

# Hafta4

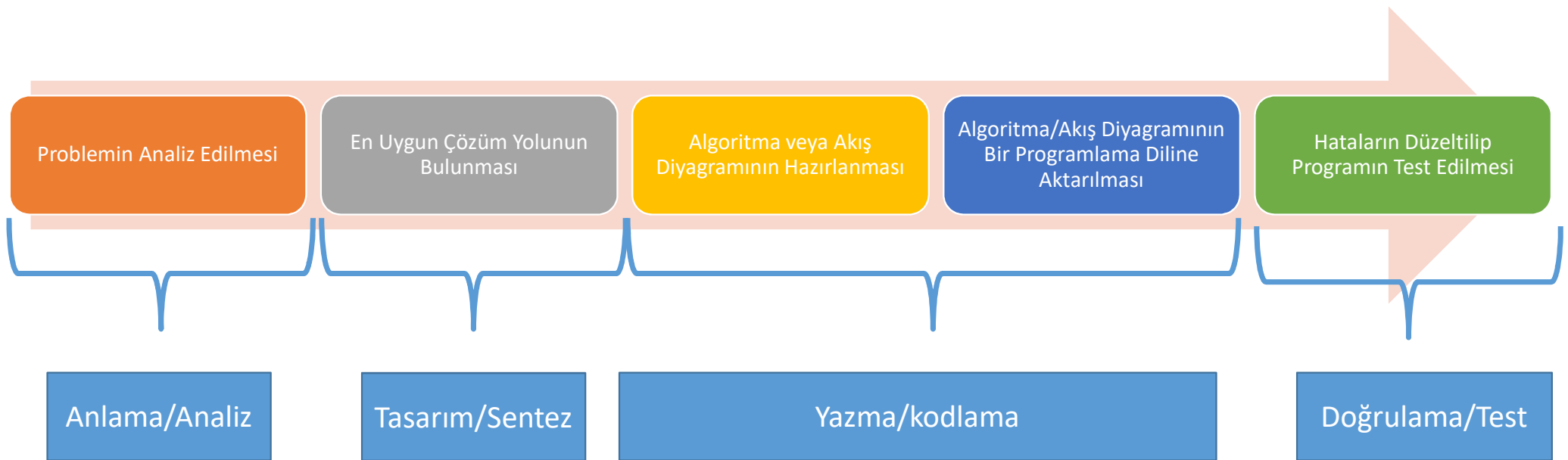
## Programlamaya Giriş

# Algoritma

(a)	(b)
<div data-bbox="607 596 996 769"><ol style="list-style-type: none"><li>1. Sabah kalk</li><li>2. Hazırlıklarını yap</li><li>3. Kahvaltı yap</li><li>4. Okula git</li><li>5. Derslere katıl</li><li>6. Eve dön</li></ol></div>	<div data-bbox="1151 542 1653 826"><ol style="list-style-type: none"><li>1. Sabah kalk</li><li>2. Hazırlıklarını yap</li><li>3. Kahvaltı malzemeleri var mı?<ol style="list-style-type: none"><li>3.1. Yoksa gidip al</li></ol></li><li>4. Kahvaltı yap</li><li>5. Otobüs durağına git</li><li>6. Otobüsü bekle</li><li>7. Otobüse binerek okula git</li><li>8. Derslere katıl</li><li>9. Eve dön</li></ol></div>
(c)	
<div data-bbox="768 925 1426 1321"><ol style="list-style-type: none"><li>1. Sabah kalk</li><li>2. Hazırlıklarını yap</li><li>3. Kahvaltı malzemeleri var mı?<ol style="list-style-type: none"><li>3.1. Yoksa gidip al</li></ol></li><li>4. Kahvaltı yap</li><li>5. Otobüs durağına git</li><li>6. Otobüsü bekle</li><li>7. Otobüs geldi mi?<ol style="list-style-type: none"><li>8.1. Geldiyse otobüse binerek okula git</li><li>8.2. Gelmediyse metro durağına git<ol style="list-style-type: none"><li>8.2.1. Metroyu bekle</li><li>8.2.2. Metroya binerek okula git</li></ol></li></ol></li><li>8. Derslere katıl</li><li>9. Eve dön</li></ol></div>	

# Algoritma

- Programlama mantığına uygun algoritmanın hazırlanması/ akış diyagramının çizilmesi programlamanın temelini oluşturur.



