

BİLGİSAYAR BİLİMLERİNE GİRİŞ I

-4-

Program Tasarımı

2

□ Algoritmaların Gösterimi

- Algoritmaların herkes tarafından anlaşılır halde ifade edilebilmeleri için kullanılan iki temel gösterim biçimi vardır.
- **Kaba Kod (Pseudocode)**, belirli bir programlama diline, veri yapısına veya söz-dizim kuralına (syntax) bağlı kalmadan, adım adım yazılmış kodlara verilen isimdir.
- İfadesel bir gösterim biçimidir.
- Daha önce yazdığımız Türkçe adım algoritmaları birer kaba kod örneğidir.

1. Başla
2. Kullanıcıdan 2 doğal sayı al
3. İki sayıyı topla
4. Sonucu yazdır
5. Dur

Program Tasarımı

3

- **Akış Şeması (Flowchart)**, bir algoritmanın veya bir işlemin (process) çalışma sırasına, yani akışına uygun biçimde ve belirli anlamları olan simgelerin oklar ile bağlanması yoluyla oluşturulan şemalara verilen isimdir.

(A Flowchart is a graphical representation of an algorithm.)

- Akış diyagramı, akış çizelgesi, akış grafiği gibi isimleri de vardır.
- Grafikselsel bir gösterim biçimidir.
- Aslında akış şemaları yalnızca bilgisayar bilimlerinde değil, birçok farklı alanda kullanılan bir gösterim biçimidir. Ancak algoritmaların ifadesinde kullanılan akış şeması, klasik akış şemalarının özelleşmiş bir halidir.

Flowcharts – Akış Şemaları

4

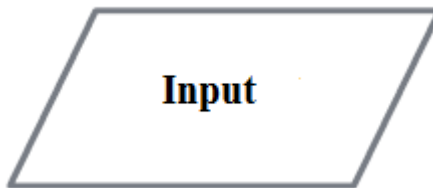


START/END

Denotes the beginning or end of the program

- Terminator (Sonlandırıcı)

- Başlangıç veya bitişi ifade eder.




Input

Denotes an input/output operation

- Input (Veri Girişi)

- Kullanıcı tarafından veri girileceğini gösterir.

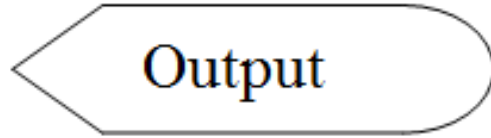


Print

For Print

- (Output) (Çıkış)

- Kullanıcıya veya ekrana mesaj verileceğini belirtir.




Output

Denotes an output operation


Flowcharts – Akış Şemaları

5



Process

Denotes a process to be carried out. addition, subtraction, division etc.



Desicion

Denotes a decision (or branch) to be made.

The program should continue along one of two routes. (e.g. IF/THEN/ELSE)

- Process (İşlem)

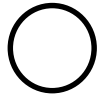
- Hesaplama yapıldığı veya veri işleme sürecini belirtir.

- Decision (Karar)

- Bir karşılaştırma, koşul ve sonucunda karar verileceğini ifade eder.

Flowcharts – Akış Şemaları

6



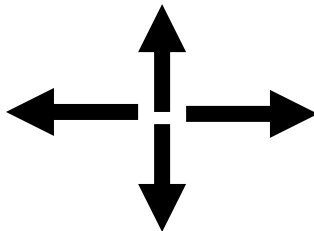
Allows the flowchart to be drawn without intersecting lines or without a reverse flow.

- Connector (Bağlayıcı)

- Bir akış şemasının farklı kısımlarının bağlanması için kullanılır.



veya



Denotes the direction of logic flow in the program

- Arrows, Directions (Oklar)

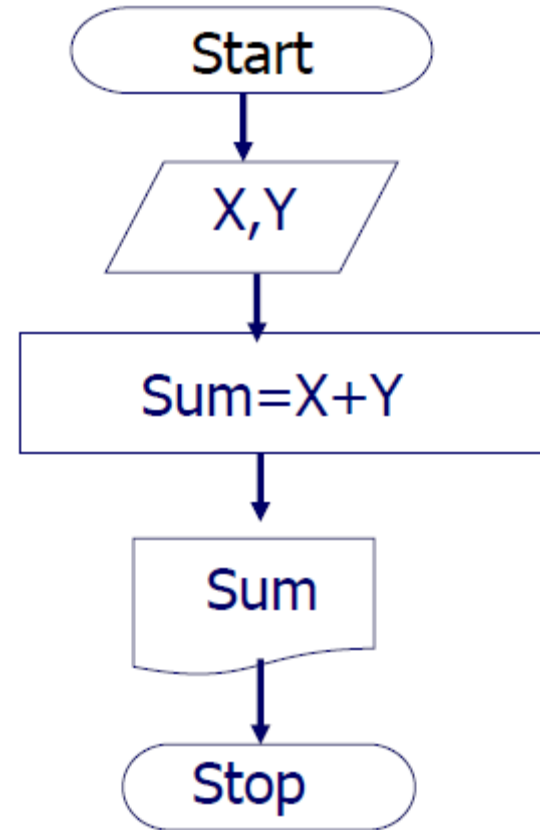
- Algoritmanın akış yönünü gösterir.

Örnek: İki sayının toplamını hesaplayan algoritma yazınız ve akış diyagramı çiziniz.

7

Pseudocode

1. START
2. Read X
3. Read Y
4. $\text{Sum} \leftarrow X + Y$
5. Print Sum
6. STOP.



Flowcharts – Akış Şemaları

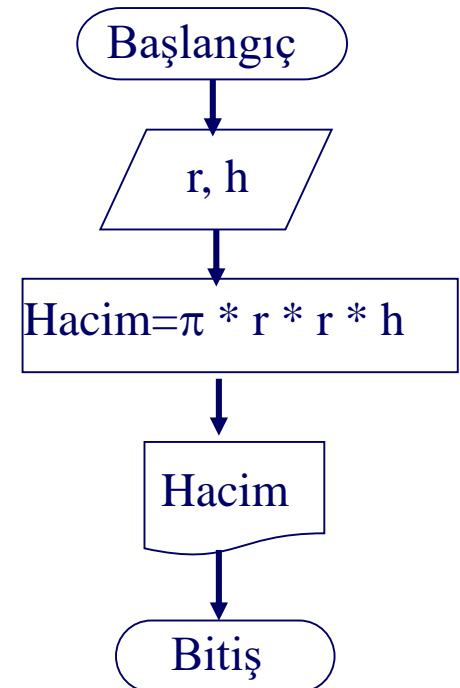
8

- Örnek:
- Gerekli parametreleri kullanıcı tarafından girilen bir silindirin hacmini hesaplayıp ekrana yazdırın.

