








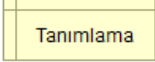
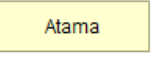
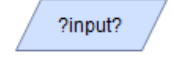
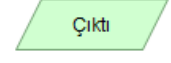

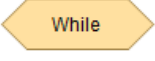
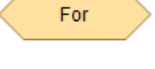
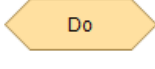
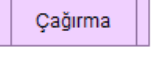
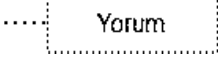

FLOWGORİTHM DERS NOTLARI:

Flowgorithm: Akış diyagramı şekilleri kullanarak basit kodlar yazma programıdır.

1. Program Arayüzü:

- a.  **Aç** : Önceden kayıtlı bir dosyayı yükler.
- b.  **Kaydet** : Mevcut çalışmayı kayıt eder.
- c.  **Çalıştır** : Akış diyagramını çalıştırır.
- d.  **Adım** : Kodları adım adım sırayla çalıştırır. (BP)
- e.  **Fonksiyon Yöneticisi** : Ana program veya alt programlar arası geçişi sağlar.
- f.   **Büyüteç** : Sayfayı büyütür ya da küçültür.

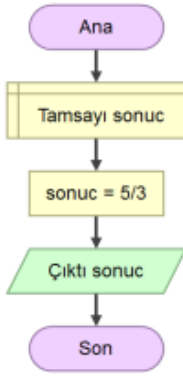
2. Akış Diyagramı Şekilleri :

- a.  : Kullanılacak değişkenleri sisteme söyler.
- b.  : Tanımlanmış değişkene değer atar.
- c.  : Klavyeden bilgi girişi için kullanılır.
- d.  : Ekranı bilgi yazdırmak için kullanılır.
- e.  : Karar verme işlemleri için kullanılır.
- f.  : Tekrar sayısı belli olmayan DÖNGÜ için kullanılır.
- g.  : Tekrar sayısı belli olan DÖNGÜ için kullanılır.
- h.  : En az 1 Kere çalışması gereken DÖNGÜ için
- i.  : Tanımlı altprogramı çağırma için kullanılır.
- j.  : Kodların çalışmasını ETKİLEMEYEN, programcı için AÇIKLAYICI BİLGİLER yazmak için kullanılır.
- k.  (KIRILMA NOKTASI / BREAK POINT): **Hata ayıklama** işlemlerinde çalışan programın bu noktaya geldiğinde DURAKLAMASINI sağlar. Programın çalışması istenildiğinde Normal olarak (▶) ya da (ADIM ADIM ▶) devam ettirilir.

3. Veri Tipleri:

a. **Integer** : Tam sayı verileri saklamak için kullanılır.

-10 3 100 10000000000 (taşma)



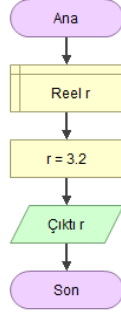
(Çıktı neden 1 olur?)

b. Real (Reel): Ondalıklı sayıları saklamak için kullanılır.

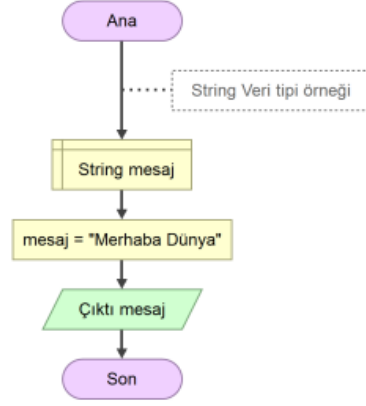
~~3,2 Virgöl OLMAZ~~

3.2 Doğru

$5/5 = 1.0$ (bölme işlemi sonucu her zaman reel)



c. **String** : Herhangi text veriyi saklamak için kullanılır.
“Hoşgeldiniz” “1. Sayıyı giriniz:”



d. **Boolean** : Sadece True/False ya da 0/1 değerini saklar.