# Algoritma

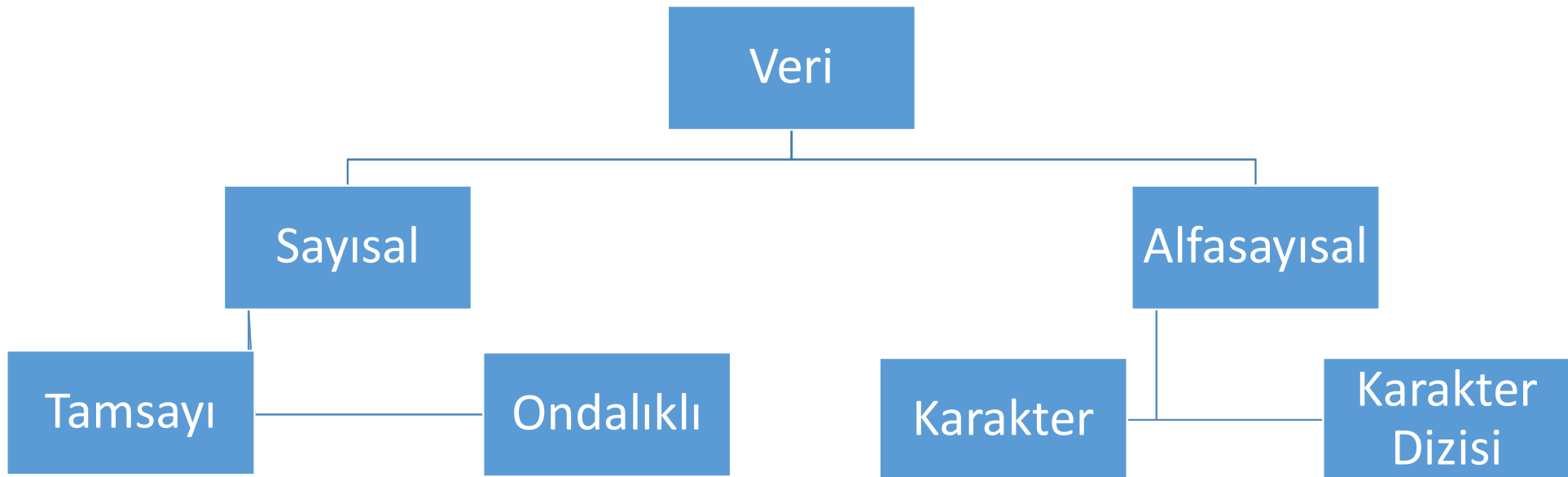
**Örnek-2.1:** Klavyeden girilen iki sayıyı toplayıp ekranda gösteren programın algoritmasının, üç değişik şekilde gösterilmesi.

Metin olarak yazılması	Sözde kodlarla yazılması	Akış diyagramının çizilmesi
<ol style="list-style-type: none"><li>1. Başla</li><li>2. Birinci sayıyı klavyeden okuyunuz (alınız, giriniz)</li><li>3. İkinci sayıyı klavyeden okuyunuz (alınız, giriniz)</li><li>4. Girilen/alınan sayıları toplayarak sonucu oluşturunuz</li><li>5. Sonucu, ekrana yazdırınız</li><li>6. Dur</li></ol>	<ol style="list-style-type: none"><li>1. Başla</li><li>2. A oku (al, gir)</li><li>3. B oku (al, gir)</li><li>4. <math>C=A+B</math></li><li>5. C yaz (göster)</li><li>6. Dur</li></ol>	<pre>graph TD; A([Başla]) --&gt; B[/A, B/]; B --&gt; C[C=A+B]; C --&gt; D[C]; D --&gt; E([Dur]);</pre>

# Algoritma

- Algoritmada Kullanılan Terimler
  - Veri
  - Tanımlayıcı
  - Değişken
  - Sabit
  - Aktarma
  - Sayaç
  - Döngü
  - V.b.