Algoritma :

1. Başla.
2. Kullanıcı yarıçapı (r) ve yüksekliği (h) girsin.
3. Silindirin hacmini bulun.
$$\text{Hacim} = \pi \times r^2 \times h$$
4. Silindirin hacmini yazdırın.
5. Bitir

Akış şeması:



Flowcharts – Akış Şemaları

9

Örnek: Kullanıcı tarafından verilen değerlere göre kişinin Vücut Kitle Endeksi (Body-Mass Index-BMI) hesaplayan algoritma yazınız ve akış şemasını çizin.

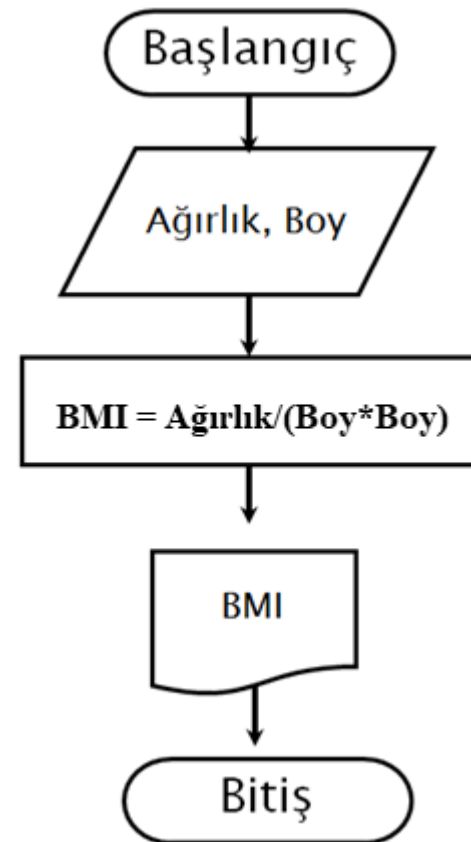
$$\text{BMI} = \text{Ağırlık}/(\text{boy}*\text{boy})$$

Flowcharts – Akış Şemaları

10

Örnek: Kullanıcı tarafından verilen değerlere göre kişinin Vücut Kitle Endeksi (Body-Mass Index-BMI) hesaplayan algoritma yazınız ve akış şemasını çiziniz.

1. Başla
2. Ağırlık, Boy al
3. $BMI = Ağırlık / (Boy * Boy)$
4. BMI yazdır.
5. Bitir



Flowcharts – Akış Şemaları

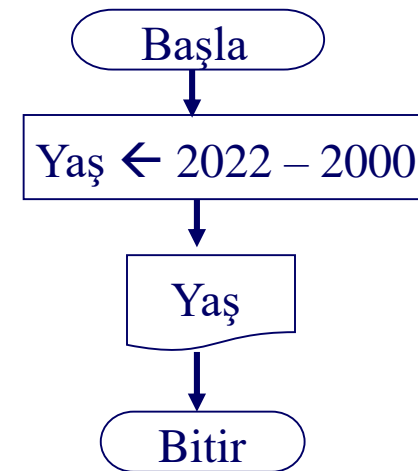
11

- **Örnek:** Doğum yılı 2000 olan bir kişinin yaşını bulan algoritmayı yazınız ve akış şemasını çiziniz.

Algoritma :

1. Başla
2. Yaşı hesapla ($\text{yaş} \leftarrow 2022 - 2000$)
3. Yaşı yazdır.
4. Bitir

Akış şeması :



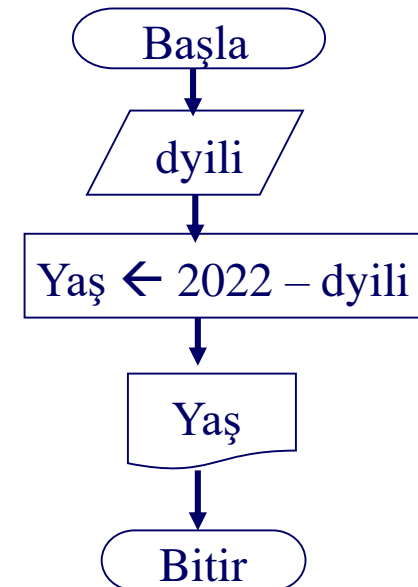
Flowcharts – Akış Şemaları

- ❑ **Örnek:** Doğum yılı kullanıcı tarafından verilen bir kişinin yaşını bulan algoritmayı yazınız ve akış şemasını çiziniz.

Algoritma :

1. Başla
2. Doğum yılını sor (dyili)
3. Yaşı hesapla ($\text{yaş} \leftarrow 2022 - \text{dyili}$)
4. Yaşı yazdır.
5. Bitir

Akış şeması :

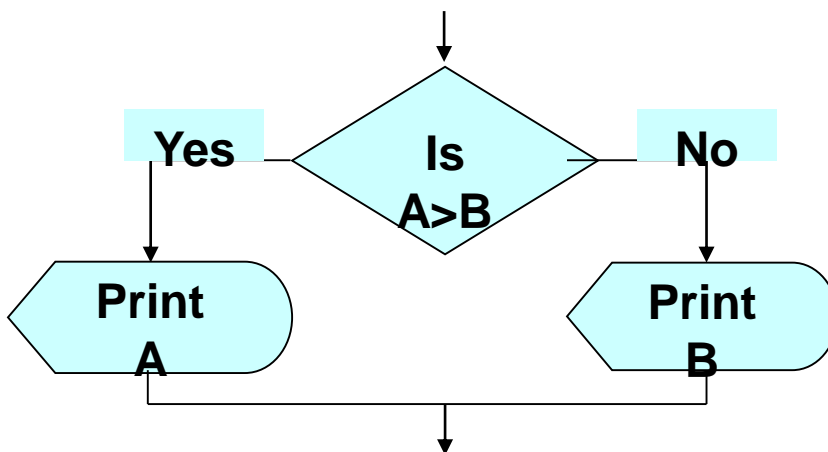


Karar Yapıları

13

□ Karar Yapıları

- ▣ $A > B$ bir mantıksal ifadedir.
- ▣ *Bu test etmek istediğimiz koşulu tanımlar.*
- ▣ **'if $A > B$ is true (Eğer A, B 'den büyükse.)'** Soldaki işlemi yapar; A 'yı yazarız.
- ▣ **'if $A > B$ is false (Eğer A, B 'den büyük değilse)'** Sağdaki işlemi yapar; B 'yi yazarız.



```
If A > B then
    print A
else
    print B
endif
```

