

PROGRAMLAMAYA GİRİŞ

DERS NOTLARI

PROGRAM

- ▶ Bilgisayar en basit olarak üç ana görevi yerine getiren bir makinedir. Girilen bilgiyi alır (INPUT), işler (PROCESSING) ve bu işlenmiş veriden bir sonuç (OUTPUT) çıkarır. Bilgisayar, sadece donanım olarak çalışmaz. Çünkü yazılım olmadan, donanım ne yapacağını bilemez. Bilgisayar donanımına ne yapacağını söyleyecek bir komutlar dizisi gerekir.
- ▶ Yapacağı görevleri, ona anlatan komutlara program diyebiliriz.
- ▶ Program, belli bir komut ve söz dizimi yapısını uygun olarak, sadece belirtilen işlemlerin yerine getirilmesini belirten komut dizisidir.

PROGRAMLAMA DILI

- ▶ Programlama Dili bilgisayarda çözülecek bir sorun için çözümün bilgisayara adım adım yazılmasını sağlayan biçimsel kuralları olan ve bu kurallara sıkı sıkıya bağımlılığı gerektiren bir tanımlar kümesidir.
- ▶ Belki daha kısa bir tanımla sizinle bilgisayar arasında bir tercümandır demek doğru olur.

C PROGRAMLAMA DİLİ

- ▶ C Programlama Dili genel amaçlı orta seviyeli ve yapısal bir programlama dilidir.
- ▶ 1972 yılında Dennis Ritchie tarafından Bell Telefon Laboratuvarında tasarlanmıştır.
- ▶ İşletim sistemleri, derleyiciler ve debug programları gibi aşağı seviyeli sistem programlarının yazılımında yoğun olarak C programlama dili kullanılır.
- ▶ Bu dil ilk olarak bir programlama dili olarak düşünülmemiştir ve özel bir amacı vardı: UNIX işletim sistemini tasarlamak (UNIX işletim sisteminin 1000 satırlık bölümü C ile yazılmıştır).
- ▶ Günümüzde Nesneye Yönelik programlama dilleri (C++, Java gibi) ve script dilleri (JavaScript, JavaApplet, PHP gibi) gibi programlama dilleri tamamen C tabanlıdır.
- ▶ Kısacası Standart C dilini (ANSI C) öğrenmekle bu dillerin tamamına iyi bir hazırlık yapmış olursunuz. Bu derste ANSI C ile programlama konu edilmiştir.

NEDEN C ?

- ▶ C güçlü ve esnek bir dildir. Esnek olması yazacağınız programda daha dikkatli olmanız gerektiği anlamına gelir. Unutulacak bir işaret belki derleyici hatasına neden olmayacaktır ancak programınızı da doğru çalıştırmayacaktır.
- ▶ C ile işletim sistemi yazabilir, kelime işlemciler oluşturabilir veya grafik çizebilirsiniz.
- ▶ C taşınabilir bir dildir. Yani herhangi bir C kodu hiçbir değişikliğe uğramadan, veya çok az bir değişimle, başka bir derleyicide derlenebilir. Örneğin, Windows işletim sistemlerinde yazılan bir C kodu, Linux, UNIX veya VAX gibi işletim sistemlerinde de derlenebilir.
- ▶ C yapısal bir dildir. C kodları fonksiyon olarak adlandırılan altprogramlardan oluşmuştur.

DERLEYİCİ

- ▶ Programda belirtilen işlem dizisi doğrudan işleme konmaz.
- ▶ Bilgisayar, verilen programı, makine dilinde işleme almak zorundadır.
- ▶ Programcı programlama dili biliyor, bilgisayar ise makine dili biliyor. Programcının, bilgisayara ne demek istediğini kim tercüme edecek yani bilgisayar ile programcı arasındaki iletişim nasıl sağlanacak?
- ▶ Bu tercüme derleyici adı verilen çeviri programları yapar. Bu sayede programlama dilleri ile yazılmış komutlar dizisinden oluşan kaynak program makine diline derlenir.

PROBLEM ÖZME

- ▶ Problem özmede soruna hemen girişmek yerine, dikkatli ve sistematik yaklaşım ilke olmalıdır.
- ▶ Problem iyice anlaşılmalı ve mümkün olduğu kadar küçük parçalara ayrılmalıdır.

