

HÜPP PYTHON I.HAFTA

ALGORİTMA MANTIĞI, AKIŞ DİYAGRAMLARI
VE PYTHON'A GİRİŞ

PROGRAMLAMAYA GİRİŞ

- Herhangi bir program yazabilmemiz için öncelikle önümüzde bir problem, soru olması gerekir. Problemi belirledikten sonra ise izlenecek adımlar;

- Sorun analizi
- Algoritma hazırlanması
- Akış diyagramının çizilmesi
- Programın kodlanması,
- Hata analizi/test etme,
- Probleme uygulanması

Şeklinde sıralanabilir.

SORUN ANALİZİ

- Giriş-Çıkış birimleri
- Programda kullanılacak değişkenler ve türleri
- Sorun çözüm yolu

ÖRNEK

- Klavyeden girilen iki sayının toplamının ekrana yazdırılması problemi nasıl çözülür?

CEVAP:

I. İstenilenler;

- I) Sayıların klavyeden girilmesi
- ii) Toplamın bulunması,
- iii) Toplamın ekranda görülmesi

II. Değişken Tanımı

- I) Birinci sayı(A),
- ii) İkinci sayı(B),
- iii) İki sayının toplamı (T)

III. Çözümün matematiksel yazımı

$$T=A+B$$

ALGORİTMA

Algoritmanın Hazırlanması

Algoritma, herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir. Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir. Diğer bir deyişle algoritma, verilerin, bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde tanımlanabilir.

Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları gözönünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdır. Aşağıda algoritma hazırlanmasına ilişkin örnekler yer almaktadır.

Örnek 1: Verilen iki sayının toplamının bulunmasının algoritması aşağıdaki gibi yazılır.

Algoritma

Adım 1 – Başla

Adım 2 – Birinci Sayıyı Oku

Adım 3 – İkinci Sayıyı Oku

Adım 4 – İki Sayıyı Topla

Adım 5 – Dur

Algoritmaya dikkat edilirse işlemlerin sıralanmasında, işlem önceliklerinin göz önünde bulundurulduğu görülür. Ayrıca algoritma yazımı sorun çözümünün başladığını gösteren “Başla” ifadesi ile başlamakta ve işlemlerin bittiğini belirten “Dur” ifadesi ile sona ermektedir.

AKIŞ DİYAGRAMLARI

Herhangi bir sorunun çözümü için izlenmesi gerekli olan aritmetik ve mantıksal adımların söz veya yazı ile anlatıldığı algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline “akış şemaları” veya FLOWCHART adı verilir. Akış şemalarının algoritmadan farkı, adımların simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir.

Programın saklanacak esas belgeleri olan akış şemalarının hazırlanmasına, sorun çözülmesi sürecinin daha kolay anlaşılır biçime getirilmesi, iş akışının kontrol edilmesi ve programın kodlanmasının kolaylaştırılması gibi nedenlerle başvurulur. Uygulamada çoğunlukla, yazılacak programlar için önce programın ana adımlarını (bölümlerini) gösteren genel bir bakış akış şeması hazırlanır. Daha sonra her adım için ayrıntılı akış şemalarının çizimi vardır.

Akış şemalarının hazırlanmasında aşağıda yer alan simgeler kullanılır.

Algoritmanın başladığını
ya da sona erdiğini
belirtmek için kullanılır.

Klavye aracılığı ile
giriş ya da okuma
yapılacağını gösterir.

Yazıcı aracılığı ile
çıkış yapılacağını gösterir.

Kart okuyucu aracılığıyla
giriş yapılacağını gösterir.

Araç belirtemeden
giriş ya da çıkış
yapılacağını gösterir.

Hesaplama ya da değerlerin
değişkenlere aktarımını gösterir.