İlişkisel Operatörler

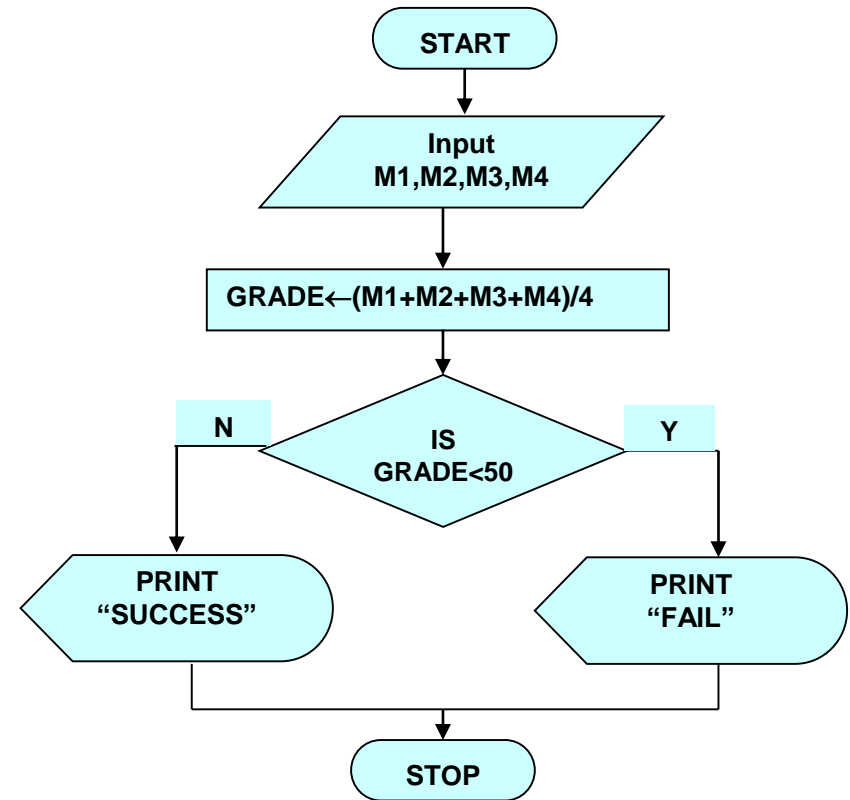
14

İlişkisel Operatörler	
Operatör	Tanım
>	Büyüktür
<	Küçüktür
==	Eşittir
>=	Büyük eşittir.
<=	Küçük eşittir.
<>	Eşit değildir.

Flowcharts – Akış Şemaları

15

- Yandaki akış şeması nasıl bir işlem yapmaktadır?



Flowcharts – Akış Şemaları

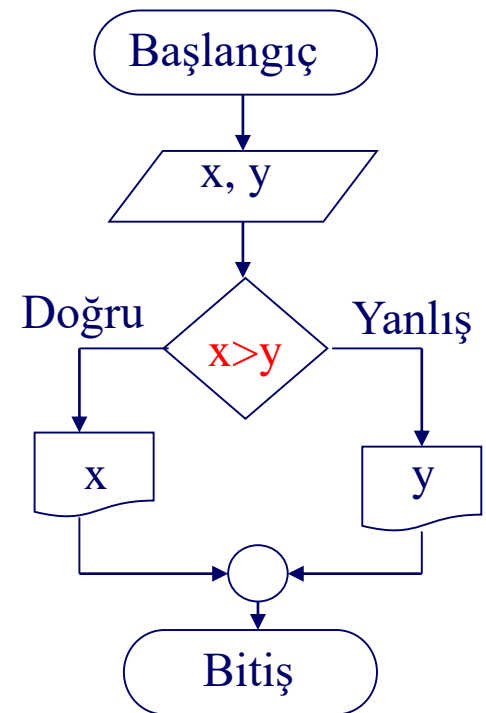
16

- **Örnek:** İki sayıdan **büyük** olanı bulan algoritmayı yazınız.

Algoritma :

1. Başla
2. Kullanıcı iki sayı girsin. (x, y)
3. Eğer x, y'den büyükse x'i yazdır, değilse y'yi yazdır.
4. Bitir

Akış Şeması :



Flowcharts – Akış Şemaları

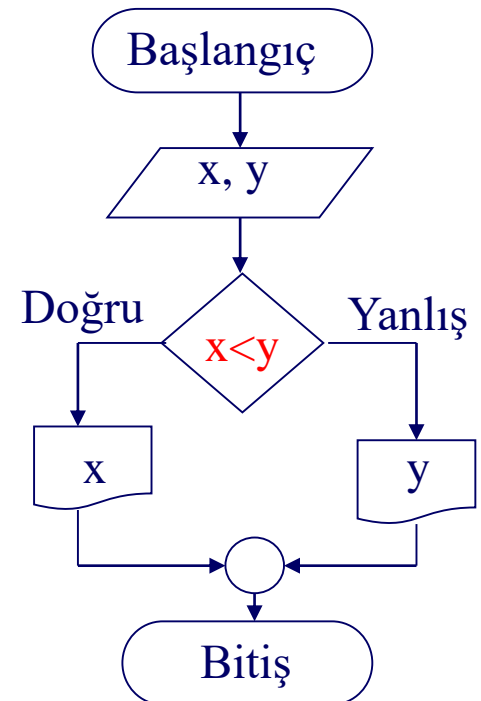
17

- **Örnek:** İki sayıdan **küçük** olanı bulan algoritmayı yazınız.

Algoritma :

1. Başla
2. Kullanıcı iki sayı girsin. (x, y)
3. Eğer x, y'den **küçükse** x'i yazdır, değilse y'yi yazdır.
4. Bitir

Akış Şeması :



Flowcharts – Akış Şemaları

18

- **Örnek:** Katsayıları verilen ikinci dereceden bir denklemin reel köklerini hesaplayan algoritmayı yazınız.

$$ax^2 + bx + c = 0$$

1. Başla
2. a , b ve c değerlerini al
3. Δ değerini hesapla ($\Delta = b^2 - 4ac$)
4. Eğer $\Delta < 0$ ise “Reel kök yok” mesajı ver
5. Eğer $\Delta = 0$ ise $x_{1,2} = -b/2a$
6. Eğer $\Delta > 0$ ise
$$x_1 = \frac{-b - \sqrt{\Delta}}{2a}$$
$$x_2 = \frac{-b + \sqrt{\Delta}}{2a}$$
7. Bitir.

