



**HOŞGELDİNİZ! :)**

**REÇELLİ EKMEK YER MİSİNİZ?**

UFACIK

# BİR HATI<sup>Ş</sup> LATMA

- Sınıfa **ağızı açık kaplarda** içeceklerle LÜTFEN gelmeyiniz (Çay, kahve...)
  - Tetikli Termos, Kapaklı Su şişesi...
- Ayrılırken;
  - + Masa üstlerinde LÜTFEN ÇÖP BIRAKMAYINIZ,
  - + Bilgisayarlarınızı LÜTFEN KAPATINIZ,
  - + Sandalyelerinizi LÜTFEN DÜZENLİ bırakınız,
- Ders başlangıcı : Max. 5 Dk. Tolerans...  
**19:05 Ders başlar!**

## Windows

An exception 06 has occurred at 0028:C11B3ADC in VxD DiskTSD(03) + 00001660. This was called from 0028:C11B40C8 in VxD voltrack(04) + 00000000. It may be possible to continue normally.

- \* Press any key to attempt to continue.
- \* Press CTRL+ALT+RESET to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue

# Aşırı Hızlı Tekrar

- Yazılım Geliştirme, KISS Rule, Öğrenme Nedir?
- Ekip Çalışması...
- Bilgisayarın Ana Bileşenleri
- Programlama Dilleri
- Yorumlayıcı (Interpreter) / Derleyici (Compiler)
- Algoritma Nedir ve ifade Şekilleri

# Görev - 2:

## Html bir programlama dili midir?

## Neden?

# Python Hakkında :)

- Yorumlanan (Interpreted)
- Nesne Yönelimli
- Çok Yüksek Seviyeli
- Çok Amaçlı bir dildir
  - Yazılım Geliştirme
  - Yazılım Test Otomasyonu
  - Veri Bilimi (Varyans, Ort, Mod, Median vs.)
  - Makine Öğrenmesi
  - Web Geliştirme (BackEnd)

# Python Hakkında :)

- Yorumlanan (Interpreted)
- Nesne Yönelimli
- Çok Yüksek Seviyeli
- ~~Çok Amaçlı~~ Diğerlerine göre kullanması çok daha KOLAY bir dildir
  - Yazılım Geliştirme
  - Yazılım Test Otomasyonu
  - Veri Bilimi
  - Makine Öğrenmesi
  - Web Geliştirme (BackEnd)

# Python Hakkında :)

- Guido van Rossum
- 1989 da Noel Tatilinde uğraşacak bir hobi arıyor
- Ofisi kapalı olduğu için evinde bilgisayar başında
- “Monty Python” komedi grubunu çok sevdiği için
- Alt yapısı C dir.

## Nerde Tercih Edilmez?

- Performans gerektiren kod yazmanız gerektiğinde
- Mobil Uygulamalar



# Alternatifleri

- Perl
- Ruby
- Cython
- Jython
- Rpython
- PyPy

# Görev – 3 :

**Bir dilim ekmeğin üstüne:  
Reçel sürüp yeme algoritmasını  
yazınız...**

# Yazılımcı Olmak

Annem dedi ki:

“ Tatlım markete git ve 1 şişe süt al. Eğer yumurta varsa 6 tane al “

6 şişe süt ile eve döndüm.

