE DÖNGÜSÜ ÖRNEK

Sayac baslangici

```
int sayac = 0;
```

```
while (sayac < 2) {
```

```
    System.out.println("Merhaba Java!");
```

```
    sayac++;
```

```
}
```

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;
```

```
while (sayac < 2) {
```

```
    System.out.println("Merhaba Java!");
```

```
    sayac++;
```

```
}
```

(sayac < 2) true değeri alır

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;
while (sayac < 2) {
    System.out.println("Merhaba Java!");
    sayac++;
}
```

Ekrana Merhaba Java yaz



# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;  
while (sayac < 2) {  
    System.out.println("Merhaba Java!");  
    sayac++;  
}
```

sayac degerini 1 arttır  
sayac şimdi 1

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;

while (sayac < 2) {

    System.out.println("Merhaba Java!");
    sayac++;
}
```

(sayac < 2) halen doğru çünkü  
sayac'ın değeri şu an 1

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;
while (sayac < 2) {
    System.out.println("Merhaba Java!");
    sayac++;
}
```

Ekrana Merhaba Java yaz

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;  
while (sayac < 2) {  
    System.out.println("Merhaba Java!");  
    sayac++;  
}
```

Sayacı 1 arttır  
Sayac şimdi 2 değerini alır

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;
```

```
while (sayac < 2) {
```

```
    System.out.println("Merhaba Java!");
```

```
    sayac++;
```

```
}
```

(sayac < 2) false olur çünkü sayac şu anda 2 değerindedir.

# WHILE DÖNGÜSÜ ÖRNEK

```
int sayac = 0;
while (sayac < 2) {
    System.out.println("Merhaba Java!");
    sayac++;
}
```

Döngüden çık ve sonraki komutu yürüt.

# WHILE : SAYAN DÖNGÜ ÖRNEĞİ

□Kullanıcı tarafından girilen 10 sayıyı toplayan program

```
int sonraki;  
//döngü başlangıcı  
int sayac = 1;  
int toplam =0;  
while(sayac <= 10) //döngü sonlandırma koşulu  
{ //döngü gövdesi  
    sonraki = giris.nextInt();  
    toplam = toplam + sonraki;  
    sayac++; // döngü sonlandırma sayacı  
}
```

# WHILE: GÖZCÜ KONTROLLÜ DÖNGÜ ÖRNEĞİ

- sonraki gözcüdür
- döngü negatif bir sayı girilince sona erer

```
//Başlangıç
int sonraki = 0;
int toplam = 0;
while(sonraki >= 0) //sonlandırma şartı
{ //gövde
    toplam = toplam + sonraki;
    sonraki = giris.nextInt();
}
```



# WHILE: MİNİMUM SIFIR TEKRAR

- birinci giriş değeri döngüden önce okunup test edildiği için *while* döngüsünün gövdesi hiç çalıştırılmayabilir

```
int sonraki;  
int toplam= 0;  
sonraki = giris.nextInt();  
while(sonraki >= 0)//sonlandırma şartı  
{ //Body  
    toplam = toplam + sonraki;  
    sonraki = giris.nextInt();  
}
```

- Eğer kullanıcının girdiği ilk numara negatif ise döngüye girilmez

# WHILE ÖRNEK: EN BÜYÜK ORTAK BÖLEN

```
import java.util.Scanner;

public class Uygulama1{

    public static void main(String[] args) {

        Scanner klavye=new Scanner(System.in);

        System.out.println("Birinci sayıyı girin");

        int sayi1=klavye.nextInt();

        int sayi2=klavye.nextInt();

        int enbbolen=1,k=2;

        while(k<=sayi1 && k<=sayi2){

            if(sayi1%k==0 && sayi2%k==0)

                enbbolen=k;

            k++;

        }

        System.out.println("En büyük ortak Bölen = " + enbbolen);

    }

}
```

Birinci sayıyı girin  
60  
İkinci sayıyı girin  
36  
enbbolen = 12

# WHILE ÖRNEK: EKRANDAN SIFIR GİRİLENE KADAR GİRİLEN SAYILARIN TOPLAMINI BULAN ÖRNEK

