

# Sözde Kod (Pseudo-code) & Algoritmalar Arasında Dönüşüm

**Dr. Fatih KALEMKUŞ**

*Kafkas Üniversitesi*

# Sözde Kod

Problemi çözmek için tasarladığımız algoritmaları kodlamamız gerekir. Bu kodlama herhangi bir programlama dilinde (C, C++, Java, Python vb.) olabileceği gibi, eğer algoritmayı hemen çalıştırmaya ihtiyacımız yoksa ara bir yapı olan sözde koda da dönüştürebiliriz.

## **Sözde Kod:**

Bilgisayarda bir programlama dili olarak çalışmayan, ancak yazı/konuşma dilinden ziyade programlama dillerine daha yakın olan algoritma ifadelerine **sözde kod (pseudo-code)** denilir.

Sözde kodların İngilizce ifadelerle belirtilmesi dünyada yaygın olarak kabul görmektedir.

# Sözde Kod

Bir sözde kod, yapısal olarak dört temel öğeye sahiptir. Bunlar;

## 1. Okuma/Yazma:

READ, GET, WRITE, DISPLAY gibi komutlarla temel okuma ve yazma işlemleri gerçekleştirilir. *Hangisi ne zaman kullanılır?*

## 2. İşlemler:

Sözde kod içinde gerçekleştirilen toplama, çıkartma, bölme, vb. Aritmetik ve diğer işlemler ve bir değişkene değer atanması gibi olaylardır.

## 3. Karar Yapıları:

Bir koşulu kontrol edip, bir alternatifin işletilip işletilmeyeceğine veya birden fazla alternatiften hangisinin işletileceğine karar veren mekanizmalardır.

# Sözde Kod

**3.1. Basit Karar Yapısı:** Bir koşula bağlı olarak, bir alternatifi yapıp yapılmayacağına karar verir.

```
IF [koşul] THEN  
    Koşul doğru (true) ise gerçekleşecek işlemler  
ENDIF
```

**3.2. İki Alternatifli Karar Yapısı:** Koşula uyan durumda bir alternatifi, uymayan durumda diğer alternatifi işletir.

```
IF [koşul] THEN  
    Koşul doğru (true) ise gerçekleşecek işlemler  
ELSE  
    Koşul yanlış (false) ise gerçekleşecek işlemler  
ENDIF
```

# Sözde Kod

## 3.3. Çok Alternatifli Karar Yapısı: Birden fazla koşul deyimi içeren karar yapısıdır.

```
IF [koşul1] THEN
    Koşul1 doğru ise gerçekleşecek işlemler
ELSEIF [koşul2] THEN
    Koşul1 yanlış, Koşul2 doğru ise gerçekleşecek işlemler
ELSE
    Koşul1 ve Koşul2 yanlış ise gerçekleşecek işlemler
ENDIF
```

**SORU:**

*Yukarıdaki yazım ile sağdaki arasında ne fark vardır?*

```
IF [koşul1] THEN
    Koşul1 doğru ise gerçekleşecek işlemler
    IF [koşul2] THEN
        Koşul 1 ve Koşul2 doğru ise gerçekleşecek işlemler
    ELSE
        Koşul1 doğru ve Koşul2 yanlış ise gerçekleşecek işlemler
    ENDIF
ENDIF
```

# Sözde Kod

## Örnek:

BiletKart uygulaması için metroda uygulanan ücret tarifesi aşağıdaki gibidir. Buna göre, uygulanan ücret politikasının algoritmasını oluşturunuz.

- Normal tarife: 3 TL
- Öğrenci ve 30 (dahil) yaşından küçük olanlar: 2.5 TL
- Öğrenci ve 30 yaşından büyük olanlar: 2.75 TL
- 60 (dahil) yaşından büyük olanlar: Ücretsiz

**NOT:** Bir kişi her iki koşulu birden taşıması durumunda, daha düşük olan ücret tarifesi uygulanır.

# Sözde Kod

## 4. Tekrarlı Yapılar:

Program içinde bir koşula bağlı olarak ya da belirli bir sayıda tekrar edecek işlemler için kullanılır.