Flowcharts – Akış Şemaları

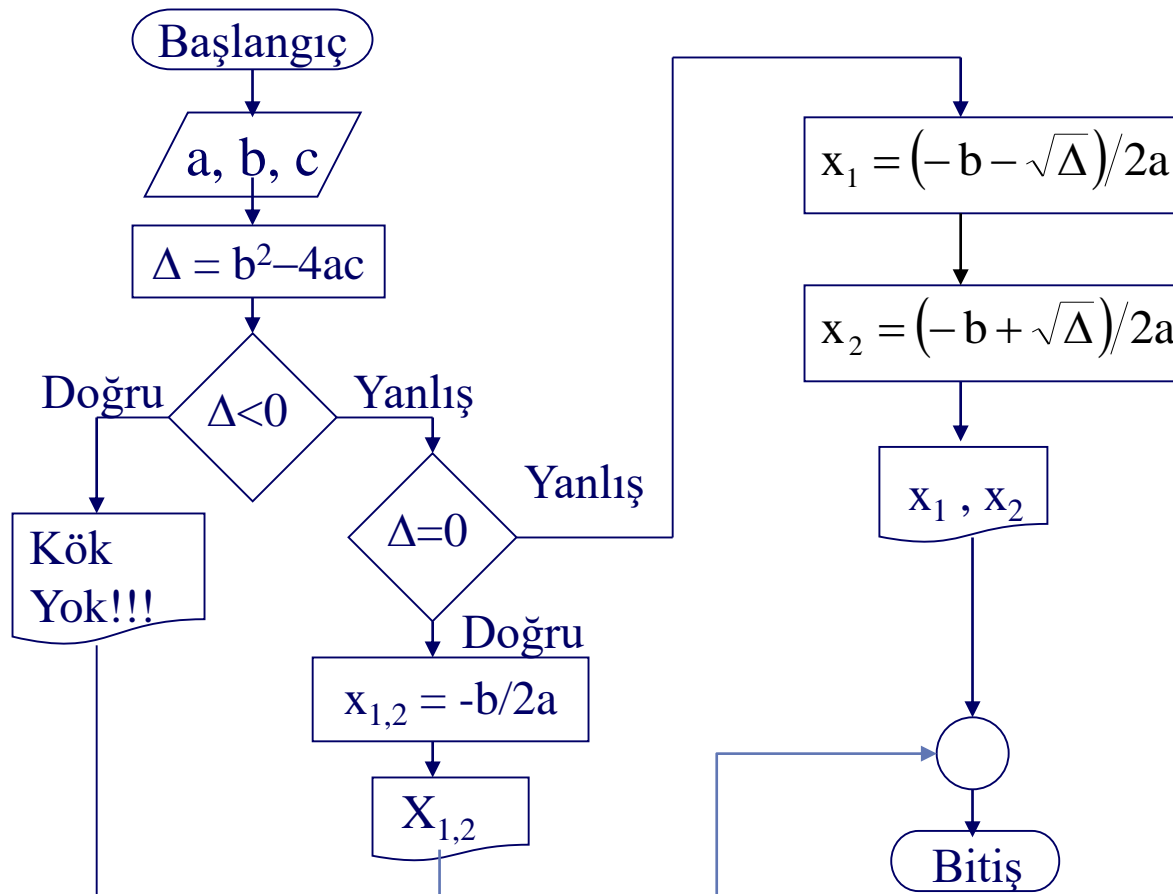
19

$$ax^2 + bx + c = 0$$

$$\Delta = b^2 - 4ac$$

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a}$$

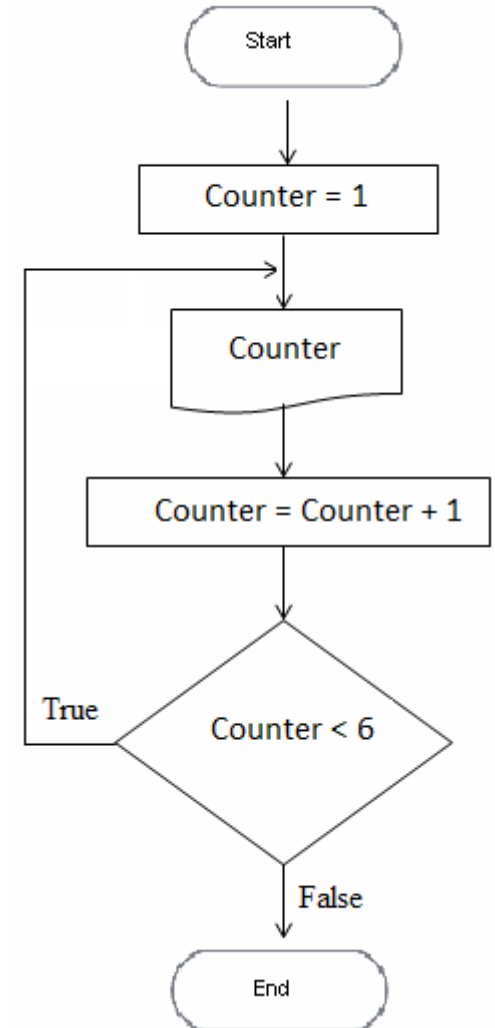
$$x_2 = \frac{-b + \sqrt{\Delta}}{2a}$$



Örnek: 1 den 5 e kadar olan sayıları yazdıran algoritmayı yazınız ve akış diyagramını çiziniz.

20

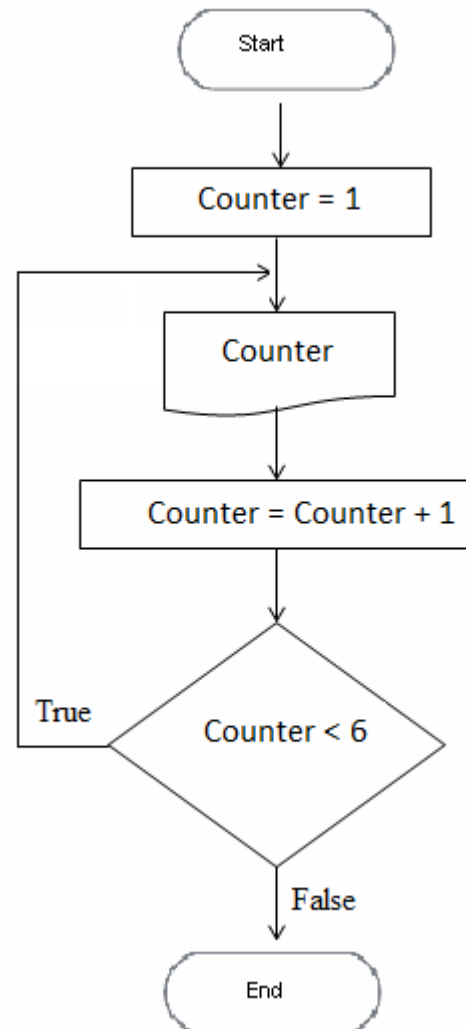
1. START
2. Counter (Sayac) = 1
3. Print the value of the counter
4. Counter = Counter + 1
5. If Counter < 6, GOTO 3
6. END.



