Algoritma

Adım adım

Açıkça tanımlanmış

Her seferinde bir basit iş

Bilgisayara ne yapacağını söyle

İşe gitmek için ne yapmalı?

Bu örnek rotayı açıklar

bir kişi evden çıkmak için alacak

Ve

işe gitmek

Ve

işyerine girdiklerinde ilk yapacakları şey.

İşe gitmek için;

evden çık

Otobüs durağına yürüyün

Otobüsü gideceğiniz yöndeki durakta bekleyin.

Otobüsünüz geldiğinde otobüse binin

Biletinizi bilet kutusuna koyun

İniş yaptığınız yere yaklaştığınızda geri yürüyün

İneceğinizi gösteren uyarı ışığına basın

Otobüs durunca inin

İş yerinize yürüyün

işyerinin giriş kapısından girin

İş arkadaşlarınıza merhaba deyin

iş kıyafetleri giyin

işine başla

Pisedokod

Sözde kodun ne anlama geldiğini tartışalım ve tahmin etmeye çalışalım

Sözde kodlar, bir algoritmayı temsil etmenin iki popüler yolundan biridir.

Sözde kod, bir bilgisayar programını veya bir algoritmayı temsil etmenin resmi olmayan bir yoludur.

Bir programlama dili gibi görünse de, çalıştırılabilmesi için bir programlama dilinde yazılması gerekir. Dilden bağımsızdır.

Sözde kod yazmak, temel olarak programınızın yapmasını istediğiniz şeyi İngilizce olarak yazmaktır.

Bir programlama dilinin genel stilini taklit etmeyi amaçlar

\*\*\* Yaygın olarak kullanılan bu anahtar kelimeler vardır, kendi anahtar kelimelerinizi kullanabilirsiniz, ancak bunlar diğer bilgisayar programcıları arasında en sık kullanılanlardır ve değişken adı olarak kullanılmamalıdır.

Keyword

Start

İnput

Print…..

Örnek

Normalde iki sayının toplamı dediğimiz zaman pisedocod da nasıl yazarım

Start

Output “sayıyı gir” girilen sayıyı RAM de saklayacak, onun için bir değişkene atamalyım.

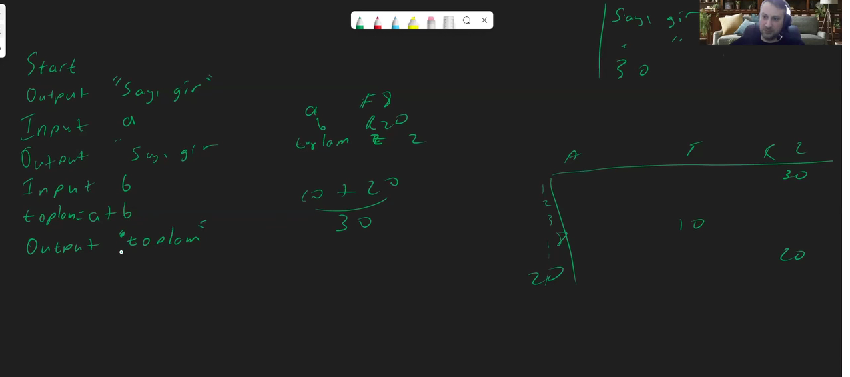
İnput a değişkenler benim bellekteki adrese erişmemi sağlıyor.

Output “sayı gir”

İnput b

toplam = a+b

Output toplam tırnak ile yazmalıyım ki toplamın değerini getirsin



Veya nasıl yazılır ?

Start

İnput “sayı gir”, a

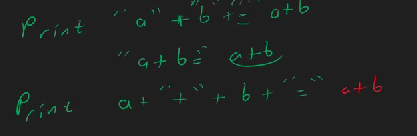
İnput “ 2 sayıyı gir”, b

Toplam=a+b

Output “toplam=” toplam

End

Print “a+b = “ a+b



Mary'nin maaşını hesaplamak için bir sözde kod yazalım.