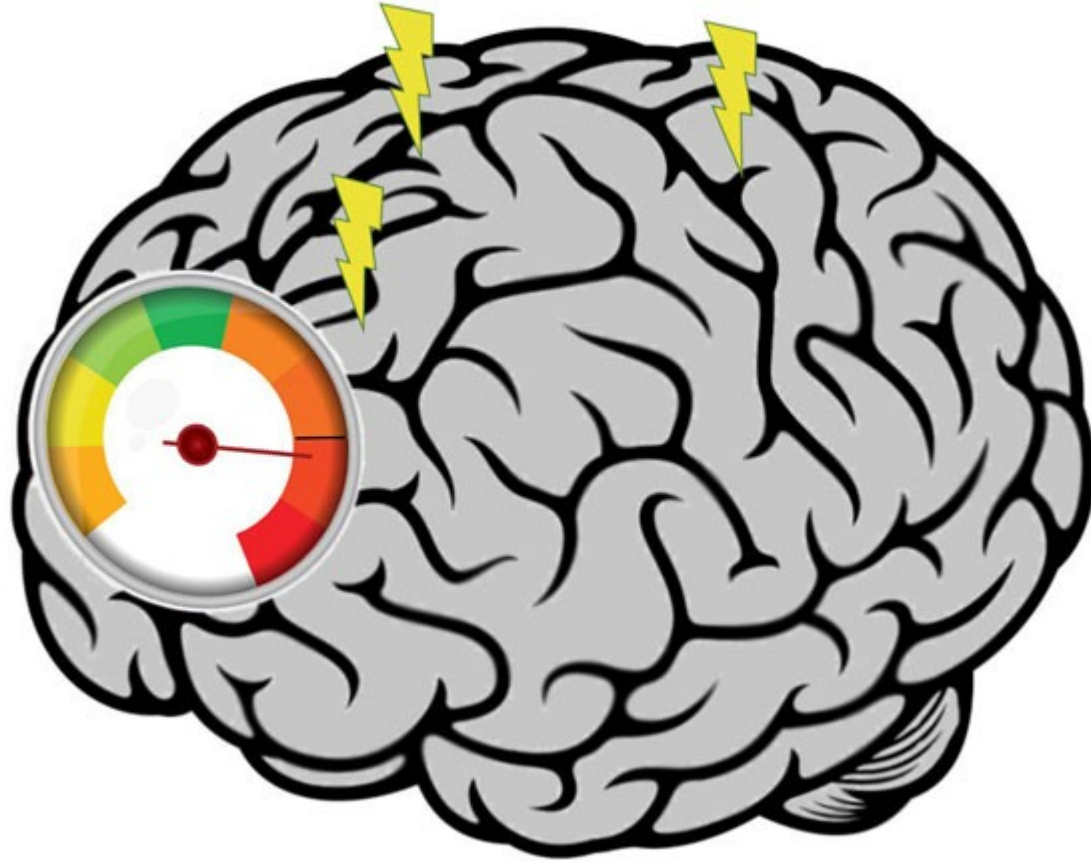
O dedi ki: “ Neden 6 şişe süt aldın? “

Cevap verdim: “ Çünkü yumurta vardı “

**YEDİK, İÇTİK  
GÜLDÜK, EĞLENDİK...**

**ARTIK HAZIR MIYIZ? >:)**

# ÖNCE BİRAZ BELLEK ÜZERİNE...



# BAZI BAZI

## DEĞİŞKENLER :

– Bir sayıyı, bir ismi program içinde kullanmak üzere hafızada saklamak amacıyla kullanılır...

`{değişken_adı} = {değer}`

```
ogrenciAdi = "Roni"
```

```
say1 = 5
```

```
say12 = 12
```

```
toplam = 12 + 5
```

```
toplam = say1 + say12
```

```
Ort = (12 + 5) / 2 → (say1+say12) / 2
```

# BAZI OPERATÖRLER

**+, -**

**\* ÇARPMA, \*\* ÜS ALMA**

**/ BÖLME**

**== EŞİT Mİ?**

**< KÜÇÜKTÜR**

**> BÜYÜKTÜR**

**<= KÜÇÜK EŞİT**

**>= BÜYÜK EŞİT**

**% MOD (BÖLÜMDEN KALAN DEĞERİ VERİR)**

# BAZI KAVRAMLAR

## KOŞUL İFADESİ

– 1 ile 10 arasında sayı giriniz...

(sayı > 0 ve sayı < 11)

## DÖNGÜ İFADESİ

– 1 den 10 a kadar sayıları toplayınız...