Örnek: 1 den 5 e kadar olan sayıları yazdıran algoritmayı yazınız ve akış diyagramını çiziniz.

21

1. START
2. Counter (Sayac) = 1
3. Print the value of the counte
4. Counter = Counter + 1
5. If Counter < 6, GOTO 3
6. END.



Counter	Screen
1	1
2	2
3	3
4	4
5	5
6	

Flowcharts – Akış Şemaları

22

Örnek: 1'den 50'ye kadar olan sayıların toplamını bulan algoritmayı yazınız.

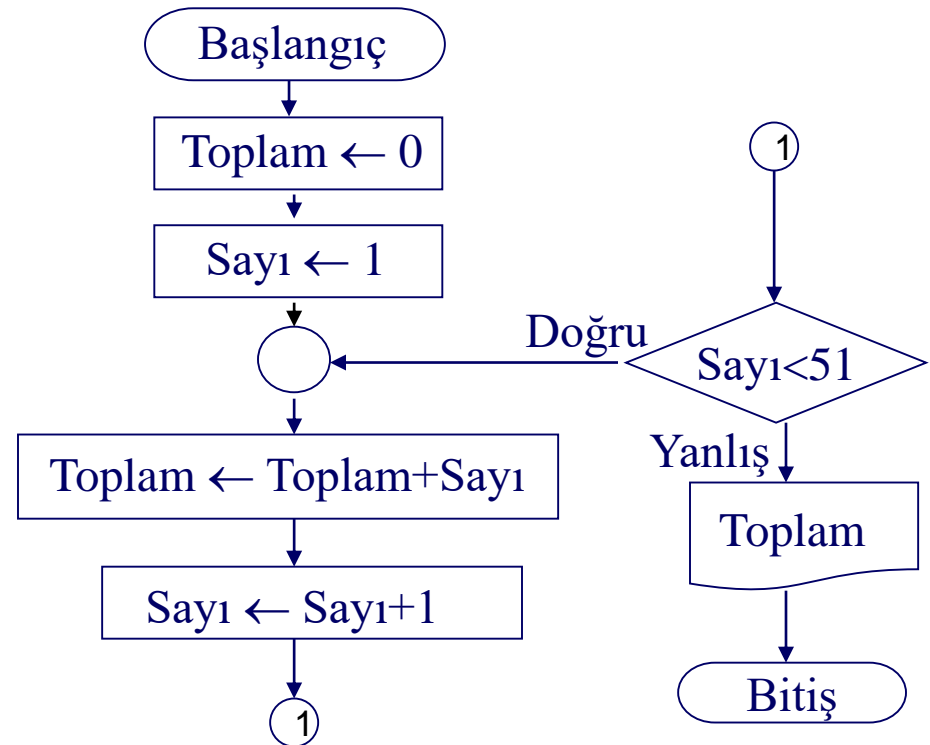
1. Başla.
2. Toplam $\leftarrow 0$
3. Sayı $\leftarrow 1$
4. Sayı 51'den küçük ise
 - a. Toplam \leftarrow Toplam + Sayı
 - b. Sayı \leftarrow Sayı + 1
 - c. 4. Adıma git
5. Değilse Toplam'ı yazdır
6. Bitir.

Flowcharts – Akış Şemaları

23

Örnek: 1'den 50'ye kadar olan sayıların toplamını bulan akış diyagramını çiziniz.

1. Başla.
2. $\text{Toplam} \leftarrow 0$
3. $\text{Sayı} \leftarrow 1$
4. Sayı 51'den küçük ise
 - a. $\text{Toplam} \leftarrow \text{Toplam} + \text{Sayı}$
 - b. $\text{Sayı} \leftarrow \text{Sayı} + 1$
 - c. 4. Adıma git
5. Değilse Toplam'ı yazdır
6. Bitir.

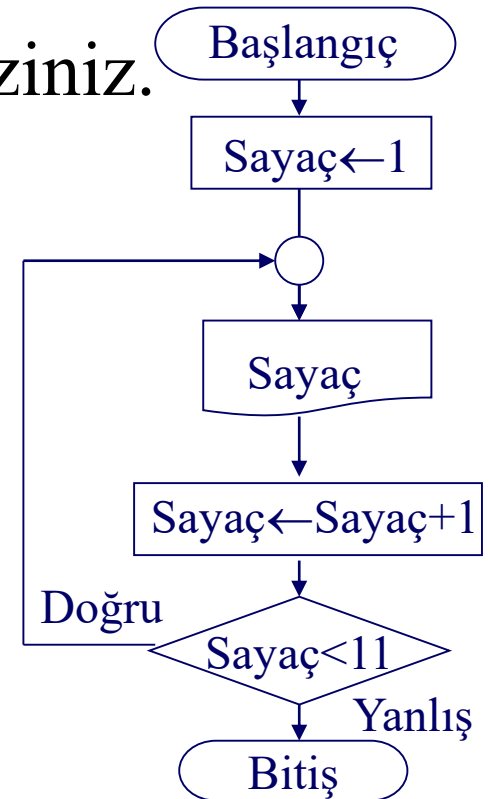


Flowcharts – Akış Şemaları (Sayaç)

24

Örnek: 1 den 10 a kadar sayıları yazan algoritmayı yazınız ve akış diyagramını çiziniz.