İnput, çalıştığı saat ve ücreti

Output, ödeme

START

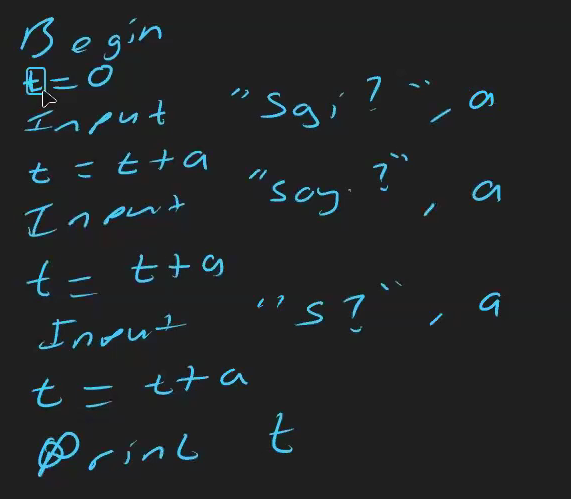
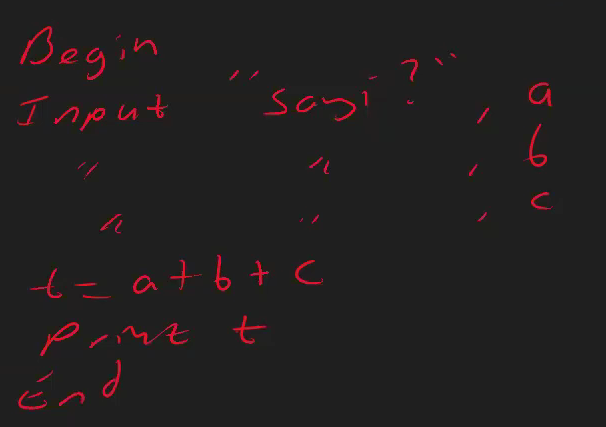
INPUT hours, rate

pay=hours \* rate

OUTPUT pay

END

Begin, başlamak demek

İkisinden hangisi daha çok yer kaplarsa ondan kaçınmam gerek.

İF ELSE

Conditation

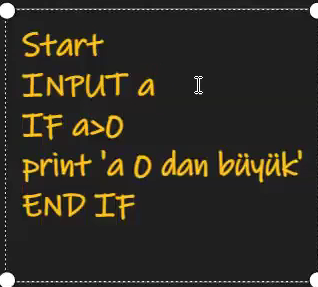
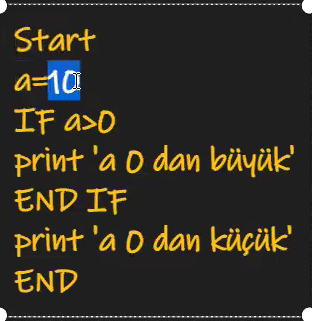
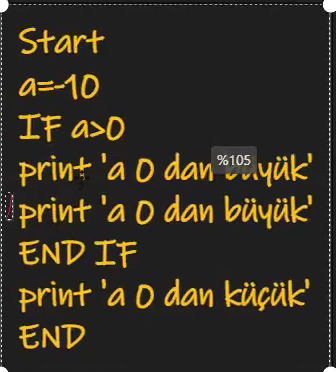
Karar vermemiz gereken durumlar olabilir

Karar yapısı, bir bilgisayar programında programın bir karar vermesine ve bu karara dayalı olarak davranışını değiştirmesine izin veren bir yapıdır.

\*\*

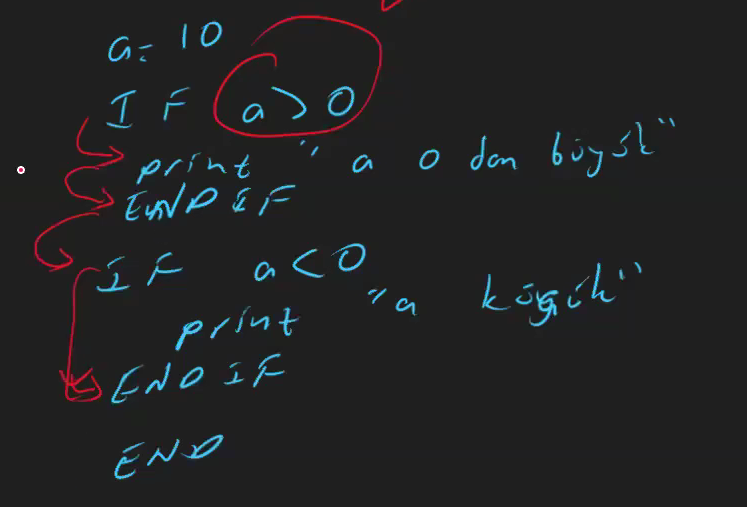
Bu anahtar kelime, yaklaşan bloğun yürütülmesi için belirli bir koşulun karşılanması gerekiyorsa kullanılır.