# Algoritma



# Algoritma

- Değişken
  - Program içerisinde farklı değerler alabilen/ aktarılan bellek alanı/veri/bilgi alanı olarak tanımlanabilir.

## Örnek-2.2:

1. Başla  
2. Birinci sayıyı (**A**) al  
3. İkinci sayıyı (**B**) al  
4. **C=A+B** işlemini yap  
5. Sonucu (**C**) yaz  
6. Dur

Yandaki algoritma, klavyeden (dışarıdan) girilen/okunan/alınan iki sayının toplamını yapıp sonucu ekrana yazdırır. Algoritmada 'A', 'B' ve 'C' değişkendir.

Program her çalıştırıldığında 'A' ve 'B' değişkenlerine farklı değerler girilir ve sonucu temsil eden 'C' de farklı değerler oluşur (Tablo 2.3).

**Tablo 2.3:** Örnek-2.2 için uygulama sonuçları

Girilen A sayısı	Girilen B sayısı	C=A+B	Yazılan C sayısı
-3	3	0	0
3	11	14	14
33	99	132	132
1	2	3	3
-100	200	100	100
50	-83	-33	-33

1. çalıştırma  
2. çalıştırma  
3. çalıştırma  
4. çalıştırma  
5. çalıştırma  
6. çalıştırma

# Algoritma

## Örnek-2.3:

1. Başla
2. Öğrencinin numarasını (**No**) al
3. Öğrencinin adını ve soyadını (**Adsoy**) al
4. Öğrencinin vize notunu (**Vize**) al
5. Öğrencinin final notunu (**Final**) al
6. **Ort** =  $0.3 * \text{Vize} + 0.7 * \text{Final}$
7. Numara (**No**) ve ortalamayı (**Ort**) yaz
8. Dur

**Tablo 2.4:** Örnek-2.3 için uygulama sonuçları

No	Adsoy	Vize	Final	Ort	Ekrana yazılan
9801.0033	Ahmet	30	70	58	9801.0033 – 58
9804.0099	Mehmet	70	90	84	9804.0099 – 84
9712.0123	Ali	60	30	39	9712.0123 – 39

1. çalıştırma

2. çalıştırma

3. çalıştırma



# Algoritma

- Sabit
  - Programlama içerisinde değeri değişmeyen veri( Veri alanları) olarak tanımlanabilir.

**Örnek-2.4:** Yarıçapı girilen dairenin çevresiyle alanını hesaplayıp yazdıran programın algoritması.

```
1. Başla
2. pi_sayisi=3.14
3. Dairenin yarıçapını (r) al
4. cevre=2*pi_sayisi*r
5. alan=pi_sayisi*r*r
6. Çevre(cevre) ve alanı(alan) yaz
7. Dur
```

# Algoritma

- Atama/Aktarma

```
1. Başla
2. T=0
3. Bir sayı (A) al
4. T=T+A işlemini yap
5. Başka bir sayı (B) al
6. T=T+B işlemini yap
7. Başka bir sayı (C) al
8. T=T+C işlemini yap
9. Yaz T
10. Dur
```

**Tablo 2.6:** Örnek-2.5 için çalıştırma sonuçları

İşlem sırası	A	B	C	Eski T	Yeni T
1	3	---	---	0	$0+3=3$
2	---	4	---	3	$3+4=7$
3	---	---	5	7	$7+5=12$
<i>T=12</i>					

# Algoritma

- Sayaç
  - Programlarda belirli sayıda tekrar etmesi gereken işlemlerde ya da üretilen değerlerin sayılması gibi işlemlerde kullanılan yapı olarak tanımlanabilir.

**Örnek-2.6:** Aşağıdaki algorithmada 1-5 arası sayılar, ekrana yazdırılmaktadır. 1-5 arası sayıları oluşturmak için sayaç ( $S=S+1$ ) kullanılmakta ve algoritmanın çalışma prensibi Tablo 2.7’de verilmektedir.

1. Başla
2. $S=0$
3. Eğer $S>4$ ise git 7
4. <b><math>S=S+1</math></b>
5. Yaz S
6. Git 3
7. Dur

**Tablo 2.7** Örnek-2.6’daki sayacın çalışması

Eski S	Yeni S	Ekrana yazılan
0	$0+1=1$	1
1	$1+1=2$	2
2	$2+1=3$	3
3	$3+1=4$	4
4	$4+1=5$	5

# Algoritma

- Döngü

- Programlarda belirli işlem bloklarını ardışık değerlerle belirli sayıda tekrar ederek çalıştırılmasını sağlayan yapılar olarak tanımlanabilir.