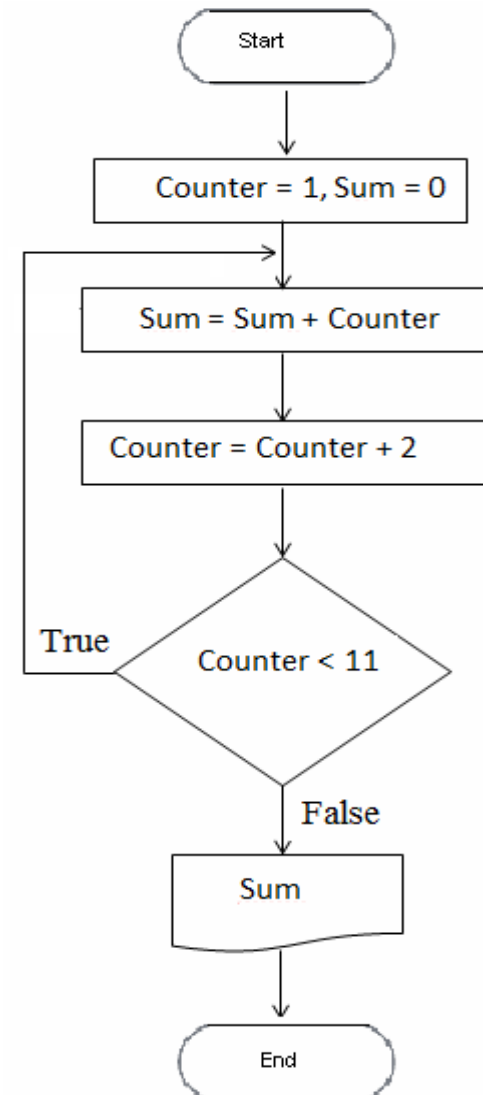
1. Başla.
2. $\text{Sayaç} \leftarrow 1$
3. Sayaç 'ı yaz.
4. $\text{Sayaç} \leftarrow \text{Sayaç} + 1$
5. Sayaç 11'den küçük ise 3. adıma git.
6. Bitir.



Örnek: 1 den 10 a kadar olan tek sayıların toplamını bulan algoritmayı yazınız ve akış diyagramını çiziniz.

25

1. START
2. Counter (Sayac) = 1
3. Sum = 0
4. Sum = Sum + Counter
5. Counter = Counter + 2
6. If (Counter < 11) GOTO 4
7. Print Sum
8. END



Örnek: 1 den 10 a kadar olan tek sayıların toplamını bulan algoritmayı yazınız ve akış diyagramını çiziniz.

26

1. START
2. Counter (Sayac) = 1
3. Sum = 0
4. Sum = Sum + Counter
5. Counter = Counter + 2
6. If (Counter < 11) GOTO 4
7. Print Sum
8. STOP

Counter	Sum
1	0
3	1
5	4
7	9
9	16
11	25

Flowcharts – Akış Şemaları

27

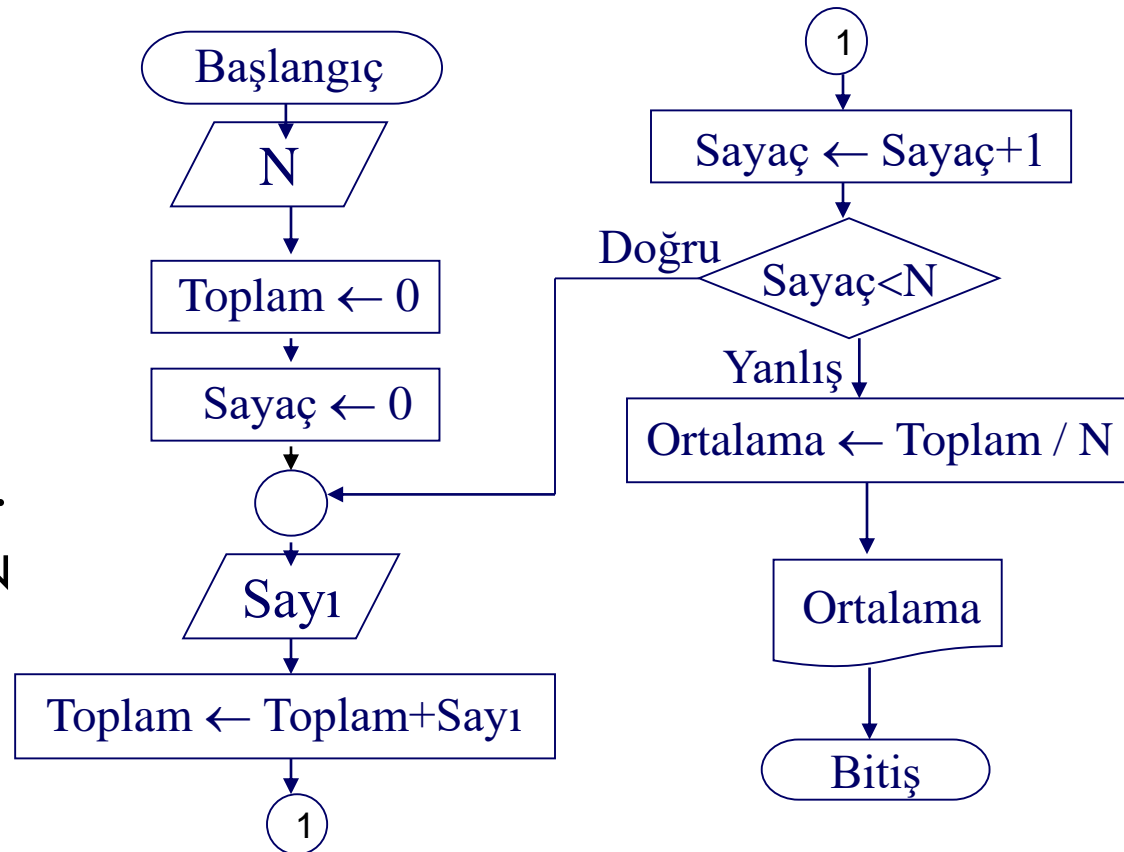
- **Örnek:** Kullanıcının gireceği N adet verinin ortalamasını bulup ekrana yazdıran algoritma ve akış şeması. Veri sayısını belirleyen N , kullanıcı tarafından girilecektir.

$$\frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N} = ?$$

Flowcharts – Akış Şemaları

28

1. Başla
2. Kullanıcı N değerini girsin.
3. $\text{Toplam} \leftarrow 0$
4. $\text{Sayaç} \leftarrow 0$
5. Kullanıcı bir Sayı girsin.
6. $\text{Toplam} \leftarrow \text{Toplam} + \text{Sayı}$
7. $\text{Sayaç} \leftarrow \text{Sayaç} + 1$
8. Eğer $\text{Sayaç} < N$ ise 5. adıma git.
9. Değilse $\text{Ortalama} \leftarrow \text{Toplam} / N$
10. Ortalamayı ekrana yazdır.
11. Bitir.



Uygulamalar

29

1. Üç sayının ortalamasını bulan algoritmayı yazınız ve akış şemasını çiziniz.
2. Girilen bir eşyanın fiyatının yüzde 20'si kadar indirim yaptırıp, yeni fiyatını ekrana yazdıran problemin algoritmasını yazınız ve akış diyagramını (flowchart) tasarlayınız.
3. Bir basketbol oyuncusunun kullandığı şut ve başarılı şut sayılarını alıp, başarılı şut oranını bulan problemin algoritmasını yazınız ve akış diyagramını (flowchart) tasarlayınız.