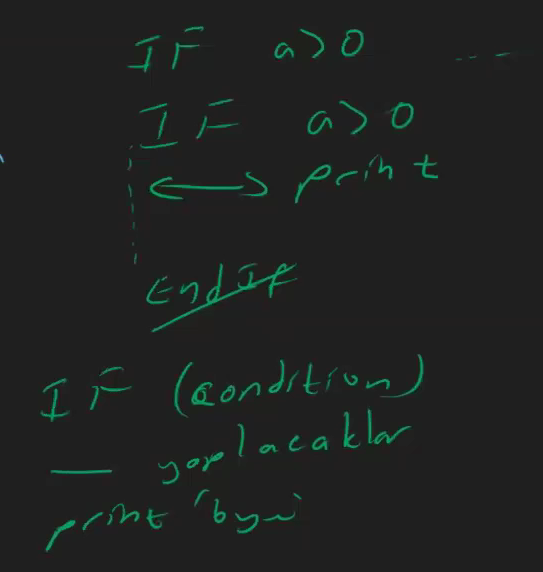
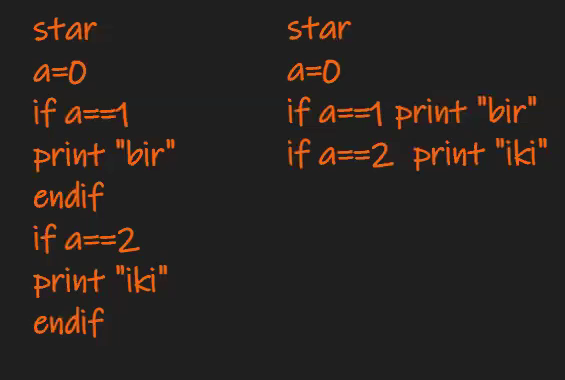
  



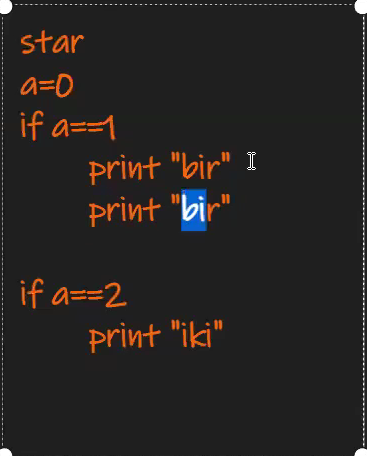
Gördüğünüz gibi “smile” ifadesinin üstündeki if ifadesinin içinde yürütüldüğünü bildirmek için girinti kullanıyoruz.



Tekrar tekrar endi f yazacağımıza if koşulunu girinti içine yazmalıyız.

Yani sağdaki gibi yazılan bir psedokod aşağıdaki gibi de yazılabilir çünkü daha sade olur, Python da

 girinti olunca end if yazmaya gerek yoktur.

İF - ELSE

Yorgun isen

Dinlen o zaman

Değilsen

Çalışmaya devam et

Sitresliysen

Rahatla o zaman

\*\*

Mutluysan

Gülümse

Sinirliysen, kızgınsan

Sakin ol

Değilse

Mutlu olmaya çalış

\*\* Girdi olarak bir sayı alan ve 10'dan büyükse doğru, aksi takdirde yanlış yazdıran bir sözde kod yazın.

INPUT num

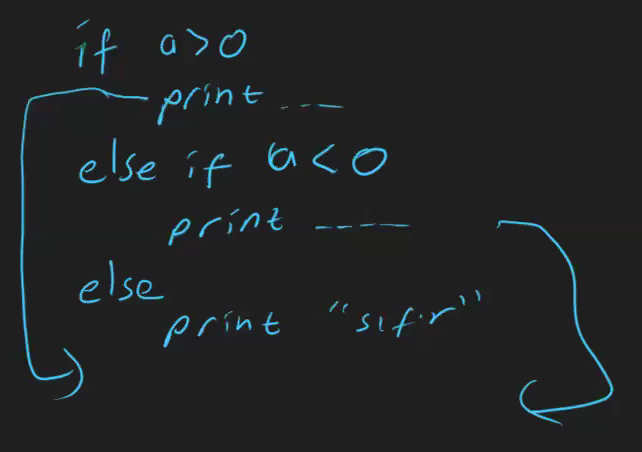
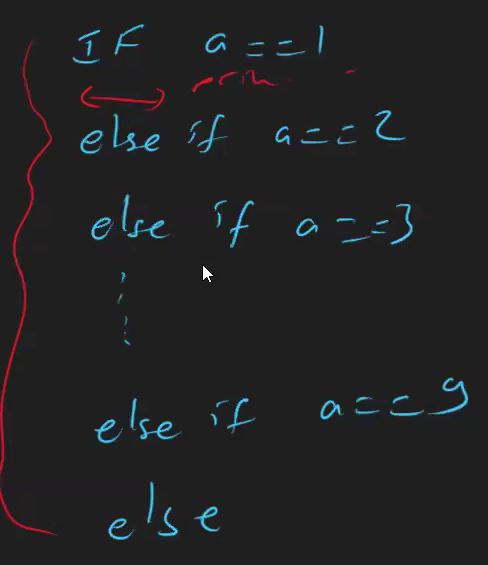
if num > 10