Koşula bağlı tekrarlı yapılarda, koşul bazen tekrarın girişine, bazen de sonuna uygulanır. Aşağıdaki yapıda koşul tekrarlı yapının girişinde uygulanmıştır. Bu durumda eğer koşul *doğru (true)* değeri vermezse, koşul içindeki tekrarlanacak işlemler hiç gerçekleşmez.

```
LOOP [koşul]
    ...Tekrarlanacak işlemler
ENDLOOP
```

**\*\*\* NOT: BREAK komutu, içinde bulunduğu döngüyü kırar.**

# Sözde Kod

Aşağıdaki koşula bağlı tekrarlı yapıdaysa, koşulun durumu her ne olursa olsun, tekrarlı yapı içindeki kod en az bir kere çalıştırılacaktır.

```
LOOP
  ...Tekrarlanacak işlemler
ENDLOOP [koşul]
```

**Sayaç tipi tekrarlı yapılarda**, belirli bir sayıdan başlanarak, belirli bir hedefe kadar sayılır. Sayma işleminde artışın ne kadar olacağını **STEP** deyimi belirtir.

```
FOR Sayac = [başlangıç değeri] TO [Hedef Sayı Sayı] STEP [artış]
  ...Tekrarlanacak işlemler
ENDFOR
```

**\*\*\* 17'ye tam bölünebilen en büyük üç basamaklı sayıyı LOOP veya FOR döngüsü kullanarak bulan ve ekrana yazdıran programın algoritmasını sözde kod (pseudo-code) olarak yazınız.**

# Algoritmalar Arasında Dönüşüm

## Satır Algoritmalarından Sözde Kod Oluşturmak

Bir satır algoritmayı sözde koda dönüştürürken aşağıdaki adımları izleriz:

- Girdi ve çıktılar (değişkenler) belirlenir
- Sıralı adımlar, karar yapıları, tekrarlı yapılar ve işlemler belirlenir
- Yapı, işlem ve adımlar uygun şekilde birleştirilir.

```
1. Başla  
2. Yaz 1  
3. Yaz 2  
4. Yaz 3  
5. Dur
```



```
DISPLAY 1  
DISPLAY 2  
DISPLAY 3
```

# Satır Algoritmalarından Söзде Kod Oluşturmak

İki sayıyı alıp, bunları toplayarak toplamı ekrana yazdıran algoritmanın satır kodu ve söзде kodu:

```
1. Başla  
2. Oku (A,B)  
3. C=A+B  
4. Yaz C  
5. Dur
```



```
GET A  
GET B  
C=A+B  
DISPLAY C
```

# Satır Algoritmalarından Söзде Kod Oluşturmak

1. Başla
2. Yaz "Yaşınızı Giriniz"
3. Oku (Yas)
4. Eğer
  - 4.1.  $Yas \geq 18$  ise Yaz "Uygulamayı İndirebilirsiniz"
  - 4.2. Değilse ise Yaz "Uygulamayı İndiremezsiniz"
5. Dur



```
DISPLAY "Yaşınızı Giriniz"  
GET Yas  
IF Yas >= 18 THEN  
    DISPLAY "Uygulamayı İndirebilirsiniz"  
ELSE  
    DISPLAY "Uygulamayı İndiremezsiniz"  
ENDIF
```

# Satır Algoritmalarından Sözcük Kod Oluşturmak

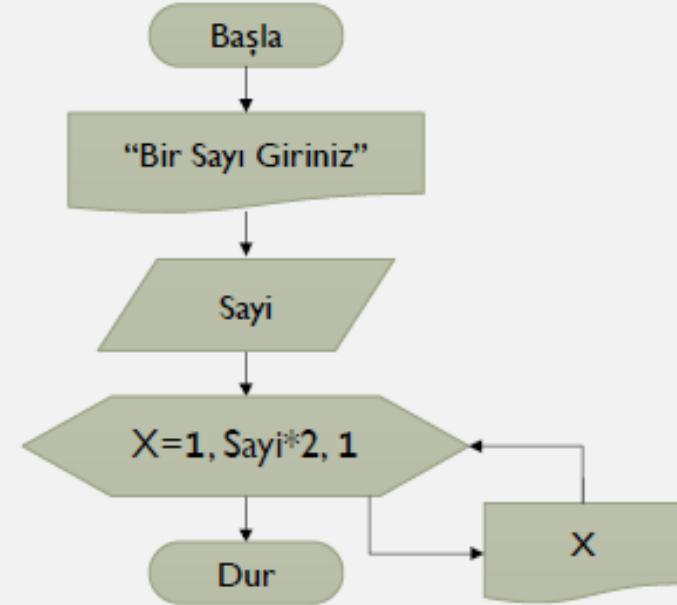
```
1. Başla
2. Toplam=0, Sayi=1
3. DÖNGÜ (X=1 TO 100 STEP 1)
4. Yaz "Bir Sayı Girin"
5. Oku Sayı
6. Toplam=Toplam+Sayı
7. DÖNGÜSONU
8. Yaz Toplam
5. Dur
```