**Örnek-2.7:** Aşağıdaki algoritma 1-10 arası tek sayıların toplamını hesaplamakta ve çalışma prensibi Tablo 2.8’de verilmektedir.

```
1. Başla
2. T=0
3. J=1
4. Eğer J>10 ise git 8
5. T=T+J
6. J=J+2
7. Git 4
8. Yaz T
9. Dur
```

*Döngü*

**Tablo 2.8** Örnek-2.7’deki döngünün çalışması

Eski J	Eski T	Yeni T	Yeni J
1	0	$0+1=1$	3
3	1	$1+3=4$	5
5	4	$4+5=9$	7
7	9	$9+7=16$	9
9	16	$16+9=25$	11
11	-	-	-

# Algoritma

- Örnek
  - Klavyeden girilen 5 sayının ortalamasını bulan programın algoritmasını yazınız.

# Algoritma

- Örnek
  - Klavyeden girilen 5 sayının ortalamasını bulan programın algoritmasını yazınız.

```
1. Başla
2. N=5
3. T=0
4. S=0
5. Eğer S>N-1 ise git 10
6. S=S+1
7. Sayıyı (A) gir
8. T = T + A
9. Git 5
10. Ortalama=T/N
11. Yaz Ortalama
12. Dur
```

Tablo 2.9 Örnek-2.8 uygulama sonuçları

Eski S	Yeni S	Sayı	Eski T	Yeni T	Ort.
0					
1					
2					
3					
4					
Ortalama=					

# Algoritma

- Örnek
  - Klavyeden girilen N sayısının faktöriyelini bulan programın algoritmasını yazınız.

# Algoritma

- Örnek
  - Klavyeden girilen N sayısının faktöriyelini bulan programın algoritmasını yazınız.

```
1. Başla
2. N sayısını al
3. F=1
4. S=0
5. Eğer  $S > N-1$  ise git 9
6.  $S=S+1$ 
7.  $F=F*S$ 
8. Git 5
9. Yaz F
10. Dur
```