Uygulamalar

30

4. Bir öğrenciye ait vize ve final notlarının ortalamasını hesaplayan ve ortalamaya göre ekrana “Geçti” – “Kaldı” yazan problemin algoritmasını yazınız ve akış diyagramını tasarlayınız.

(Not: Ortalama hesaplanırken vizenin %40, finalin %60’ı alınacak, geçme notu 70’tir.)

Kontrol Yapıları (Control Structures)

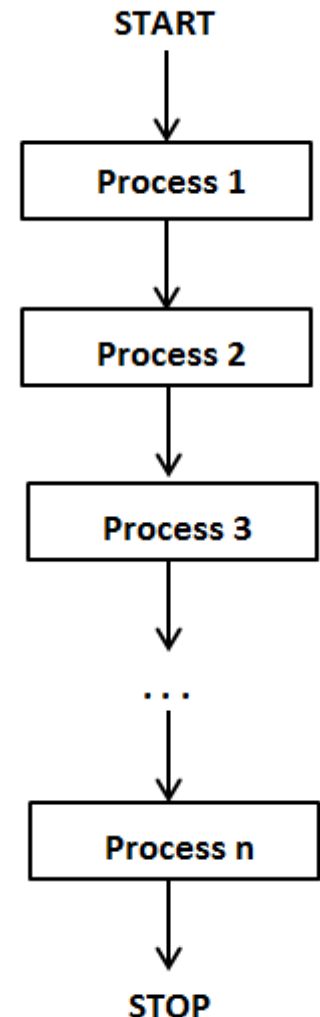
31

- Daha iyi algoritma tasarımının ve dolayısıyla programlamanın anahtarı, kontrol yapısını üç yapıyla sınırlamaktır.
- Sıralı yapı (The sequence structure)
- Karar Yapısı (Decision Structure)
- Tekrarlı Yapı (Repetition or Iteration Structure)

Sıralı Yapı (The Sequence Structure)

32

- Sıra yapısı, bir algoritmadaki adımların hiçbir koşul adımına gerek duyulmayacak şekilde yapılandırıldığı bir durumdur.

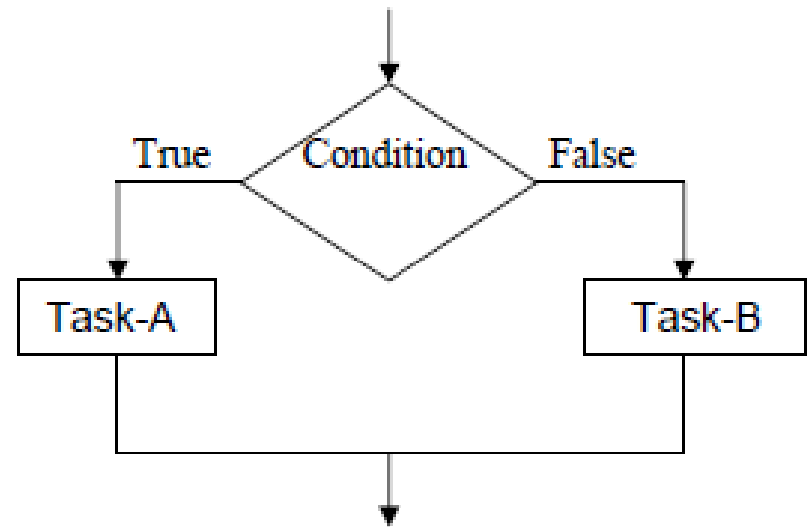


Karar Yapısı (Decision Structure)

33

- Karar yapısı, alıgoritmada verilen bir kořula göre karar vererek iki alternatif arasından seçim yapmak zorunda olduđu durumdur.

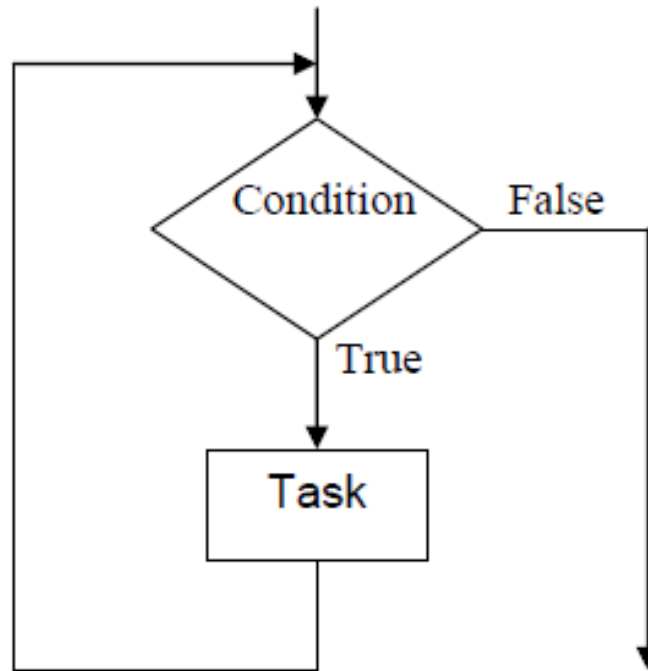
*If condition is true
Then do task A
else
Do Task-B*



Tekrarlı Yapı (Repetition Structure)

34

- Üçüncü bir yapı, belirli adımların tekrarlandığı tekrarlı yapıdır.



Operatörler

35

- **Aritmetik Operatörler**
- **İlişkisel Operatörler**
- **Mantıksal Operatörler**

Aritmetik Operatörler

36

Operation	
addition	$a + b$
subtraction	$a - b$
multiplication	$a * b$
division	a/b

$a+b-c+2abc-7$	$a+b-c+2*a*b*c-7$
$a+b^2-c^3$	$a+b^2-c^3$
$a - \frac{b}{c} + 2ac - \frac{2}{a+b}$	$a-b/c+2*a*c-2/(a+b)$
$\sqrt{a+b} - \frac{2ab}{b^2 - 4ac}$	$(a+b)^{(1/2)}-2*a*b/(b^2-4*a*c)$
$\frac{a^2+b^2}{2ab}$	$(a^2+b^2)/(2*a*b)$

İlişkisel Operatörler

37

==	Eşittir (equal to)
<>	Eşit değil (not equal to)
<	Küçüktür (Less than)
<=	Küçük veya eşit (less than or equal to)
>	Büyüktür (greater than)
>=	Büyüktür veya eşittir (Greater than or equal to)

Mantıksal Operatörler

38

- İlişkisel operatörler doğru/yanlış değerleri üretir.
- Üç mantıksal operatör şunlardır:
 - `&&` (AND)
 - `||` (OR)
 - `!` (NOT)

Mantıksal Operatörler (Logical Operators (**AND**))

39

- `&&` (ve) operatörü doğru/yanlış olmak üzere iki değer alır ve her iki koşul da doğruysa doğru (1) sonucu üretir.