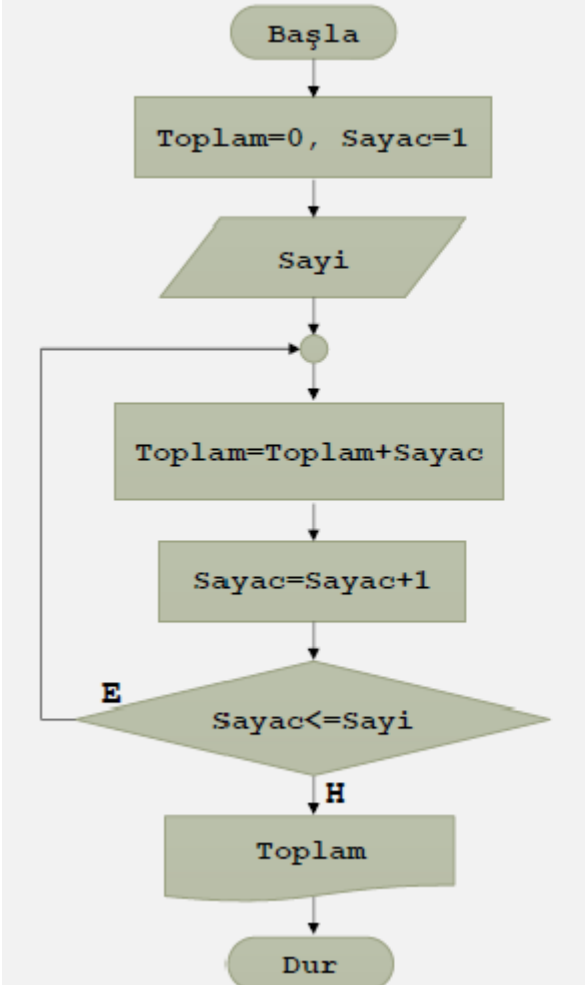
```
Toplam=0
FOR(X=1 TO 100 STEP 1)
  DISPLAY "Bir Sayı Girin"
  GET Sayı
  Toplam = Toplam + Sayı
ENDFOR
DISPLAY Toplam
```