print "true"

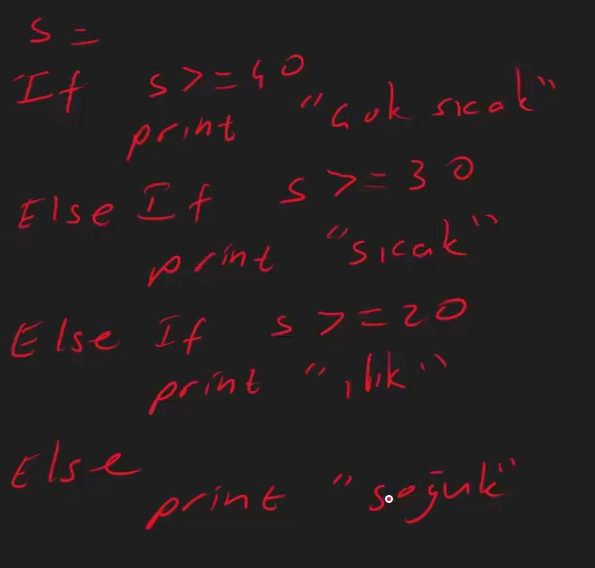
else

print "false"

Örnek verelim

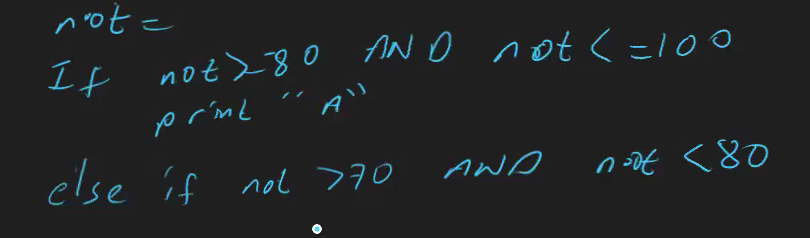
 

Sıcaklık



Peki not olsaydı ne olacaktı ?

Ders notları 80 ve 100 arası şu şu olsun diye yapmış olacaktı



\*\*

\*\* Bir sayı alır ve sayının pozitif, negatif veya nötr olduğunu gösterir

START

Input number

IF number>0

print ‘positive’

ELSE IF number<0

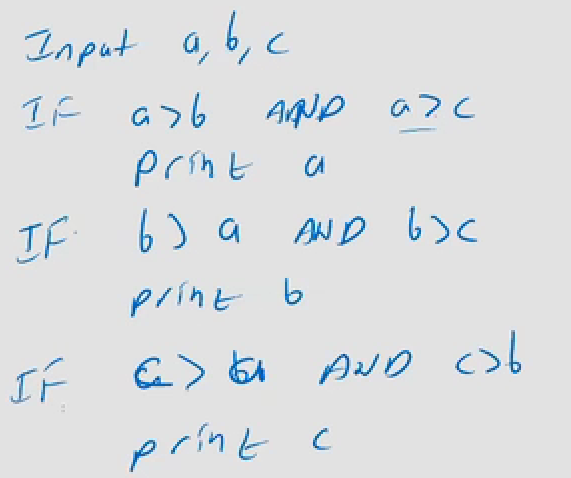
print ‘negative’