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

– Peki ya 1000 derse? :)



# Algoritma Örnekleri

## ÖRNEK 1 :

Kullanıcı tarafından girilecek 2 sayının toplamını ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

# Algoritma Örnekleri

## ÖRNEK 1 :

Kullanıcı tarafından girilecek 2 sayının toplamını ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

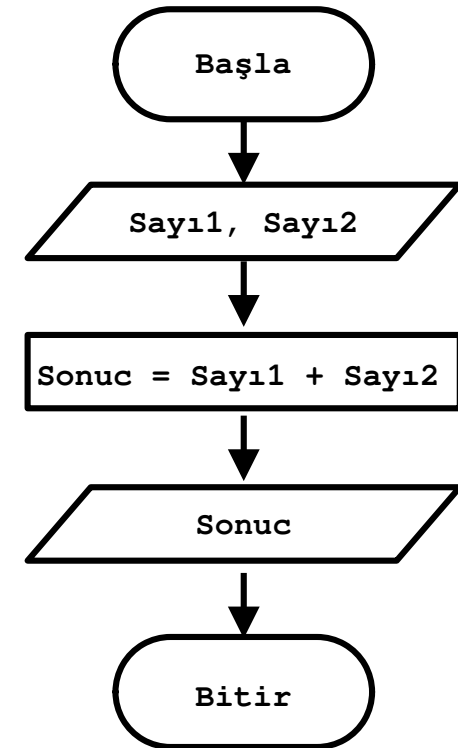
Başla:

Adım 1. Giriş: sayı1, sayı2

Adım 2. İşlem:  $\text{sonuc} = \text{sayı1} + \text{sayı2}$

Adım 3. Yaz : Sonuc

Bitir:



# Algoritma Örnekleri

## ÖRNEK 2 :

Kullanıcı tarafından girilecek 2 sayıdan büyük olanı ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

# Algoritma Örnekleri

## ÖRNEK 2 :

Kullanıcı tarafından girilecek 2 sayıdan büyük olanı ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

Başla:

Adım 1. Giriş: `sayi1, sayi2`

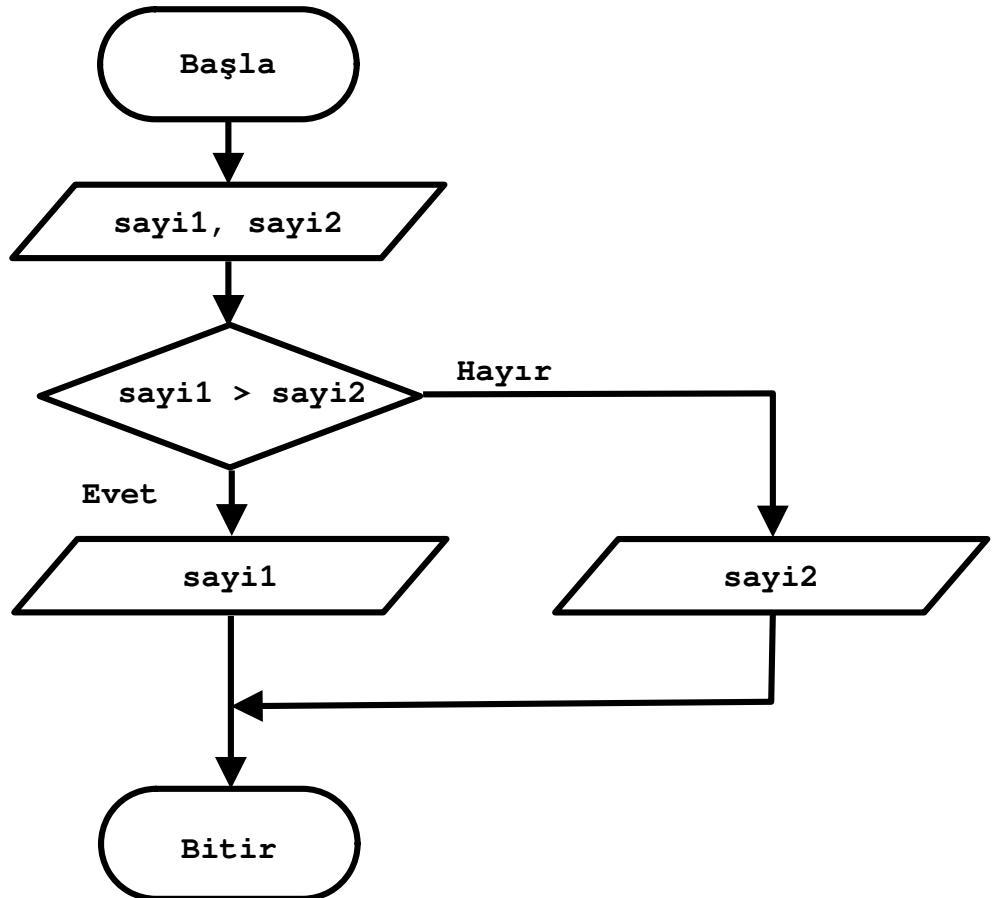
Adım 2. EĞER `sayi1 > sayi2`

Yaz : `sayi1`

DEĞİLSE

Yaz : `sayi2`

Bitir:



# Algoritma Örnekleri

## ÖRNEK 3 :

Kullanıcı tarafından girilecek olan ad değişkenini, 10 kere ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

# Algoritma Örnekleri

## ÖRNEK 3 :

Kullanıcı tarafından girilecek olan ad değişkenini, 10 kere ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

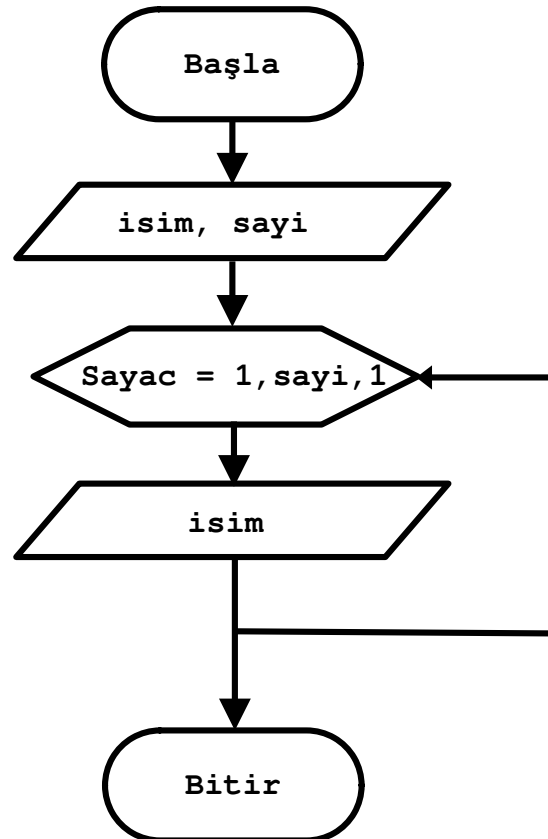
Başla:

Adım 1. Giriş: isim, sayi

Adım 2. Tekrarla : 1,sayi,1

Yaz : isim

Bitir:



# Algoritma Örnekleri

## ÖRNEK 3 :

Kullanıcı tarafından girilecek olan ad değişkenini, 10 kere ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.

Başla:

Adım 1. Giriş: isim, sayi

Adım 2. sayac = 1

Adım 3. Eğer sayac < sayi

Yaz : isim

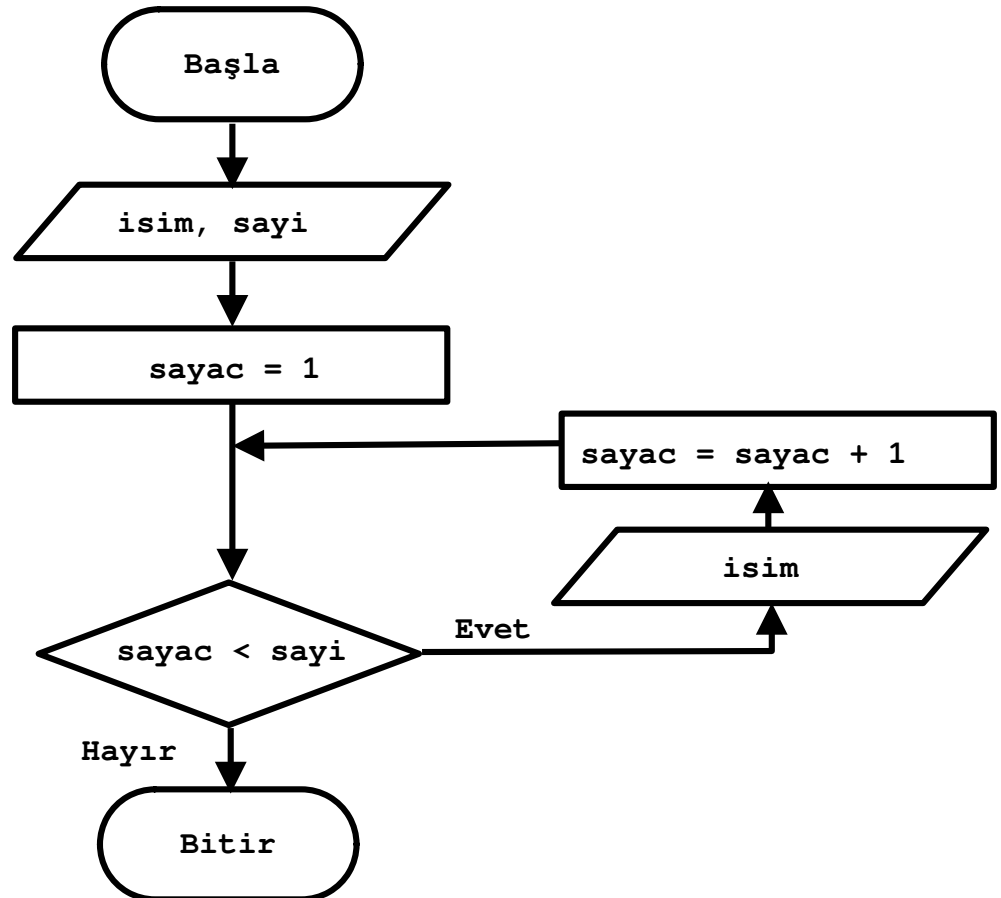
sayac = sayac + 1

Git : Adım 3.

DEĞİLSE

Çık:

Bitir:



# Sıra Sizde... >:)

## PROBLEM 1 :

Kullanıcı tarafından girilecek olan sayının, “Sıfıra Eşit”, “Sıfırdan Büyük” veya “Sıfırdan Küçük” olup olmadığını ekrana yazdıran algoritma ve akış diyagramını tasarlayınız.



# Sıra Sizde... >:)

ÇÖZÜM 1 :

Başla:

Adım 1. Giriş: sayi

Adım 2. Eğer  $\text{sayi} < 0$

Yaz : "Sıfırdan Küçüktür"