# Satır Algoritmalarından Akış Diyagramı Oluşturmak

1. Başla
2. Yaz "Bir Sayı Giriniz"
3. Oku Sayı
4. Döngü (X=1 TO Sayı\*2 STEP 1)
5. Yaz X
6. DöngüSonu
7. Dur



# Akış Diyagramından Sözde Kod Oluşturmak

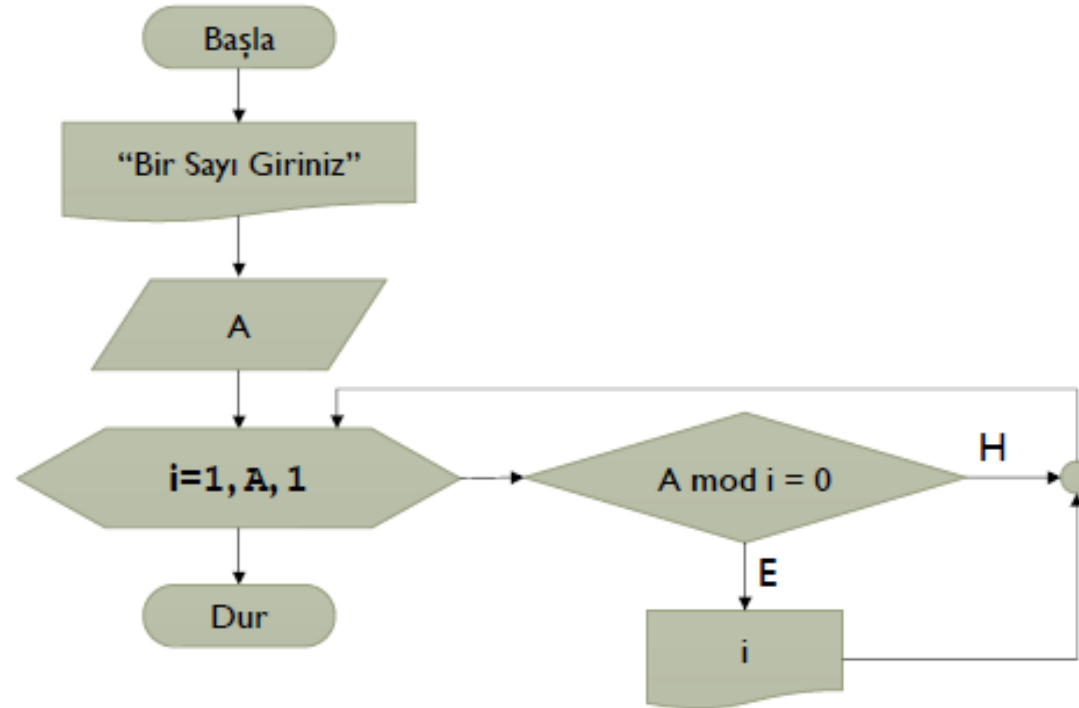


## Akış Diyagramlarından Sözde Kod Oluşturmak

```
Toplam=0  
GET Sayi  
FOR Sayac=1 TO Sayi STEP 1  
    Toplam=Toplam+Sayac  
ENDFOR  
DISPLAY Toplam
```

# Örnek

- Klavyeden girilen pozitif bir A tamsayısının tam bölenlerini hesaplayıp listeleyen programı tasarlayarak
  - Satır algoritma
  - Akış diyagramı ve
  - Sözde kodolarak ifade ediniz.

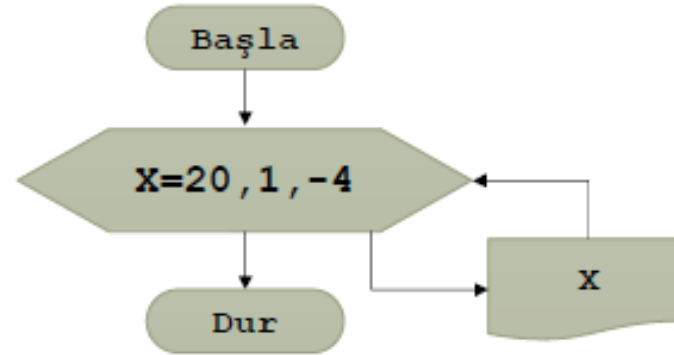


# Örnek

- 20'den başlayıp 1'e kadar, dörder dörder geriye doğru sayıp ekrana yazdıran algoritmayı tasarlayınız.

1. Başla
2. Döngü (X=20 TO 1 STEP -4)
3. Yaz X
4. DöngüSonu
5. Dur

```
FOR X=20 TO 1 STEP -4  
  DISPLAY X  
ENDFOR
```



# Çalışma Soruları

- Klavyeden girilen  $n$  sayısına göre;
  - 1'den  $n$ 'e kadar tamsayıların toplamını ( $t_1$ )
  - 1'den  $n$ 'e kadar tek tamsayıların toplamını ( $t_2$ )
  - 2'den  $n$ 'e kadar çift sayıların toplamını ( $t_3$ )

hesaplayan ve ekrana yazdıran programı tasarlayarak;

- Satır algoritma
- Akış diyagramı ve
- Söзде kod

olarak ifade ediniz (*Seçkin, s. 185*).

# Çalışma Soruları

- Klavyeden girilen A ve B sayıları arasındaki (sınırlar dahil) asal sayıları bulup ekrana yazdıran programı tasarlayıp;
  - Satır algoritma
  - Akış diyagramı ve
  - Söзде kodolarak ifade ediniz (*Kodlab, s. 166*).

# Çalışma Soruları

- Fibonacci sayı dizisi 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89... dizilişindeki sayılardan oluşan bir dizidir. Dizideki ilk iki sayı “0, 1” dir ve sonra gelen sayılar, kendisinden önceki iki sayının toplamıdır.

|                 |                 |           |
|-----------------|-----------------|-----------|
| 0               | 1               | 1         |
| İki önceki sayı | Bir önceki sayı | Yeni sayı |

- Fibonacci dizisinin ilk 10 elemanını hesaplayarak ekrana yazdıran programı tasarlayarak
  - Satır algoritma
  - Akış diyagramı ve
  - Sözde kodolarak ifade ediniz (*Kodlab, s.160*).

# Sonuç



*Dr. Fatih KALEMKUŞ*

# Sorular



*Dr. Fatih KALEMKUŞ*

# TEŐEKKÜRLER

*Dr. Fatih KALEMKUŐ*