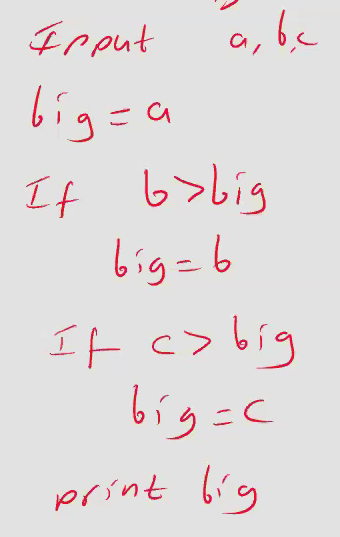
ELSE

print ‘nötr’

END

\*\* Girilen 3 sayıdan en büyüğünü bulun.

 böyle yazmak çok yer kaplar, bunu biraz daha kısaltabiliriz



\*\*Öğrencinin girdiği nota göre başarılı mı yoksa başarısız mı olduğunu öğrenin.

START

INPUT “enter student grade”, grade

IF grade >=0 and grade<50

print “successful”

else if grade >=50 and grade <=100

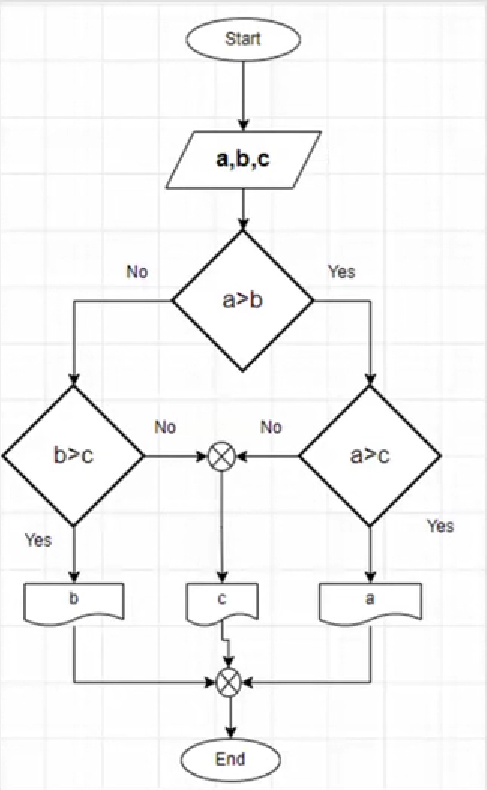
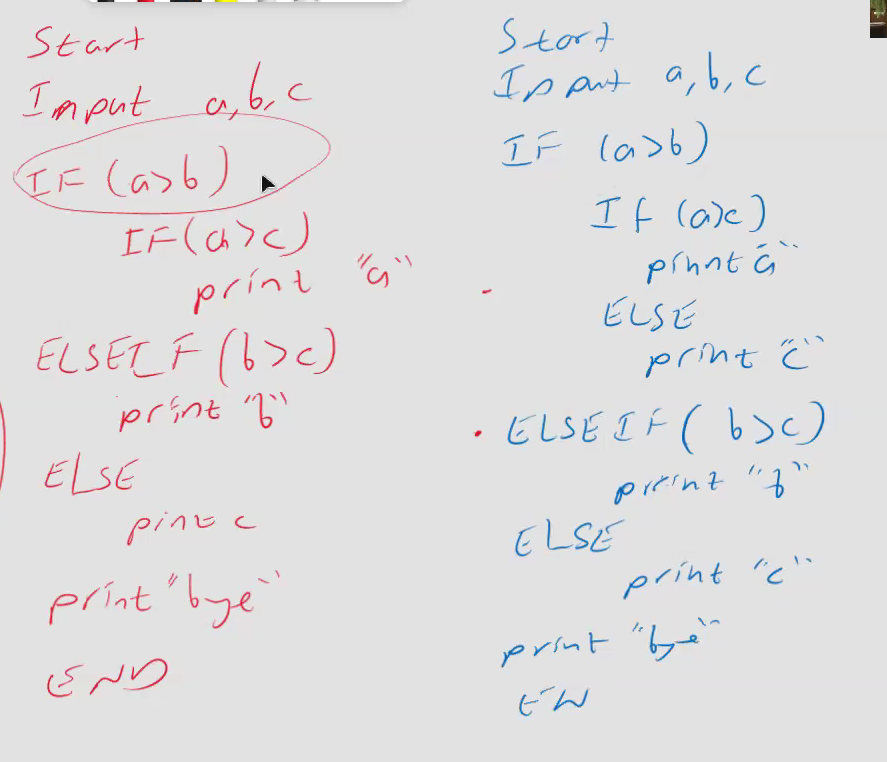
print “unsuccessful”

else

print “incorrect entry”