(İF ile birlikte iyi gider 😊)

sonuc = 5>3 (sonuc TRUE olur)

önce eşitliğin sağ tarafı

sonuc = 1>2 (sonuc FALSE olur)

4. Operatörler:

a. Matematiksel: İşlem Önceliğine Dikkat

- i. Toplama + : 3+2
- ii. Çıkarma - : 9-7
- iii. Çarpma * : 6*3
- iv. Bölme / : 9/3
- v. Mod % : 10%3
- vi. Üs Alma ^ : 4^3

| Level | Name | Operators | Notes |
|-------|-------------|-------------------------|---|
| 8 | Unary | - ! not | In Visual Basic, "not" precedence level is far lower - above "and", but below all relational operators. |
| 7 | Exponent | ^ | The exponent operator does not exist in C# or Java. |
| 6 | Multiply | * / % mod | Division will always be high-precision (floating point) |
| 5 | Addition | + - | In Flowgorithm, "+" will only work with numbers. |
| 4 | Concatenate | & | C# and Java use the ambiguous "+" operator for addition and concatenation. |
| 3 | Relational | > >= < <= == = != <> | |
| 2 | Logical And | and && | |
| 1 | Logical Or | or | |

b. Mantıksal Operatör

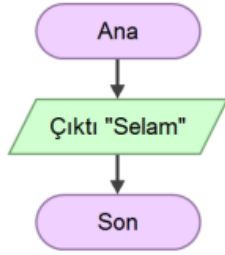
- i. Eşit Mi == 3==3 > True
- ii. Eşit Değil mi != <> 4!=5 True
- iii. Büyük mü >
- iv. Büyük Eşit mi >=
- v. Küçük mü <
- vi. Küçük Eşit mi <=
- vii. Veya || OR
- viii. Ve && AND
- ix. Değil ! NOT

c. String Birleştirme (Concatenation) &

"Merhaba" & "Dünya"

Merhaba String ifadesi ile Dünya String ifadesini birleştirir. MerhabaDünya String ifadesini oluşturur.

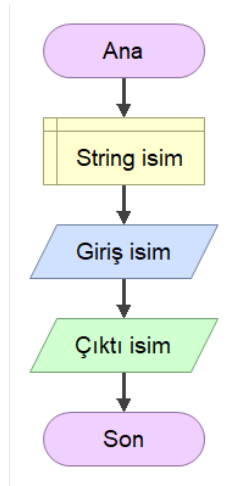
Örnek-1: Ekranaya Selam Yazan program.



Örnek-2: Kullanıcı klavyeden adını girecek.

Program da "Merhaba X" olarak ekranaya çıktı üretecek.

(x kişinin girdiği adını temsil ediyor. Herkes farklı isim girebilir.)



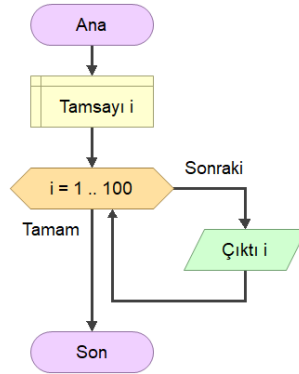
Açıklama [WK1]: Sol taraftaki algoritmayı, Merhaba XXXXXX şeklinde ekranaya çıktı verecek şekilde düzeltiniz.

DÖNGÜ: Tekrar eden yapılardır.

| Tekrar Sayısı Belli | Tekrar Sayısı Belli Değil |
|---------------------|---------------------------|
| FOR | WHİLE |

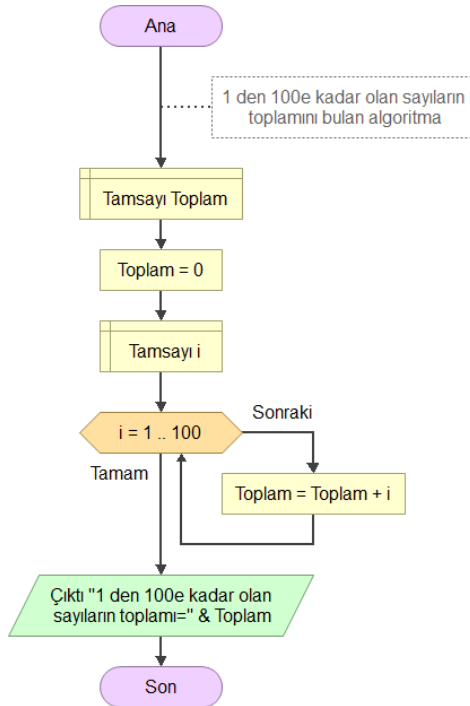
Örnek-3: 1 den 100 e kadar (100 dahil) tam sayıları ekrana yazdırın. Tekrar sayısı belli bir döngü.