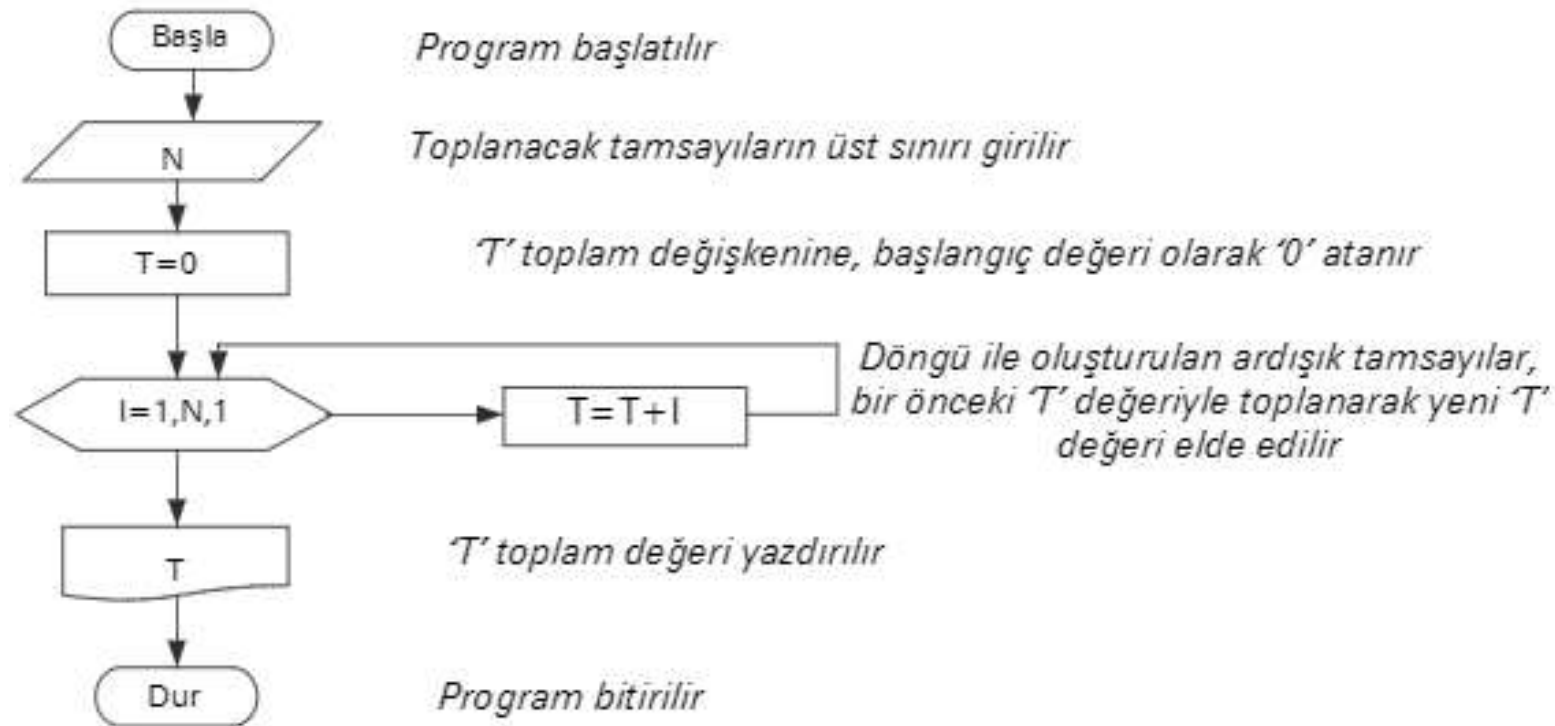
**Tablo 2.10:** Örnek-2.9 için uygulama sonuçları

Eski S	Yeni S	Eski F	Yeni F	Sonuç




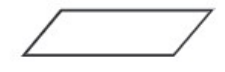


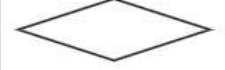



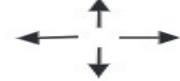
# Akış Diyagramı

- Algoritmanın özel geometrik şekiller kullanılarak ifade edilmesi olarak tanımlanabilir.



# Akış Diyagramı

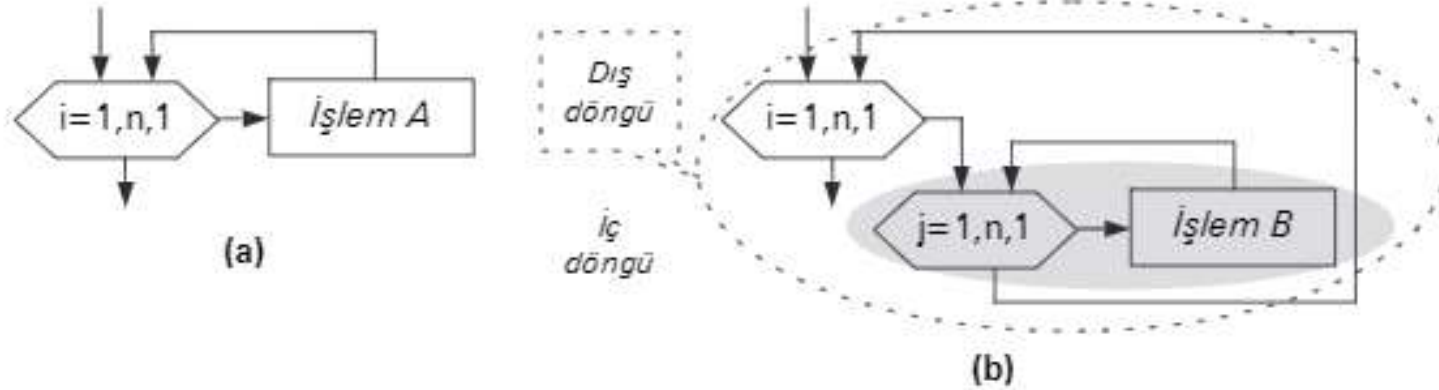
- Akış Diyagramında Kullanılan Şekiller

Şekil	Anlamı
	Başla / Dur
	Veri / Bilgi Girişi
	İşlem
	Döngü
	Karar / Karşılaştırma
	Veri / Bilgi Yazma
	Önceden Tanımlı İşlem ( Alt program / Fonksiyon)
	Bağlantı
	Akış Yönü