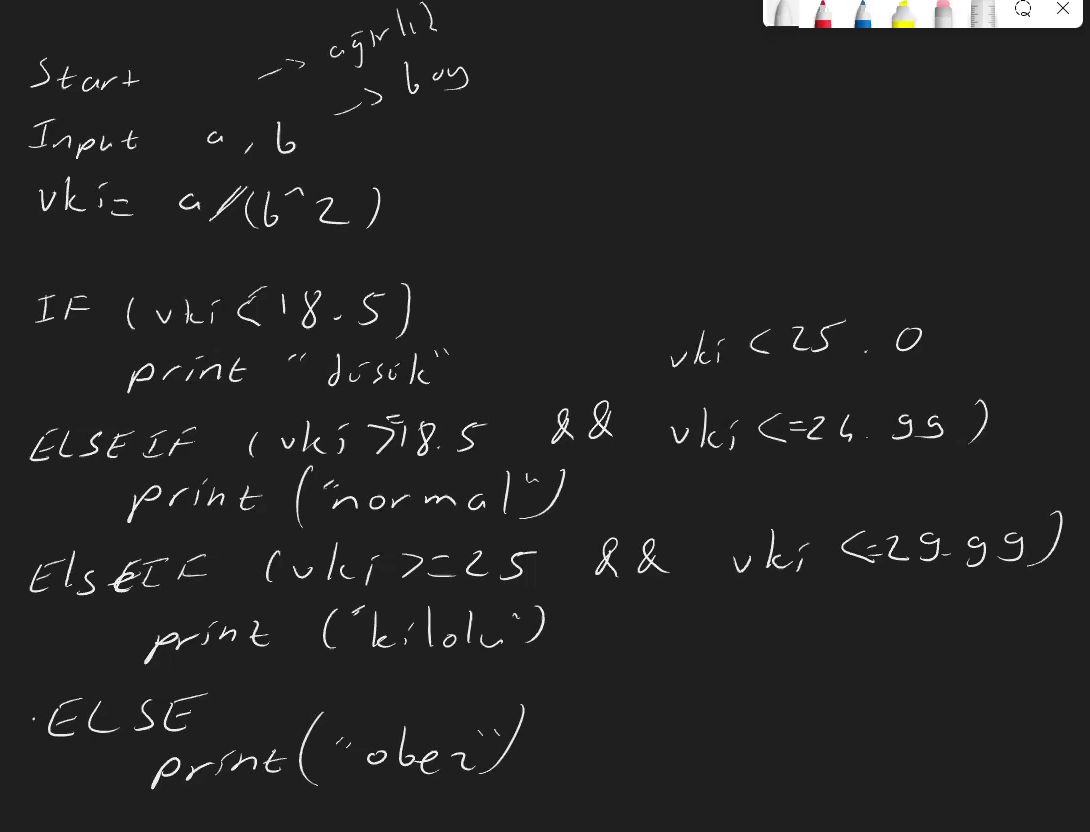
End

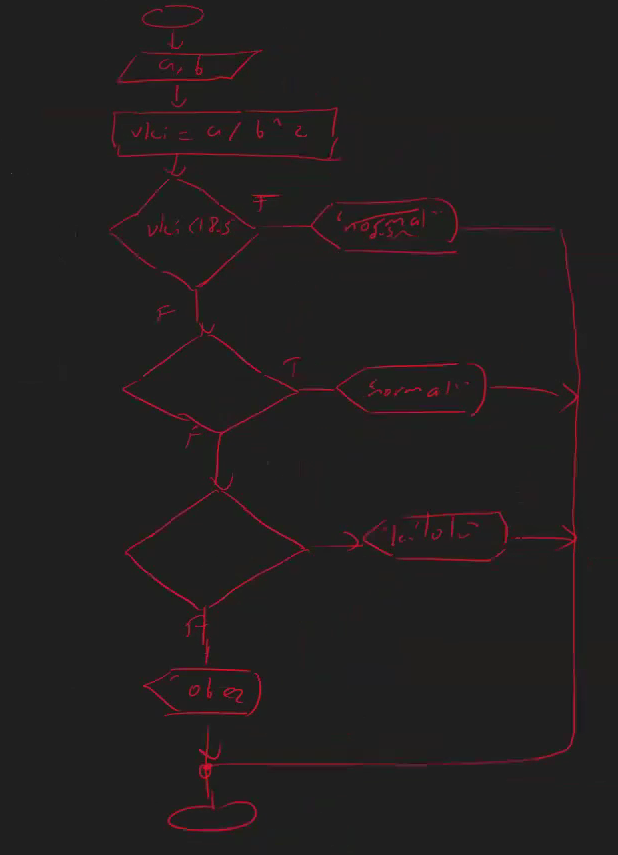
Örnek olarak bunun psedocod unu çıkaralım