```
import java.util.Scanner;

public class Uygulama1{

    public static void main(String[] args) {

        Scanner klavye=new Scanner(System.in);

        int top=0;

        System.out.println("Sıfırdan farklı sayı girin");

        int sayi=klavye.nextInt();

        while(sayi!=0){

            top=top+sayi;

            System.out.println("Sayı girişi: çıkış için 0 girin");

            sayi=klavye.nextInt();

        }

        System.out.println("Toplam = " + top);

    }

}
```

```
Sıfırdan farklı sayı girin
5
Sayı girişi: çıkış için 0 girin
6
Sayı girişi: çıkış için 0 girin
7
Sayı girişi: çıkış için 0 girin
0
Toplam = 18
```

# NOT

Bir döngü kontrolünde eşitlik karşılaştırması için noktalı sayı değerleri kullanmayınız. Noktalı sayı değerleri bazı değerlerin tahmini olduğundan bunların kullanılması yanlış sayma veya doğru olmayan sonuçlara yol açabilir.  $1 + 0.9 + 0.8 + \dots + 0.1$  hesaplamak için aşağıdaki kodu düşünelim :

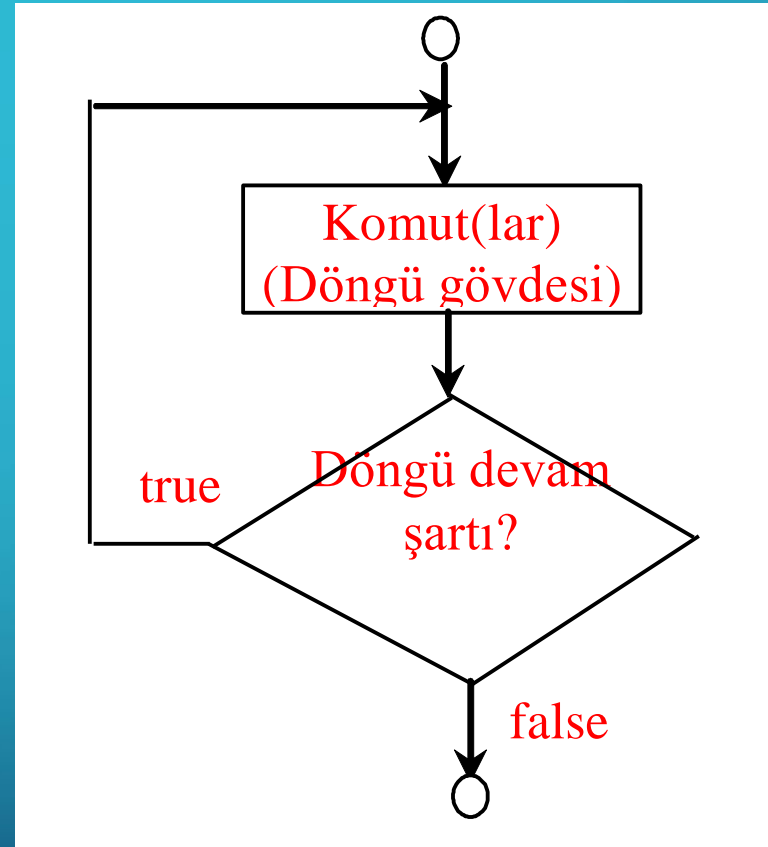
```
double item = 1; double sum = 0;
while (item != 0) { // No guarantee item will be 0
    sum += item;
    item -= 0.1;
}
```

```
System.out.println(sum);
```

item değişkeni 1 ile başlar ve her defasında 0.1 azaltılır. Döngü item 0 olunca sonlanır. Fakat item değişkeninin tam 0 olması garanti edilemez. Döngü görünüşte sonsuz olmasına rağmen aslında sonsuz bir döngüdür.

# DO-WHILE DÖNGÜSÜ

```
do {  
    // Döngü gövdesi;  
    Komut(lar);  
} while (Döngü-devam-şartı);
```



# DO-WHILE ÖRNEK