«DISCOURSE ON METHODS» DESCARTES

- ▶ Doğruluđu kesin olarak kanıtlanmadıkça, hiçbir şeyi doğru olarak kabul etmeyin; tahmin ve önyargılardan kaçının
- ▶ Karşılaştığınız her güçlüđu mümkün olduğu kadar küçük parçalara bölün
- ▶ Düzenli bir şekilde düşünün; anlaşılması en kolay olan şeylerle başlayıp yavaş yavaş daha zor ve karmaşık olanlara doğru ilerleyin
- ▶ Olaya bakışınız çok genel, hazırladığınız ayrıntılı liste ise hiçbirşeyi dışarıda bırakmayacak kadar kusursuz ve eksiksiz olsun.

ALGORİTMALAR

- Bir problemin çözümüne yönelik hazırlanacak olan programa ilişkin sonlu sayıda ve anlaşılır adım yada işlemin ardı ardına tanımlanması ve böylece izlenecek yolun veya yöntemin ortaya konulması için hazırlanan plana algoritma denir.

ALGORİTMALAR

- ▶ Sıralı olmalıdır. Bir başlangıç noktasından başlayarak, sonlu sayıda, belirsizlik içermeyen adımlardan sonra sona erebilmelidir.
- ▶ Problemlerle ilgili olarak ortaya çıkabilecek her türlü durumu içermelidir.
- ▶ Aynı tür problemler için de geçerli olmalıdır.

Bir insanın evden ıkıp iŖe giderken izleyeceđi yolu tanımlayan algoritma

1. Evden dıŖarıya ık
2. Otobüs durađına yürü
3. Durakta gideceđin yöndeki otobüsü bekle
4. Otobüsün geldiđinde otobüse bin
5. Biletini bilet kumbarasına at
6. İneceđin yere yakınlaŖtıđında arkaya yürü
7. İneceđini belirten ikaz lambasına bas
8. Otobüs durunca in
9. İŖyerine dođru yürü
10. İŖ yeri giriŖ kapısından ieriye gir

ALGORİTMİK YAKLAŞIM

- ▶ Algorİtmalar İki farklı şekilde kağıt üzerinde ifade edilebilirler;
 1. Pseudo Code (Kaba Kod veya Yalancı Kod veya Söзде Kod), bir algoritmanın yarı programlama dili kuralı, yarı konuşma diline dönük olarak ortaya koyulması/ tanımlanmasıdır. Bu şekilde gösterim algoritmayı genel hatlarıyla yansıtır.
 2. Akış şeması, algoritmanın görsel/şekilsel olarak ortaya koyulmasıdır. Problemin çözümü için yapılması gerekenleri, başından sonuna kadar, geometrik şekillerden oluşan simgelerle gösterir.

SÖZDE (PSEUDO) KOD

- ▶ Sözde programlar, doğrudan konuşma dilinde ve programlama mantığı altında, eğer, iken gibi koşul kelimeleri ve $>$ $=$ $<$ gibi ifadeler ile beraber yazılır. İyi bir biçimde yazılmış, sözde koddan, programlama diline kolaylıkla geçilebilir.
- ▶ Örnek: Verilen bir sıcaklık derecesine göre suyun durumunu belirten bir sözde program yazınız.
 1. Başla.
 2. Girilen Sıcaklığı Oku.
 3. Eğer Sıcaklık $<$ 0 ise Durum="Buz"
 4. Eğer Sıcaklık \geq 100 ise Durum="Buhar"
 5. Değilse Durum ="Su"
 6. Sonucu Ekrana Yaz.
 7. Son.