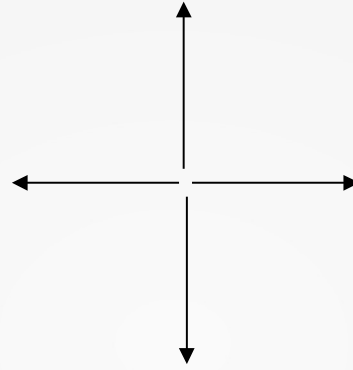
Aritmetik ve mantıksal
ifadeler
için karar verme
ya da karşılaştırma
durumunu gösterir.

Diskten okuma
veya diskete
yazmayı
gösterir.

Disketten okuma
veya diskete
yazmayı
gösterir.

Teyp kütüğünü
gösterir.

Yapılacak işler birden fazla sayıda
yinelemek ise, diğer bir deyişle
iş akışında çevrim (döngü)
var ise bu sembol kullanılır.



Oklar işin akış yönünü
gösterir.

AKIŞ ŞEMALARI

Akış şemaları içerik ve biçimlerine göre genel olarak üç grupta sınıflandırılabilirler.

- Doğrusal Akış Şemaları
- Mantıksal Akış Şemaları
- Döngüsel (iteratif, çevrimli, yineli) Akış Şemaları

DOĞRUSAL AKIŞ ŞEMASI

İş akışları giriş, hesaplama, çıkış biçiminde olan akış şemaları bu grup kapsamına girer.

Örnek 1: İki sayının çarpımının bulunmasıyla ilgili algoritma aşağıdaki gibidir.

Değişkenler:

A: Birinci sayıyı

B: İkinci sayıyı

C: İki sayının çarpımını ($A*B$) gösterecek

Algoritma:

Adım 1 – Başla

Adım 2 – A'yı oku

Adım 3 – B'yi oku

Adım 4 – $C=A*B$ 'yi hesapla

Adım 5 – C'yi yaz

Adım 6 - Dur

DOĞRUSAL AKIŞ ŞEMASI

- Örnek 2: Klavyeden girilen İki sayının toplamını hesaplayıp yazan algoritma ve akış şemasını hazırlayınız.

(X: Birinci sayı, Y: İkinci sayı, Z: toplam)

A1 : Başla

A2 : Klavyeden oku X

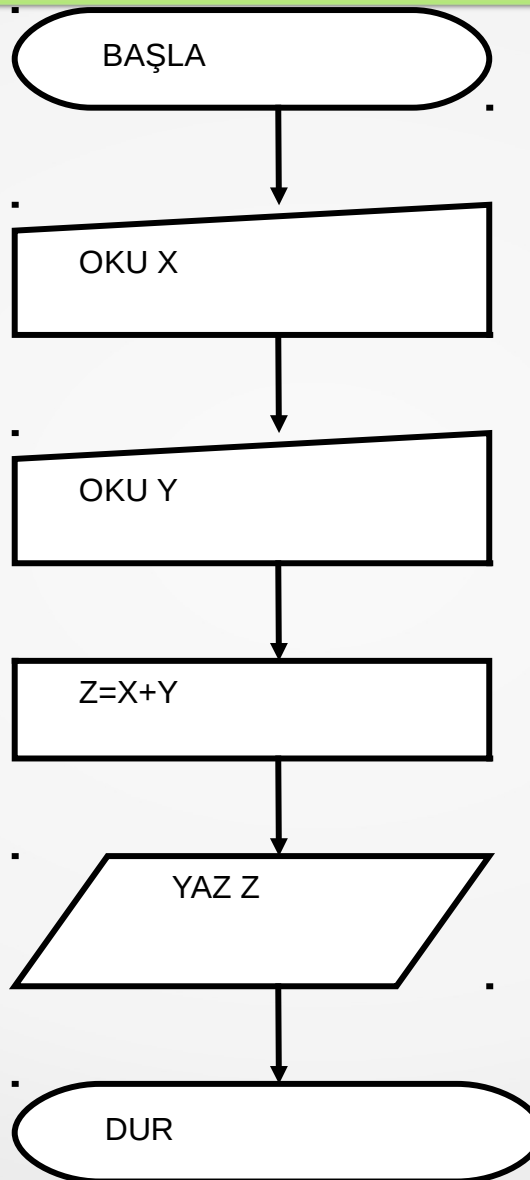
A3 : Klavyeden oku Y

A4 : Hesapla $Z = X + Y$

A5 : Yaz Z

A6 : Dur

DOĞRUSAL AKIŞ ŞEMASI



MANTIKSAL AKIŞ ŞEMASI

- Geniş ölçüde mantıksal kararları içeren akış şemalarıdır. Hesap düzenleri genellikle basittir.