\*\* Öğrencinin girdiği nota göre başarılı mı yoksa başarısız mı olduğunu öğrenme

\*\* Kişinin boy ve kilo değerlerini alarak vücut kitle indeksinin sınıflandırmasını bulan bir sözde kod yazınız.





\*\* Tabloda verilen bilgilere göre nakliye ücretini hesaplayın

LOOPS -- DÖNGÜLER

Döngü, belirli bir koşula ulaşılana kadar sürekli olarak tekrarlanan bir talimat dizisidir.

For döngüsü, bir grup içindeki her öğe için çalışır.

\*\* Bulaşıkları yıkayalım. Tüm araçlara sahip olduğumuzu düşünün vb.

kirli bulaşıkları topla

çevrenizde bulaşık makinesi varsa

kirli bulaşıkları bulaşık makinesine koyun

bulaşık makinesinin ayarlarını yapmak

ayarlanan süre henüz bitmemişken

Beklemek

else

bulaşıklar temiz değilken

bulaşıklardan birini al

elinle yıka

kurutun ve bir kenara koyun

While, for döngüsüne benzer, farklı olarak, sağlanan koşul sağlanamayana kadar döngüyü çalıştırır.

Örnek:

elma = 5

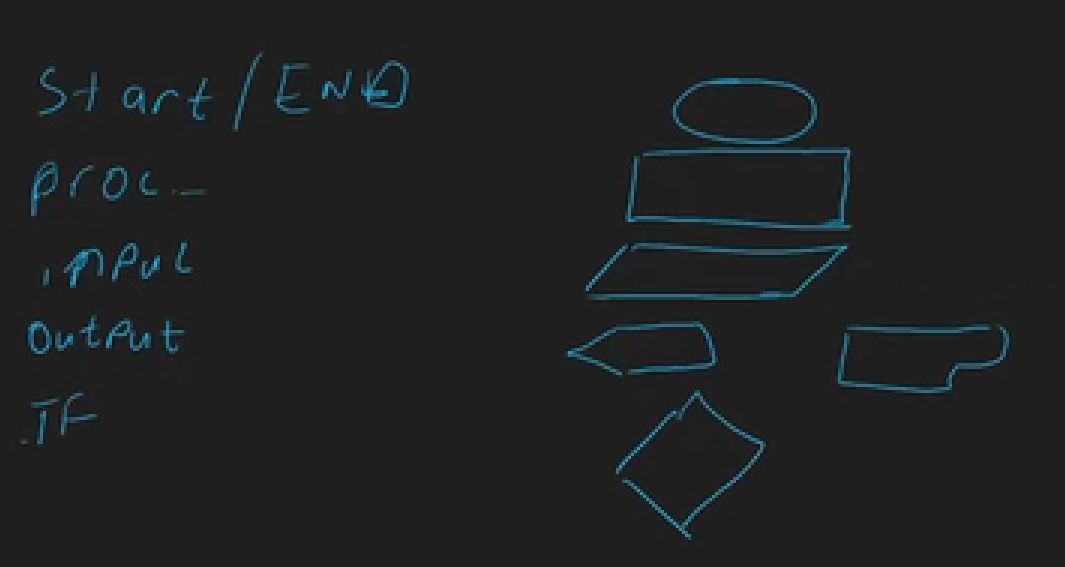
portakal = 10

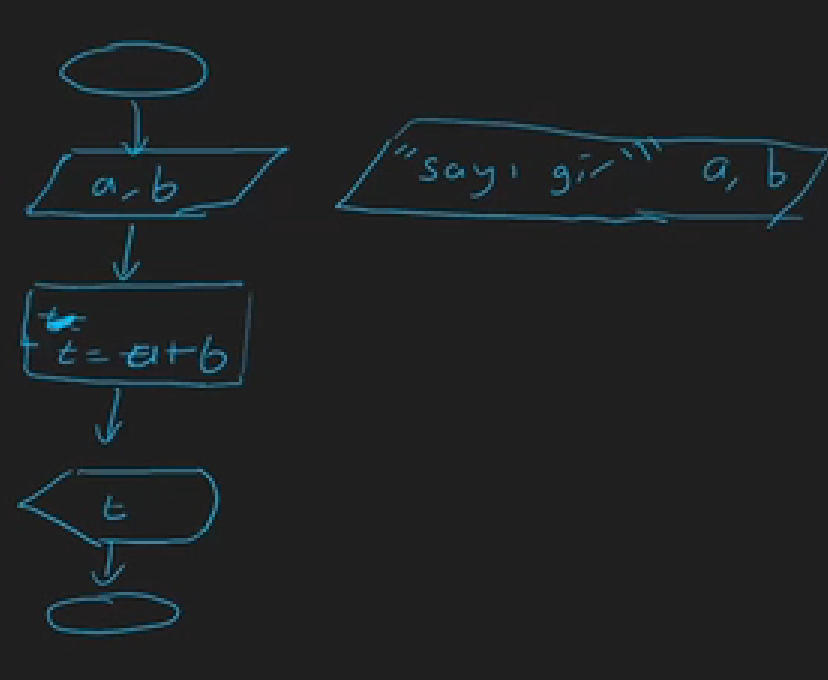
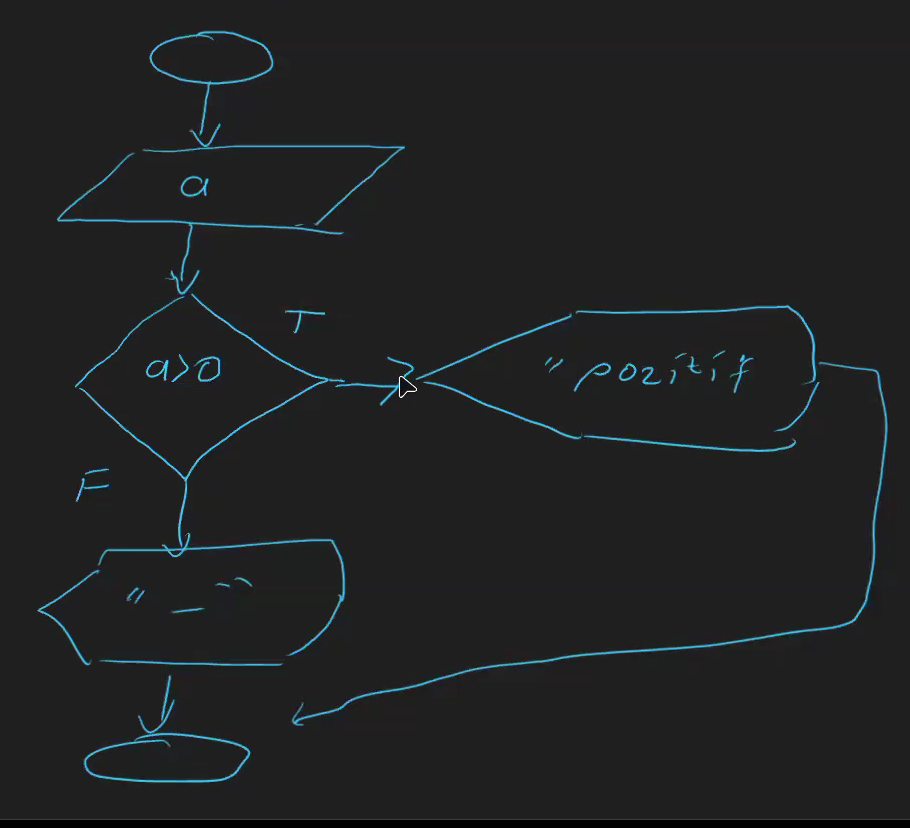
elma < portakal iken

elmaları artırmak;

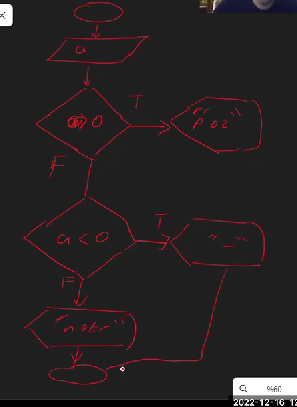
son zamanlarda

FLOW CHART



iki sayının toplamı pozitif mi negatif mi



gitmind den çizebiliriz, akış şemasını