<i>p</i>	<i>q</i>	<i>p && q</i>
F	F	F
F	T	F
T	F	F
T	T	T

<i>p</i>	<i>q</i>	<i>p && q</i>
0	0	0
0	1	0
1	0	0
1	1	1

Mantıksal Operatörler (Logical Operators (OR))

40

- \parallel (veya) operatörü doğru/yanlış değeri alır ve koşullardan biri doğruysa bir doğru (1) sonuç üretir.

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Mantıksal Operatörler (Logical Operators **(NOT)**)

41

- Mantıksal durumu tersine çevirmek için kullanılır. Bir koşul doğruysa, Mantıksal NOT operatörü yanlış yapar.

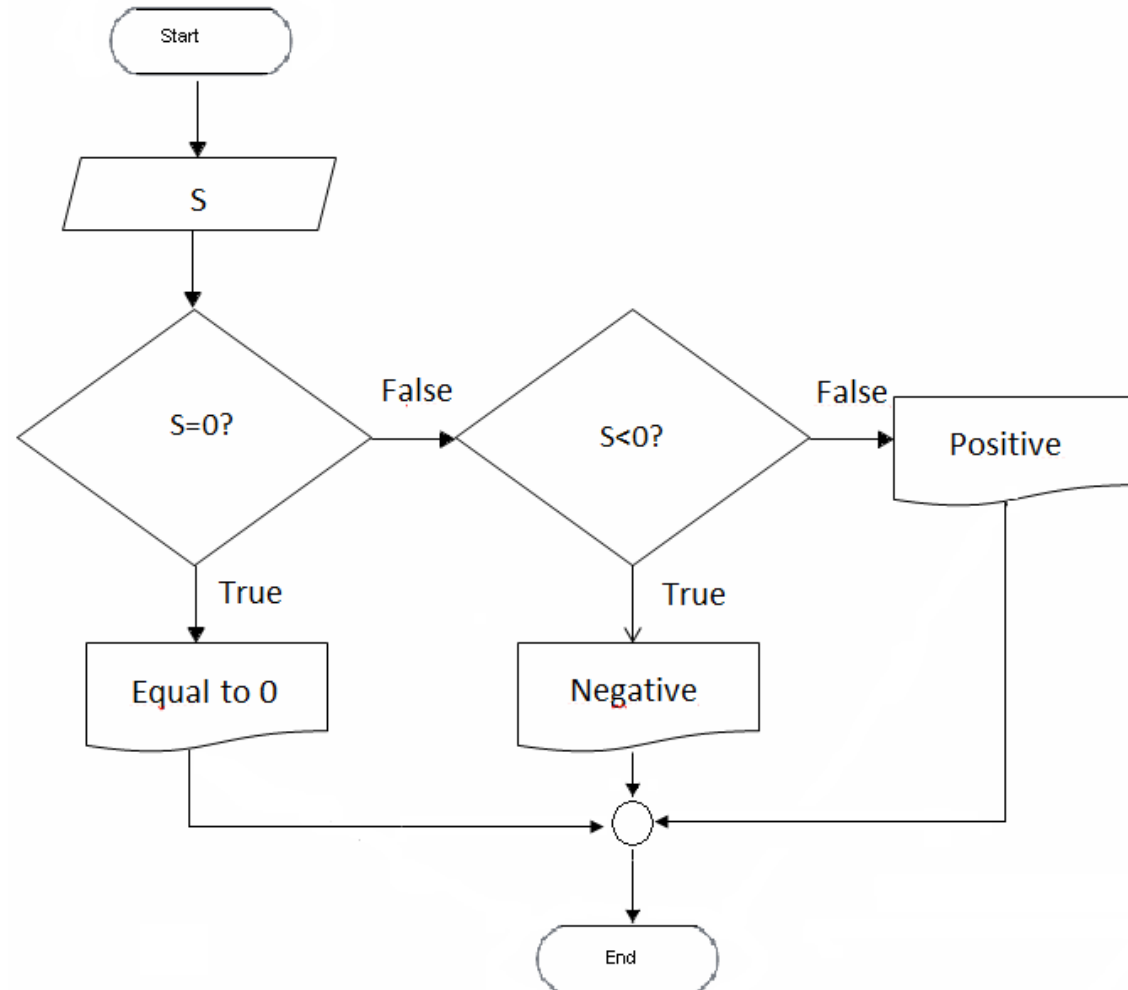
p	\bar{p}
T	F
F	T

p	\bar{p}
1	0
0	1

Örnek: Klavyeden girilen bir sayının pozitif, negatif veya sıfıra eşit olma durumunu bulup yazdıran programın algoritmasını yazınız ve akış şemasını çiziniz.

42

1. START
2. Read S
3. IF ($S = 0$)
 Print “Equal to 0”,
 Goto 6
4. IF ($S < 0$)
 Print “Negative”,
 Goto 6
5. ELSE ($S > 0$)
 Print “Positive”,
6. STOP.



Uygulamalar

43

1. 2^m değerini hesaplayan algoritmayı ve akış şemasını gösteriniz.
2. $N!$ Değerini hesaplayan algoritmayı ve akış şemasını gösteriniz.
3. Aylık cep telefonu tarifende 700 dakika konuşma ücretin vardır. 700 dakikayı bitirdikten sonra dakikası 35 kuruştan konuşabiliyorsun. Ay sonunda toplam kaç dakika konuşulduğu bilindiğine göre, faturana eklenecek tutarı hesaplayan bir algoritmayı hem kaba kod hem de akış şeması olarak yazınız.
4. Bir mağazada tüm ürünlerde yüzde 20 indirim vardır. Kasaya giden bir müşterinin tüm ürünlerinin indirim gerçekleştikten sonraki toplam ödeyeceği fiyatı veren algoritmanın hem kaba kodunu hem de akış şemasını yazınız.
5. Kullanıcının gireceği n adet sayıdan en büyük olanını bulduran algoritma ve akış şemasını yazınız.
6. Verilen bir sayının kaç basamaklı olduğunu bulan algoritma.
7. Bankamatik algoritması (İstenilen miktar için her bir para tipinden kaç banknot verilmeli)

Sorular???

44