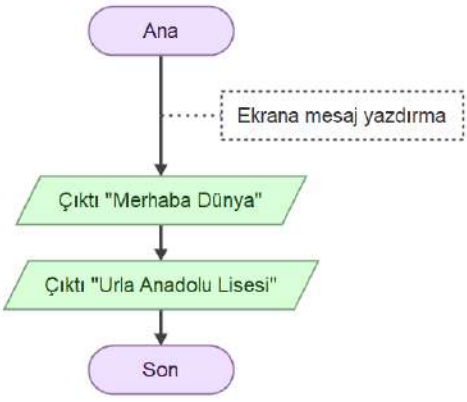
Açıklama [WK2]: Başlangıcı=1
Bitiş=100
Adım=1
Tekrar sayısı belirli olduğu için
FOR döngüsü



Örnek-4: 1 den 100 e kadar olan sayıların **TOPLAMI.** (Tekrar sayısı belli)

Açıklama [WK3]: Toplam = Toplam + i
teknîği ile, her i değeri, Toplam değişkenine
ilave edilir.

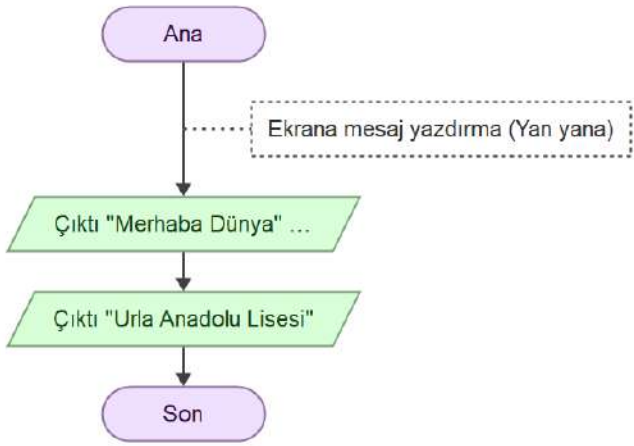




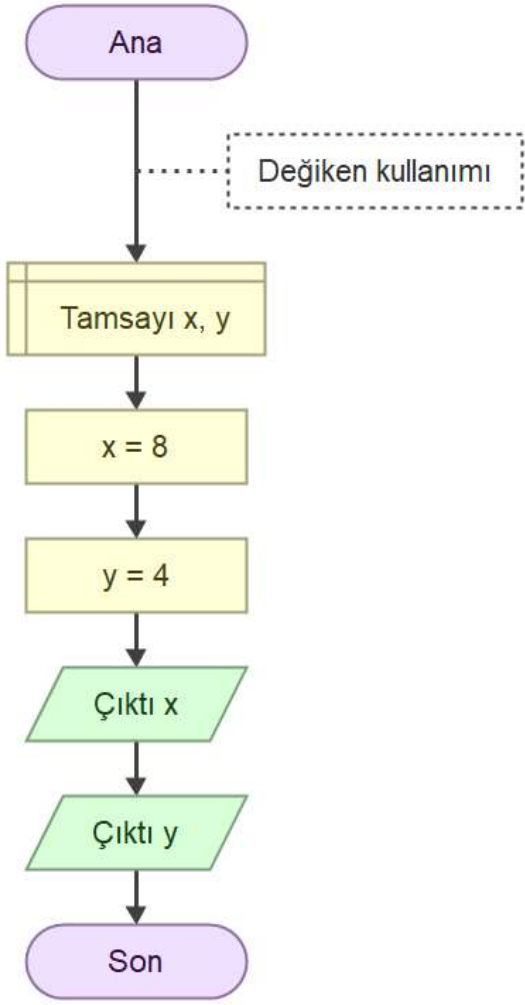
Merhaba Dünya

Urla Anadolu Lisesi



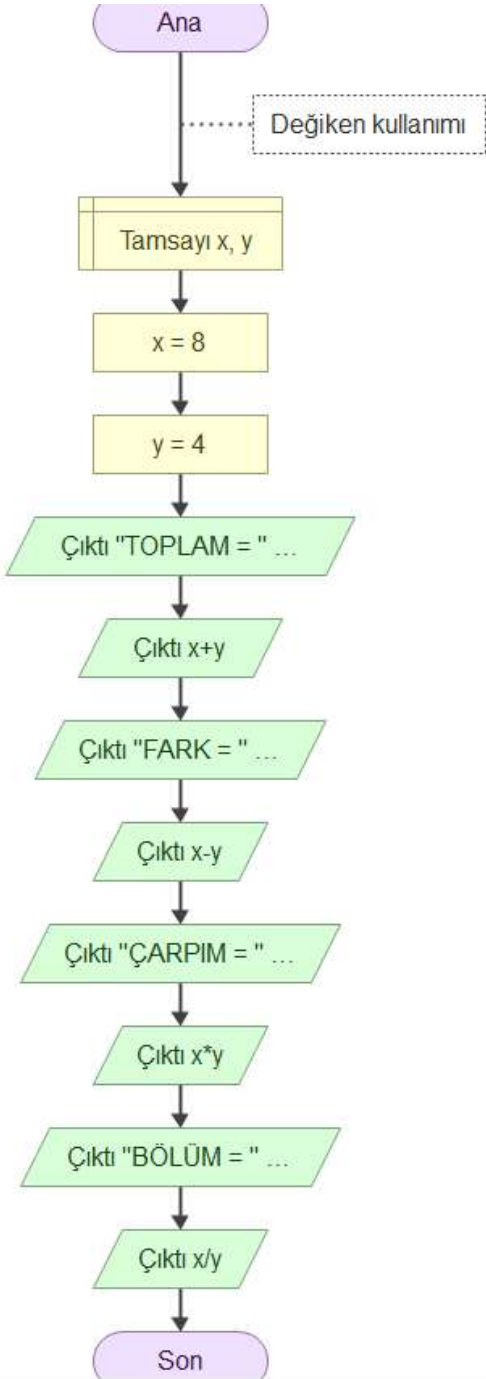


Merhaba DünyaUrla Anadolu Lisesi



8

4



TOPLAM = 12

FARK = 4

ÇARPIM = 32

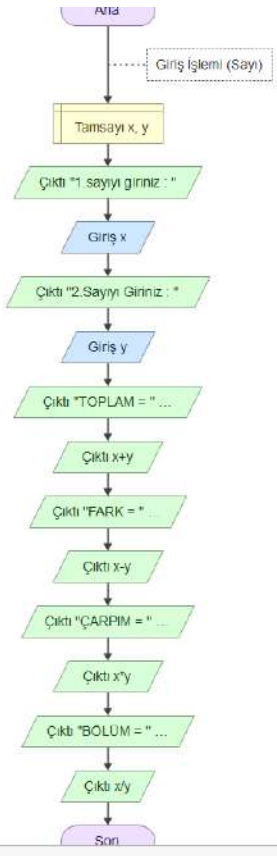
BÖLÜM = 2



A vertical terminal window with a grey border and a scroll bar on the left. The output is as follows:

```
İsminizi giriniz :  
Merhaba Ahmet
```

Below the terminal window is a white input field with a small upward-pointing arrow on the right side. To the right of the terminal window, there is a blue speech bubble containing the text "Ahmet".



1.sayıyı giriniz :

10

2.Sayıyı Giriniz :