Örnek 3: Klavyeden girilen bir sayının pozitif, negatif veya sifıra eşit olma durumunu hesaplayıp yazdıran algoritma ve akış şemasını hazırlayınız.

(S : Sayı)

A1 : Başla

A2 : Oku S

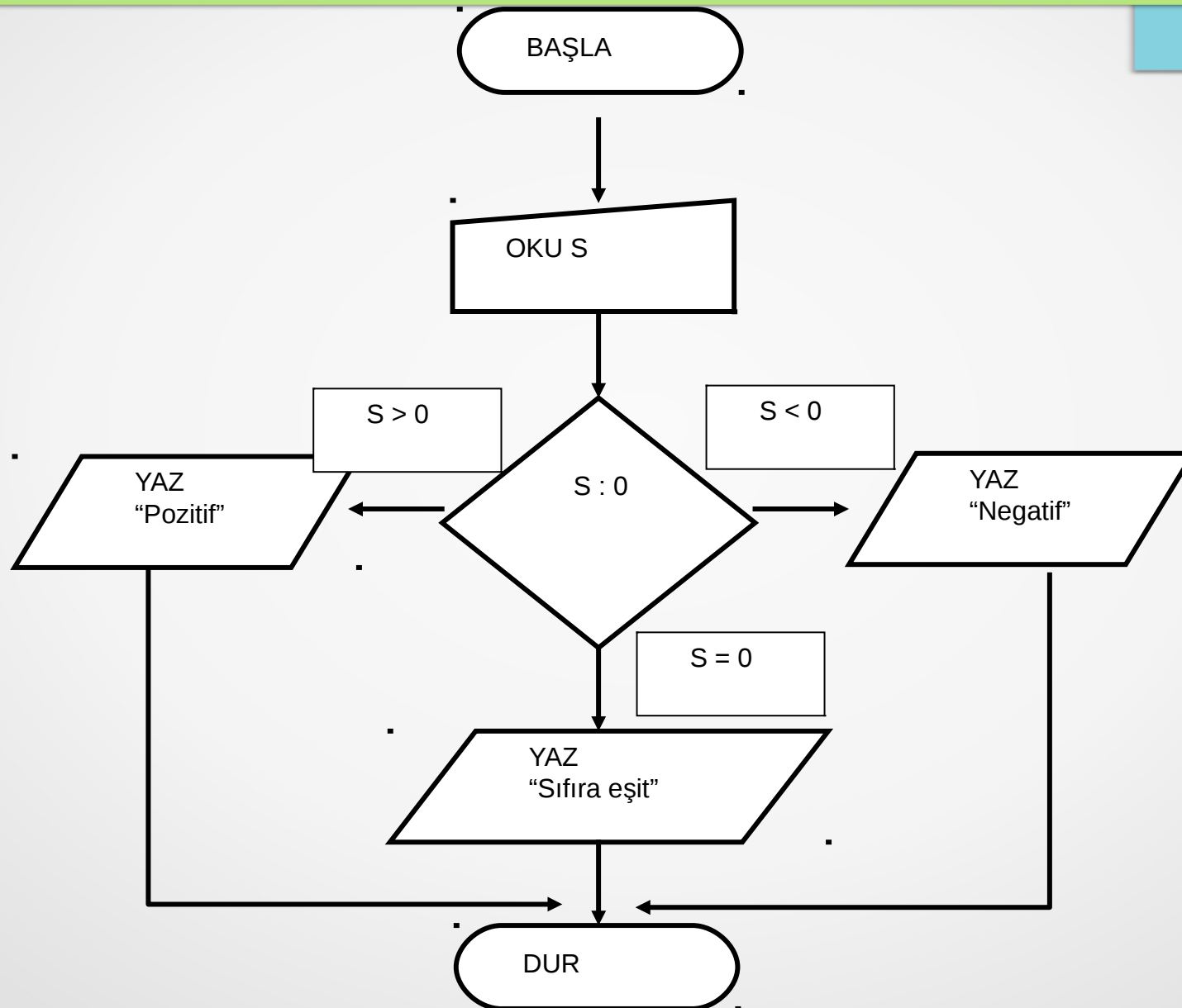
A3 : Eğer $S > 0$ ise "Pozitif" yaz,

A4 : Eğer $S < 0$ ise "Negatif" yaz,

A5 : Eğer $S = 0$ ise "Sıfıra eşit" yaz,

A6 : Dur

MANTIKSAL AKIŞ ŞEMASI



DÖNGÜSEL AKIŞ ŞEMASI

Sorunun çözümü için, çözümde yer alan herhangi bir adım ya da aşamanın birden fazla kullanıldığı akış şemalarına