# Akış Diyagramı

- Döngü 

– İç içe Döngüler



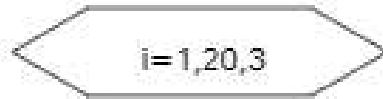
Şekil 3.9: (a) Tek döngü (b) İç içe iki döngü gösterimi

# Akış Diyagramı

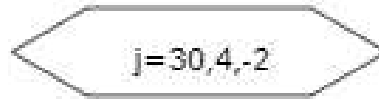
- Döngü



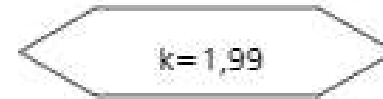
- Artan Döngü
- Azalan Döngü



*1'den 20'ye kadar 3'er artan 'i' döngüsü*



*30'dan 4'e kadar 2'şer azalan 'j' döngüsü*



*1'den 99'a kadar 1'er artan 'k' döngüsü*

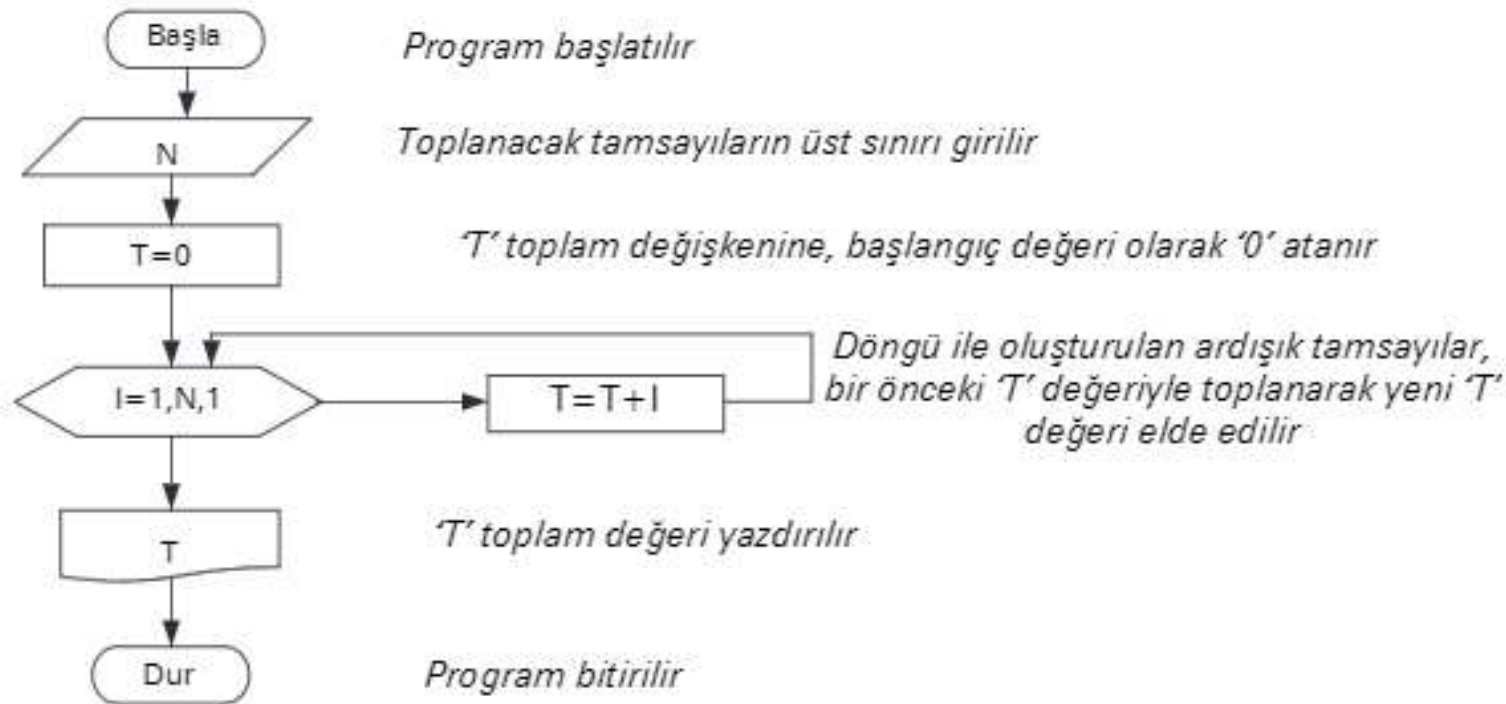
**Şekil 3.7:** Döngü örnekleri

# Akış Diyagramı

- 1'den  $N$ 'e kadar olan sayıların toplamını bulan programın akış diyagramını çiziniz.

# Akış Diyagramı

- 1'den N'e kadar olan sayıların toplamını bulan programın akış diyagramını çiziniz.



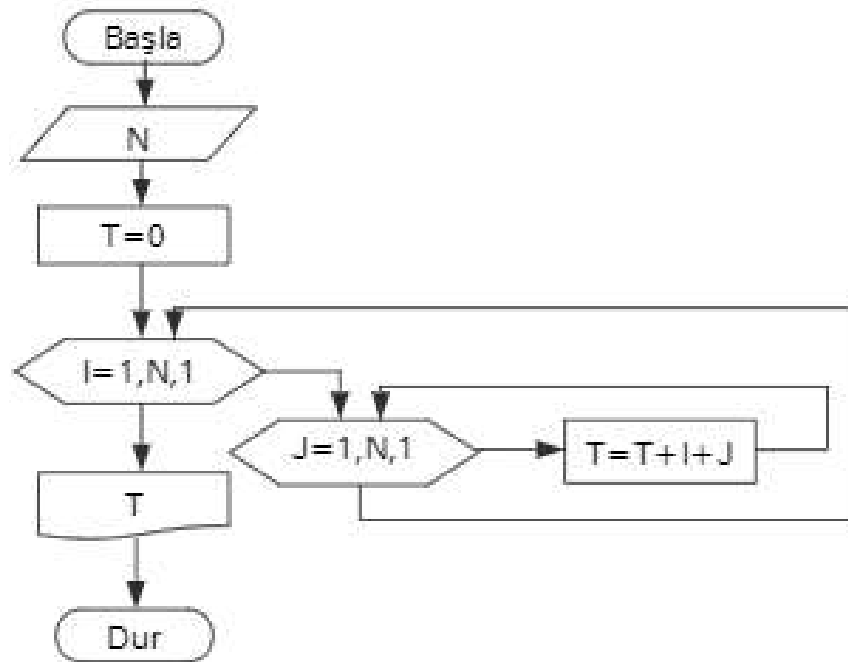
# Akış Diyagramı

- 1'den N'e kadar olan sayıların toplamını bulan programın akış diyagramını çiziniz.
  - Klavyeden 5 girilirse;

I	Eski T	Yeni T
1	0	$0+1=1$
2	1	$1+2=3$
3	3	$3+3=6$