```
int sayac = 1;
int sayi = 5;
do //1'den 5'e kadar sayıları bir satırda göster
{
    System.out.print(sayac + " ");
    sayac++;
} while (sayac <= sayi);
```

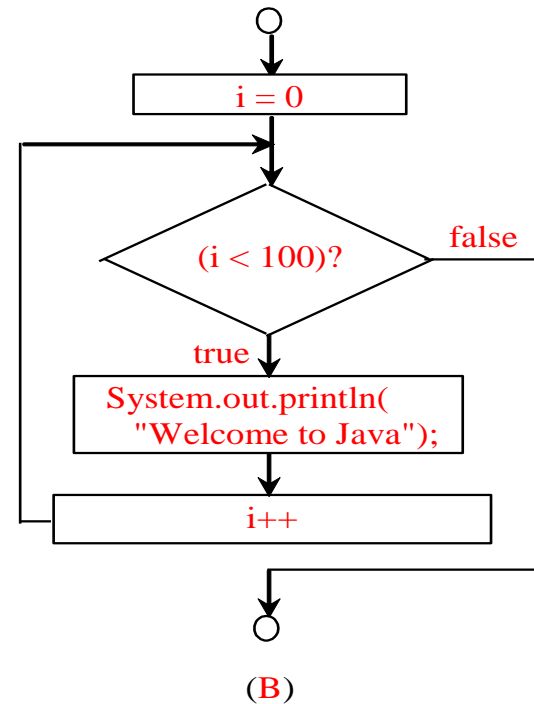
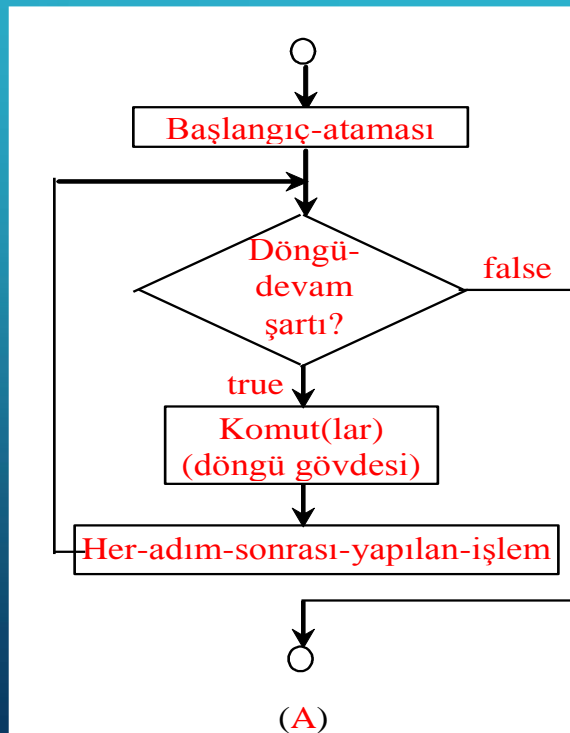
Çıktı:

1 2 3 4 5

# FOR DÖNGÜLERİ

```
for (başlangıç-ataması;  
    döngü-devam-şartı; her-  
    adım-sonrası-yapılacak-  
    işlem) {  
    // döngü gövdesi;  
    komut(lar);  
}
```

```
int i;  
for (i = 0; i < 100; i++) {  
    System.out.println(  
        "Merhaba Java!");  
}
```



# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println(  
        "Merhaba Java!");  
}
```

i değişkenini tanımla



# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println(  
        "Merhaba Java!");  
}
```

Başlatıcıyı çalıştır  
i değişkeni şimdi 0

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println( "Merhaba Java!");  
}
```

(i < 2) true  
çünkü i = 0

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

Merhaba Java yaz

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

Ayarlama komutunu çalıştır  
i = 1

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

( $i < 2$ ) halen doğru  
Çünkü  $i = 1$

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

Merhaba Java yaz

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

Ayarlama komutunu çalıştır  
i şimdi 2

# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

( $i < 2$ ) şartı false  
çünkü  $i = 2$



# FOR DÖNGÜSÜ ÖRNEK

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Merhaba Java!");  
}
```

Döngüden çık. Döngüden sonraki komutu çalıştır

# FOR ÖRNEK

- 3den 1e kadar say

```
for(int sayac = 3; sayac >= 1; sayac--)  
{  
    System.out.print("T = " + sayac);  
    System.out.println("ve sayiyor");  
}
```

```
System.out.println("Son!");
```

Çıktı:

```
T = 3 ve sayiyor  
T = 2 ve sayiyor  
T = 1 ve sayiyor  
Son!
```

# NOT

Bir for döngüsünde başlangıç-ataması sıfır veya daha çok virgül ile ayrılabilir. Aynı şekilde şart ve döngü sonrası işlem aşamaları da virgül ile ayrılabilir. Böylece aşağıdaki iki döngü kullanımı doğrudur.