4

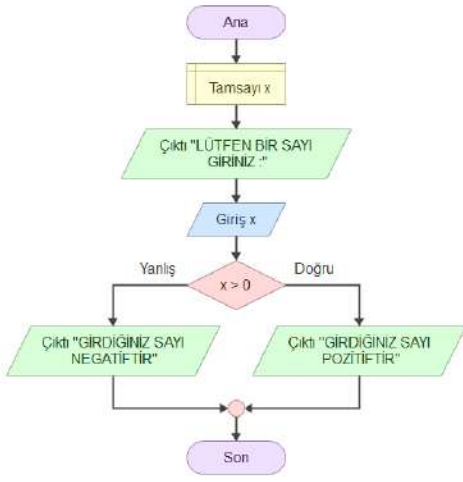
TOPLAM = 14

FARK = 6

ÇARPIM = 40

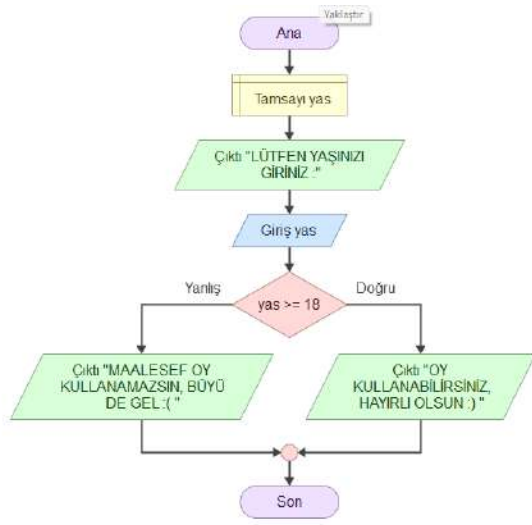
BÖLÜM = 2.5





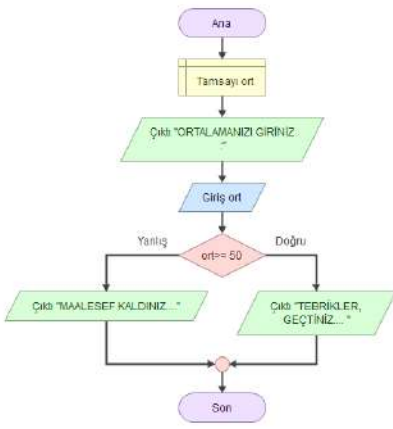
LÜTFEN BİR SAYI GİRİNİZ :

GİRDİĞİNİZ SAYI POZİTİFTİR



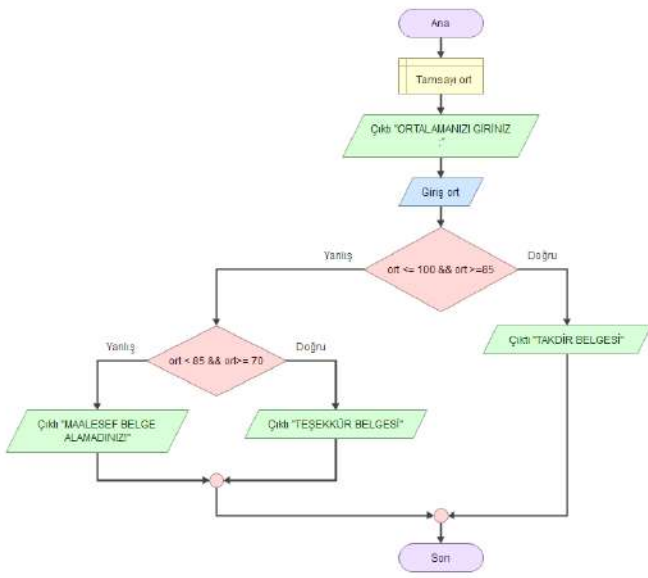
LÜTFEN YAŞINIZI GİRİNİZ :

MAALESEF OY KULLANAMAZSIN, BÜYÜ DE GEL (



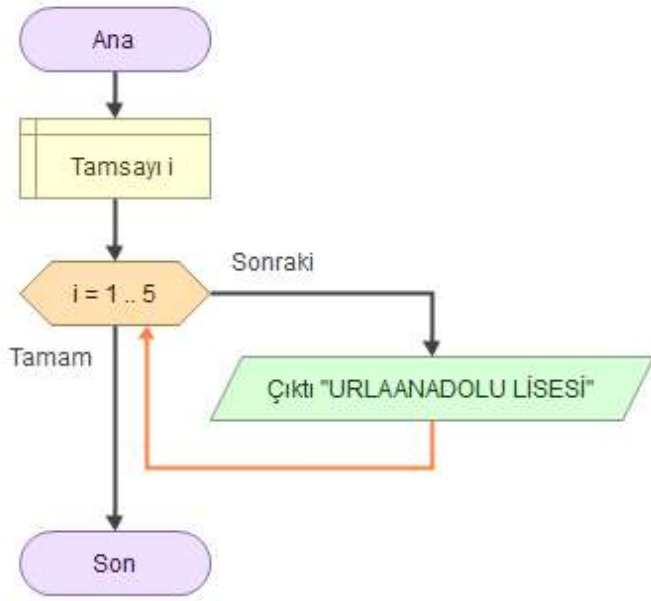
ORTALAMANIZI GIRINIZ :

TEBRİKLER, GEÇTİNİZ...



ORTALAMANIZI GIRINIZ .