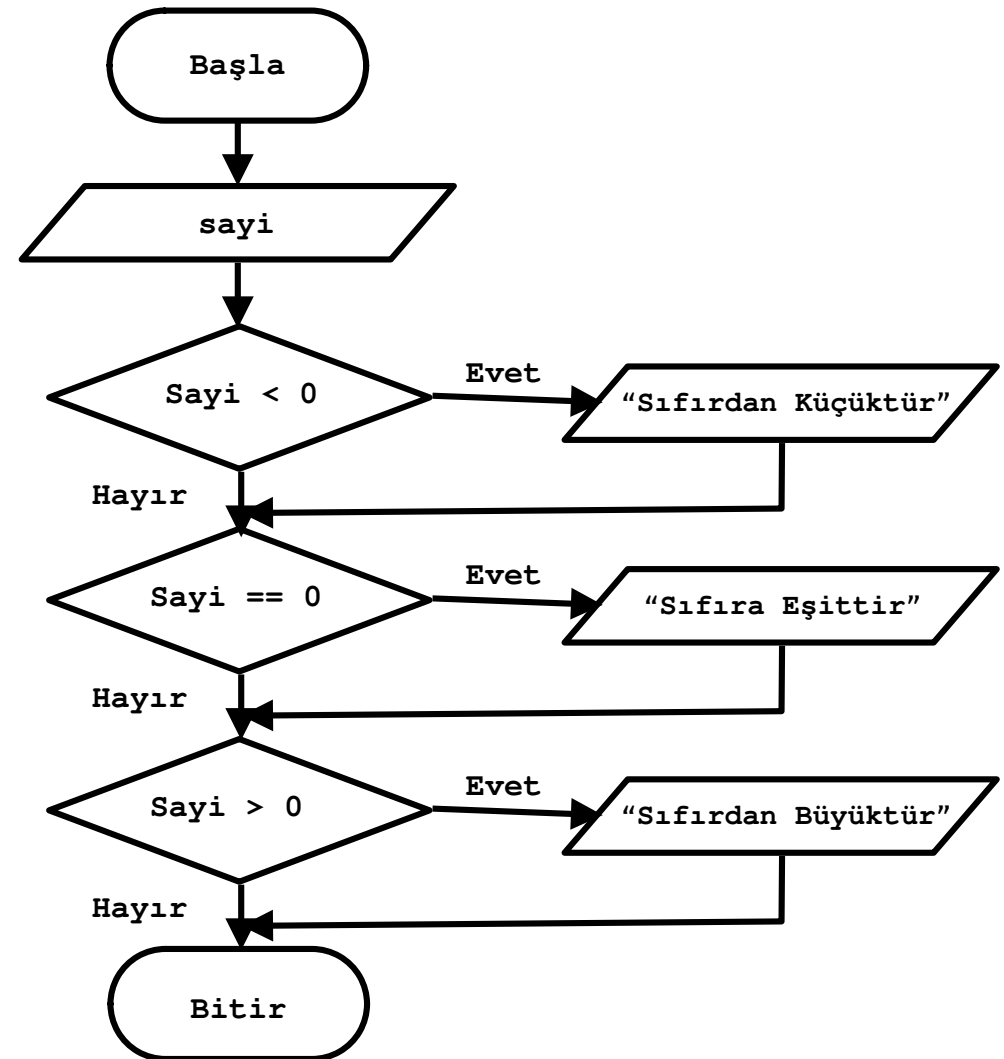
Adım 3. Eğer  $\text{sayi} == 0$

Yaz : "Sıfıra Eşittir"

Adım 4. Eğer  $\text{sayi} > 0$

Yaz : "Sıfırdan Büyüktür"

Bitir:



# Sıra Sizde... >:)

## PROBLEM 2 :

1 den, Kullanıcı tarafından girilecek olan sayıya kadar olan sayıların toplamını ekrana yazdıran algoritma ve akış diyagramını DÖNGÜ KULLANARAK tasarlayınız.

# Sıra Sizde... >:)

ÇÖZÜM 2 :

Başla:

Adım 1. Giriş: *sayi*

Adım 2. *sayac* = 1

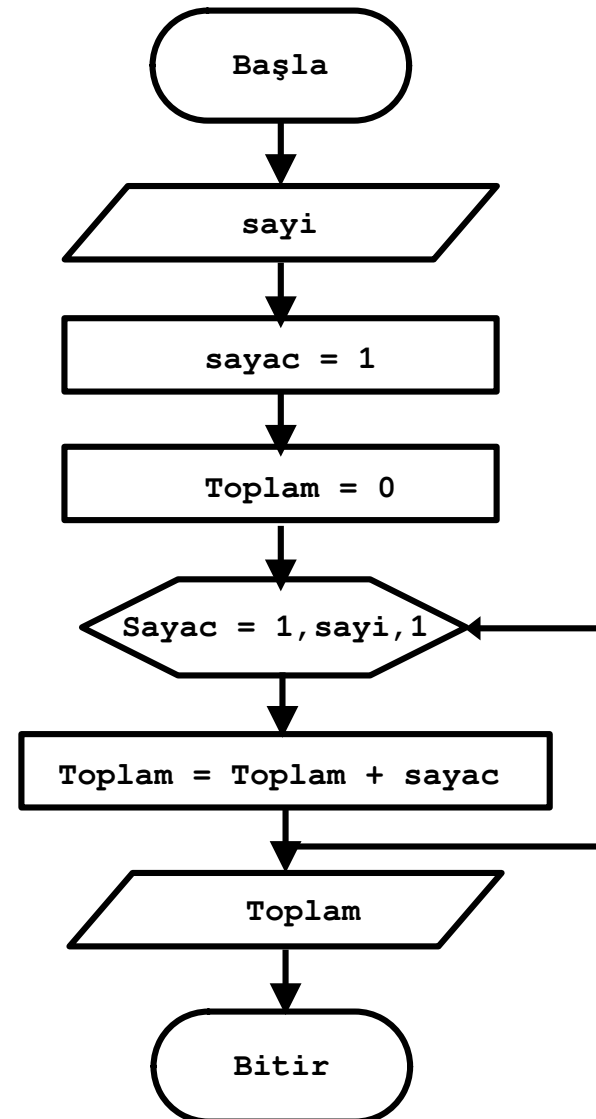
Adım 3. *Toplam* = 0

Adım 3. Tekrarla:1, *sayi*, 1

$\text{Toplam} = \text{Toplam} + \text{sayac}$

Adım 4. Yaz: *Toplam*

Bitir:



# Sıra Sizde... >:)

## PROBLEM 3 :

Bir elektrik devresinin güç kaynağının oluşturduğu, yüklü elektronları (akım) iletken bir döngü boyunca iten ve ışık oluşturma gibi iş gerçekleştirmelerini sağlayan basınca gerilim denir. Belirli bir süre boyunca bir devredeki noktadan geçen elektronların miktarına ise akım denir. Bir elektrik devresinde akım akışına karşı oluşan etkinin ölçümüne ise Direnç denir. Bu bilgilere paralel olarak:

Bir devre elemanı üzerindeki gerilim (Volt-V), üzerinden geçen akım (Amper-I) ve devre elemanının direncinin (Resistance - R) çarpımına eşittir ve  $V=I \times R$  ile gösterilir. Kullanıcı tarafından girilecek olan akım ve direnc değerleri ile devre elemanı üzerinden geçen gerilim değerini hesaplayan algoritma ve akış diyagramını tasarlayınız.

# Sıra Sizde... >:)

ÇÖZÜM 3 :

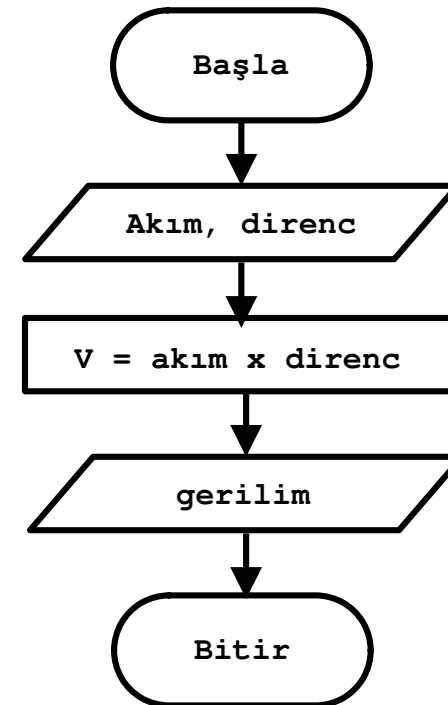
Başla:

Adım 1. Giriş: akım, direnc

Adım 2.  $\text{gerilim} = \text{akım} \times \text{direnc}$

Adım 3. Yaz: v

Bitir:



# Görev - 4:

1 den, Kullanıcı tarafından girilecek olan sayıya kadar olan sayı aralığında; tek sayıların Toplamı ve Çarpımları ile, çift Sayıların karelerinin Toplamlarını Ekrana yazdıracak olan algoritma ve akış diyagramını tasarlayınız...

BUGÜNLÜK BU KADAR  
BİR SONRAKİ DERSTE  
GÖRÜŞMEK ÜZERE... :)