4	6	$6+4=10$
5	10	$10+5=15$

# Akış Diyagramı

- İç içe iki döngü içeren akış diyagramının incelenmesi



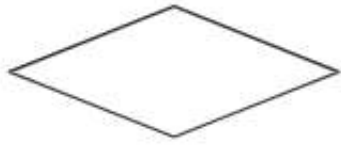
**Tablo 3.4:** Örnek-3.2'nin çalışma prensibi

I	J	Eski T	Yeni T
1	1	0	$0+1+1=2$
1	2	2	$2+1+2=5$
1	3	5	$5+1+3=9$
2	1	9	$9+2+1=12$
2	2	12	$12+2+2=16$
2	3	16	$16+2+3=21$
3	1	21	$21+3+1=25$
3	2	25	$25+3+2=30$
3	3	30	$30+3+3=36$



# Akış Diyagramı

- Karar / Karşılaştırma



Tablo 3.7: Karşılaştırma ifadelerinin genel kullanım şekilleri

Biçim	Akış diyagramı gösterimi	Açıklama
Yarım		"koşul" doğru ise (E) "işlem" yapılır, değilse (H) akış devam eder.
Tam		"koşul" doğru ise (E) "işlem-1", değilse (H) "işlem-2" yapılır.
Çok koşullu		"koşul-1" doğru ise (E) "işlem-1", değilse (H) "koşul-2" kontrol edilir ve doğru ise (E) "işlem-2" yapılır. "koşul-2" yanlış ise (H) sıradaki koşul kontrol edilir ve doğru olup olmamasına göre işlem devam eder. Eğer tüm koşullar yanlış ise "işlem-n" yapılır.