TEŞEKKÜR BELGESİ



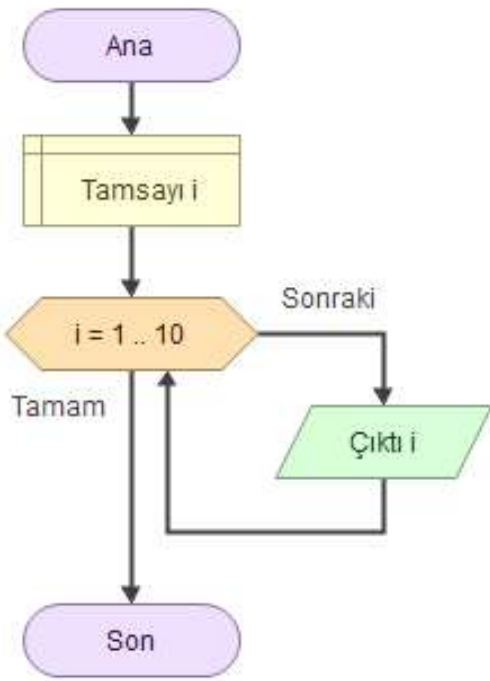
URLAANADOLU LİSESİ

URLAANADOLU LİSESİ

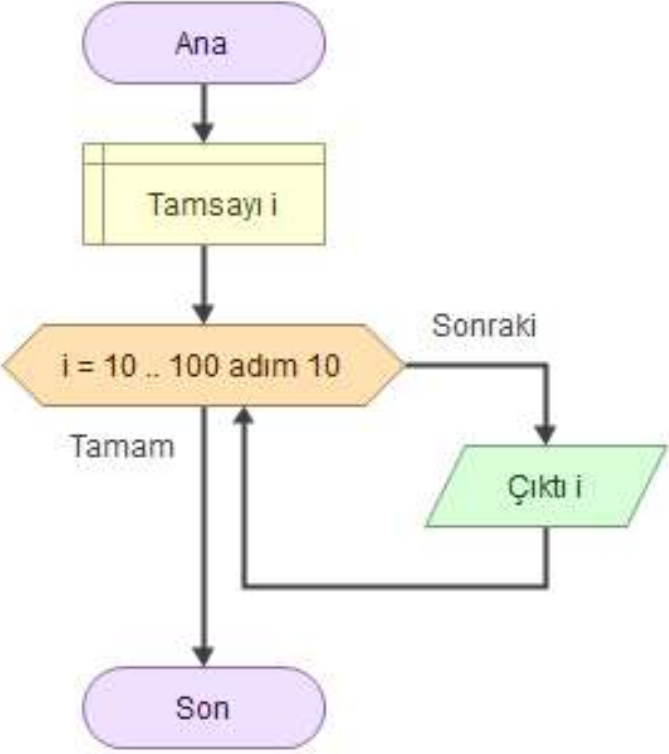
URLAANADOLU LİSESİ

URLAANADOLU LİSESİ

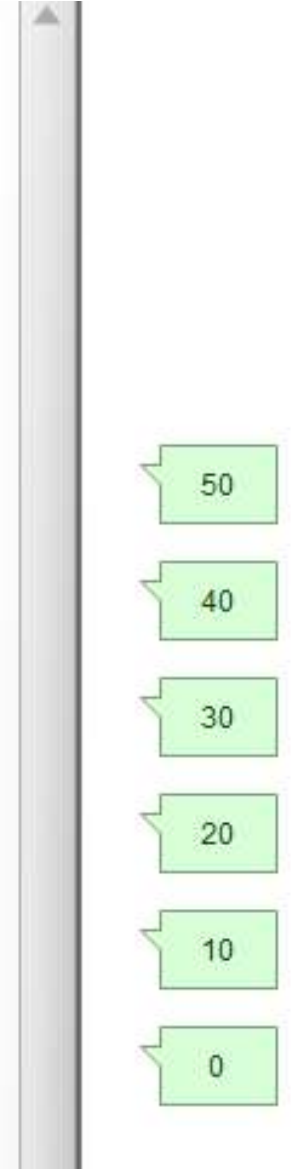
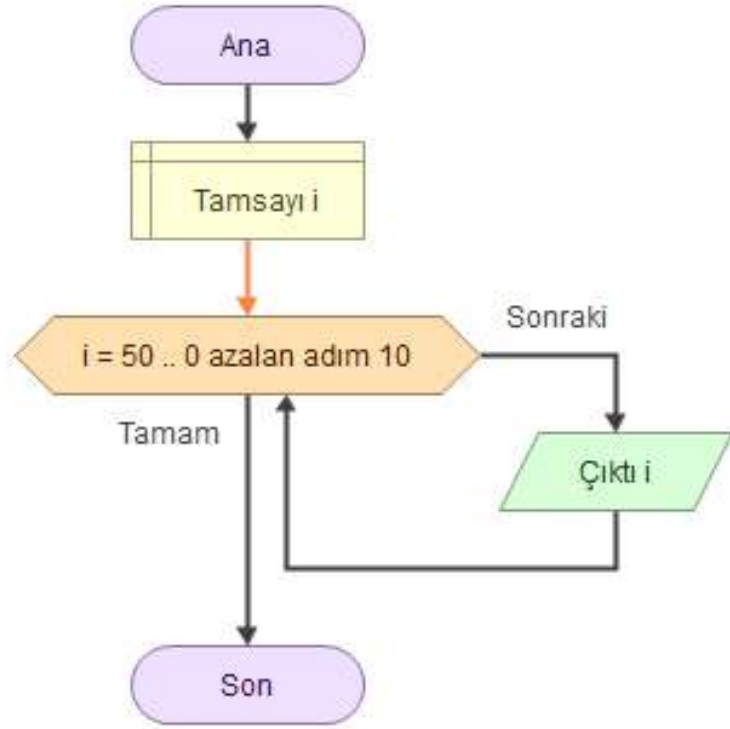
URLAANADOLU LİSESİ