```
for (int i = 1; i < 100; System.out.println(i++));
```

```
for (int i = 0, j = 0; (i + j < 10); i++, j++) {
```

```
    // döngü gövdesi
```

```
}
```

# NOT

Bir for döngüsünde döngü-devam-şartı yazılmazsa, döngü sürekli doğru olur. Böylece aşağıdaki şekilde verilen sonsuz bir döngü oluşur. Bu döngü while ile de yapılabilir.

```
for ( ; ; ) {  
    // Döngü gövdesi  
}
```

(a)

Eşdeğer

```
while (true) {  
    // Döngü gövdesi  
}
```

(b)

# DİKKAT


For döngüsünde döngü gövdesinden önce noktalı virgül koymak ortak bir hatadır.

Mantıksal  
hata


```
for (int i=0; i<10; i++);  
{  
    System.out.println("i is " + i);  
}
```

# DİKKAT

Benzer şekilde aşağıdaki döngü de yanlıştır :

```
int i=0;  
while (i < 10);  Mantıksal hata  
{  
    System.out.println("i is " + i);  
    i++;  
}
```

do-while döngüsünde ise aşağıdaki gibi döngü sonunda noktalı virgöl koymak gerekir.

```
int i=0;  
do {  
    System.out.println("i is " + i);  
    i++;  
} while (i<10);  Doğru
```

# HANGİ DÖNGÜ KULLANILMALI?

Döngü komutlarının üç formu vardır: while, do-while, ve for. Bu döngülerden özellikle for ve while birbirini yerine kullanılabilir. Örneğin aşağıda (a)'da verilen while (b)'deki gibi for ile yazılabilir.

```
while (döngü-devam-şartı) {  
    // Döngü gövdesi  
}
```

(a)

Eşdeğer

```
for ( ; döngü devam şartı; ) {  
    // Döngü gövdesi  
}
```

(b)

Bir for döngüsü aşağıda verilen şekilde while döngüsüne dönüştürülür.

```
for (başlangıç-ataması;  
     döngü-devam-şartı;  
     her-adım-sonrası-işlem) {  
    // Döngü gövdesi;  
}
```

(a)

Eşdeğer

```
Başlangıç-ataması;  
while (döngü-devam-şartı) {  
    // Döngü-gövdesi;  
    Her-adım-sonrası-işlem;  
}
```

(b)

# İÇ İÇE DÖNGÜLER

- Bir döngü gövdesinde başka bir döngü de olmak üzere her çeşit komut olabilir.

```
for (satir = 0; satir < 4; satir++)
```

```
    for (yildiz = 0; yildiz < 5; yildiz ++)
```

```
        System.out.print('*');
```

```
    System.out.println();
```

Dış döngü  
gövdesi

İç döngü  
gövdesi

- Dış döngünün bir defa yürütülmesine karşılık, iç döngü 5 defa çalıştırılır.

Çıktı:

```
*****
*****
*****
*****
```



# ÖRNEK: ÇARPIM TABLOSU

```
import java.util.Scanner;
public class Uygulama1 {
    public static void main(String[] args) {
        int i,j;
        System.out.println("ÇARPIM TABLOSU");
        System.out.print(" ");
        for(i=1;i<=9;i++)
            System.out.printf("%4d", i );
        System.out.println("\n-----");
        for (int k = 1; k < 10; k++) {
            System.out.print(k+"|");
            for (int l = 1; l < 10; l++) {
                System.out.printf("%4d",k*l);
            }
            System.out.println("");
        }
    }
}
```

ÇARPIM TABLOSU									
	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

## DÖNGÜLERLE İLGİLİ ÖRNEKLER

- 1- For döngüsü kullanarak ekrana 5 defa BMU 111 yazdırın:

```
public static void main(String[] args) {  
    for (int i = 0; i < 5; i++) {  
        System.out.println("BMU 111");  
    }  
}
```

- 2- Yukarıdaki örneği bir sayaç kullanarak, while döngüsü ile yapın:

```
public static void main(String[] args) {  
    int sayac = 0;  
    while (sayac < 5){  
        System.out.println("BMU111");  
        sayac++;  
    }  
}
```

- 3- For döngüsü kullanarak 32-256 arasındaki ASCII karakterleri ekrana yazdırın:

```
public static void main(String[] args) {  
    for (int i = 32; i < 256; i++) {  
        char c = (char) i;  
        System.out.println(c);  
    }  
}
```

- 4- Aşağıdaki şekilleri iç içe for döngüleri ile elde edin:

a)

```
* * * *  
* * * *  
* * * *
```

```
public static void main(String[] args) {  
    int sayac = 0;  
    for (int i = 0; i < 3; i++) {  
        System.out.println("");  
        for (int j = 0; j < 4; j++) {  
            System.out.print("* ");  
            sayac++;  
        }  
    }  
}
```

b)

\*  
\*\*  
\*\*\*  
\*\*\*\*

```
public static void main(String[] args) {  
    int sayac = 0;  
    for (int i = 0; i < 5; i++) {  
        for (int j = 0; j < i; j++) {  
            System.out.print("*");  
            sayac++;  
        }  
        System.out.println(" ");  
    }  
}
```

5- 0 ile 100 arasındaki sayıların toplamını for döngüsü kullanarak bulunuz:

```
public static void main(String[] args) {  
    int toplam = 0;  
    for (int i = 0; i <= 100; i++) {  
        toplam += i;  
    }  
    System.out.println("TOPLAM = " + toplam);  
}
```

6- Aynı örneği while döngüsü ile yazınız:

```
public static void main(String[] args) {  
    int toplam = 0, sayac=0;  
    while(sayac<=100){  
        toplam += sayac;  
        sayac++;  
    }  
    System.out.println("TOPLAM = " + toplam);  
}
```

7- 0 ile 100 arasındaki tek ve çift sayıların toplamalarını ayrı ayrı ekrana yazdıran bir program:

```
public static void main(String[] args) {  
    int sayac=0, tekSayac=0, ciftSayac=0;  
    while(sayac<=100){
```

```

        if(sayac%2==0)
            ciftSayac+=sayac;
        else
            tekSayac+=sayac;
        sayac++;
    }
    System.out.println("Tek sayılar toplamı = " + tekSayac +
        "\nÇift Sayılar toplamı = " + ciftSayac);
}

```

- 8- While döngüsü kullanarak 0 ile 1000 arasında kaç tane tek, kaç tane çift sayı olduğunu bulup ekrana yazdırınız:

```

public static void main(String[] args) {
    int sayac = 0, tekSayac = 0, ciftSayac = 0;
    while (sayac <= 1000) {
        if (sayac % 2 == 0) {
            ciftSayac ++;
        } else {
            tekSayac ++;
        }
        sayac++;
    }
    System.out.println("Tek sayıların adedi= " + tekSayac +
        "\nÇift sayıların adedi = " + ciftSayac);
}

```

- 9- For döngüsü kullanarak "Firat Universitesi Bilgisayar Mühendisliği" cümlesindeki a ve i harflerinin adedini bulan bir program yazınız:

```

public static void main(String[] args) {
    int iSayac = 0, aSayac = 0;
    String cumle = "Firat Universitesi Bilgisayar Mühendisliği";

    for (int i = 0; i < cumle.length(); i++) {
        char c = cumle.charAt(i);
        if (c == 'i') {
            iSayac++;
        }
        if (c == 'a') {
            aSayac++;
        }
    }
    System.out.println("iSayac = " + iSayac + "\naSayac = " + aSayac);
}

```

10-  $\sum 3x^2 + 2x + 5$  denkleminin 0-100 arasındaki sonucunu bulun:

```
public static void main(String[] args) {  
    int sonuc = 0;  
    for (int x = 0; x < 100; x++) {  
        sonuc += 3 * (x*x) + 2 * x + 5;  
    }  
    System.out.println("sonuc = " + sonuc);  
}
```