# Akış Diyagramı

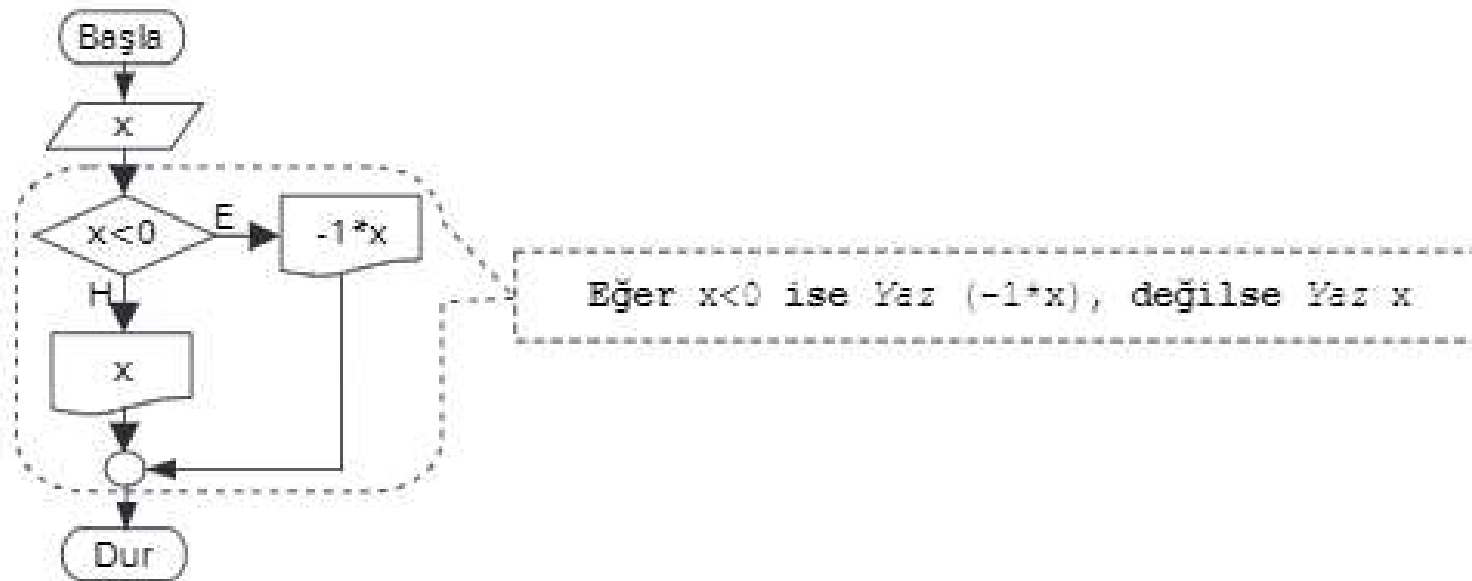
- Klavyeden girilen bir sayısının mutlak değerini ekrana yazdıran programın akış diyagramını çiziniz.
- Kural: Herhangi bir x sayısının mutlak değeri;

$$|x| = \begin{cases} -1.x & , \quad x < 0 \\ x & , \quad x \geq 0 \end{cases}$$

Şeklinde ifade edilir.

# Akış Diyagramı

- Klavyeden girilen bir sayısının mutlak değerini ekrana yazdıran programın akış diyagramını çiziniz.
- Kural: Herhangi bir x sayısının mutlak değeri;  $|x| = \begin{cases} -1 \cdot x & , \quad x < 0 \\ x & , \quad x \geq 0 \end{cases}$



## References

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ Bayram AKGÜL, C Programlama Ders notları