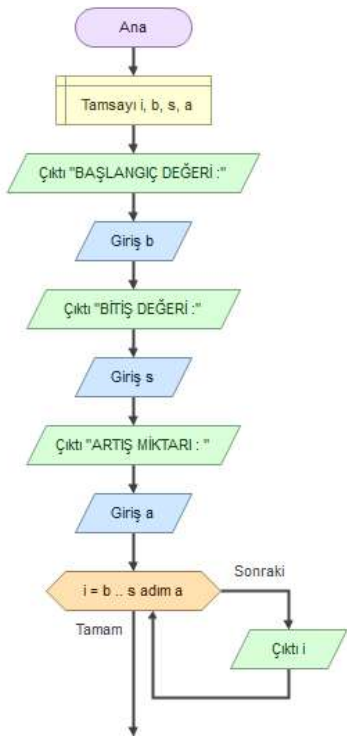


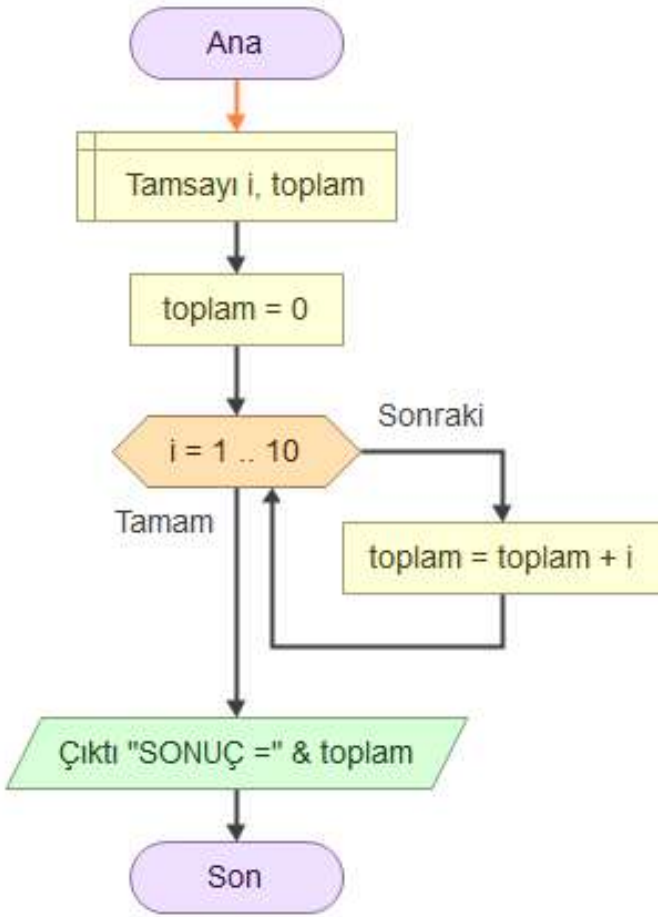
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10



- 10
- 20
- 30
- 40
- 50
- 60
- 70
- 80
- 90
- 100







SONUÇ =55