denir. İş akışları genel olarak giriş ya da başlangıç değeri verme, hesaplama, kontrol biçiminde olmaktadır.

DÖNGÜSEL AKIŞ ŞEMASI

Örnek 4: Klavyeden girilen bir yazıyı 5 kez yazdıran algoritma ve akış şemasını oluşturunuz.

(Y : Yazı, S : Sayaç)

A1 : Başla

A2 : Oku Y

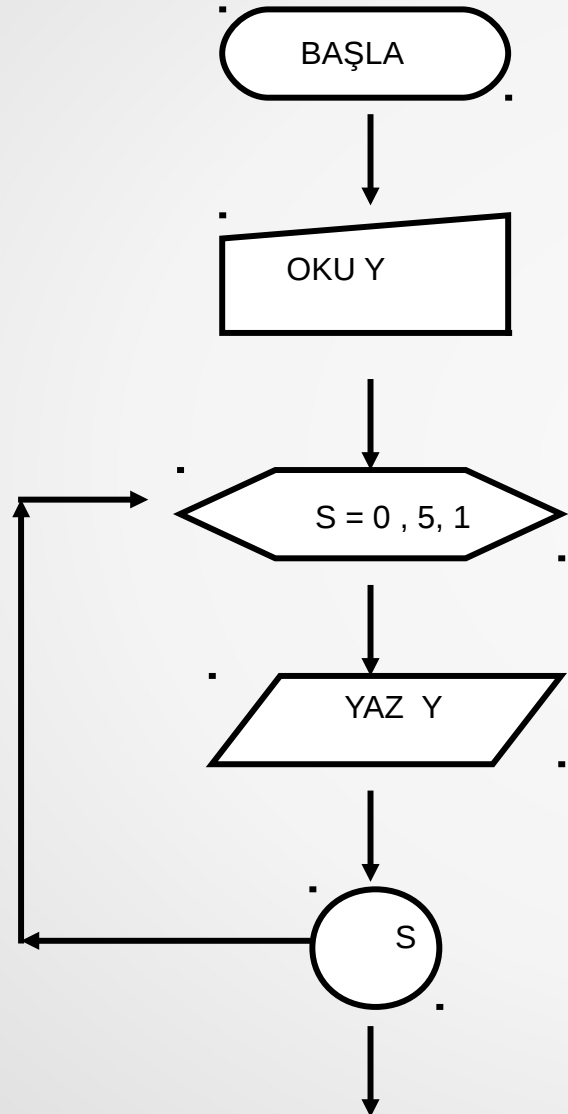
A3 : Yaz Y

A4 : $S = S + 1$

A5 : Eğer $S < 5$ ise A3 e git

A6 : Dur

DÖNGÜSEL AKIŞ ŞEMASI



Sayısı bilinen
döngülerde;
başlangıç değeri,
son değeri, artış
sayısı

Döngünün çalıştığı
bölüm