SÖZDE (PSEUDO) KOD

- ▶ Örnek: Bilgisayara verilecek iki sayıyı toplayıp sonucu ekrana yazacak bir program için algoritma geliştirmek istiyoruz
- ▶ 1. BAŞLA
- ▶ 2. A sayısını oku
- ▶ 3. B sayısını oku
- ▶ 4. $TOPLAM = A + B$ işlemini yap
- ▶ 5. TOPLAM değerini ekrana yaz
- ▶ 6. SON

AKIŞ ŐEMALARI

- ▶ Algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş Őekline “akış Őemaları” veya FLOWCHART adı verilir.
- ▶ Akış Őemalarının algoritmadan farkı, adımların simgeler Őeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir.
- ▶ Programın saklanacak esas belgeleri olan akış Őemalarının hazırlanmasına, sorun çözümlenmesi 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 Őeması hazırlanır. Daha sonra her adım için ayrıntılı akış Őemalarının çizimi vardır.
- ▶ En basit Őekliyle dikdörtgen kutulardan ve oklardan oluşur. Akış Őeması sembolleri ANSI (American National Standards Institute) standardı olarak belirlenmiş ve tüm dünyada kullanılmaktadır.

TEMEL DİYAGRAM ÖĞELERİ

A) Başla/Dur

Başla

Dur

Her akış diyagramında olur

B) Bilgi girişi



Bilgisayara dışarıdan
(klavyeden) bilgi/veri
girişi yapılacağını gösterir

A, B, C

C) İşlem



$c=(a^2+b^2)^{(1/2)}$

D) Döngü



i=1, 20, 3

j=30, 4, -2

k=1, 99

TEMEL DİYAGRAM ÖĞELERİ

E) Yazdır/ Çıktı



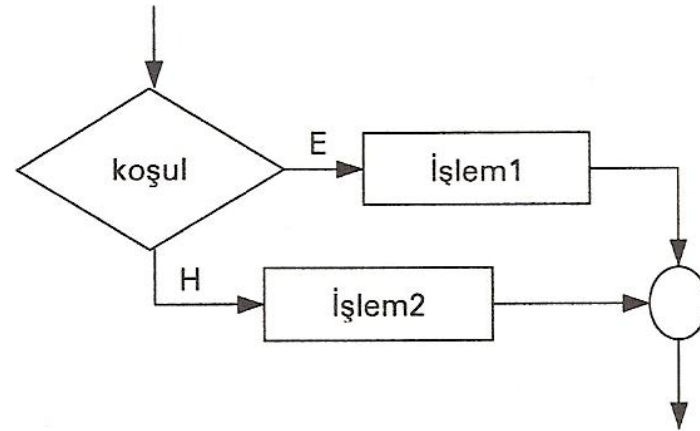
F) Bağlantı



G) İşlem akış yönü



H) Karar (Karşılaştırma)



AKIŞ ŐEMALARI

Algoritma:

A1 : Basla

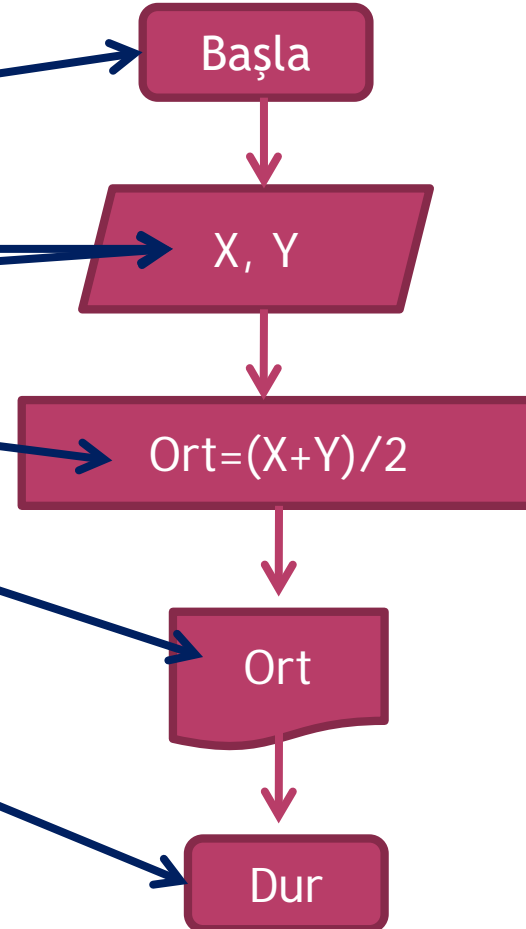
A2 : X'i oku

A3 : Y' yi oku

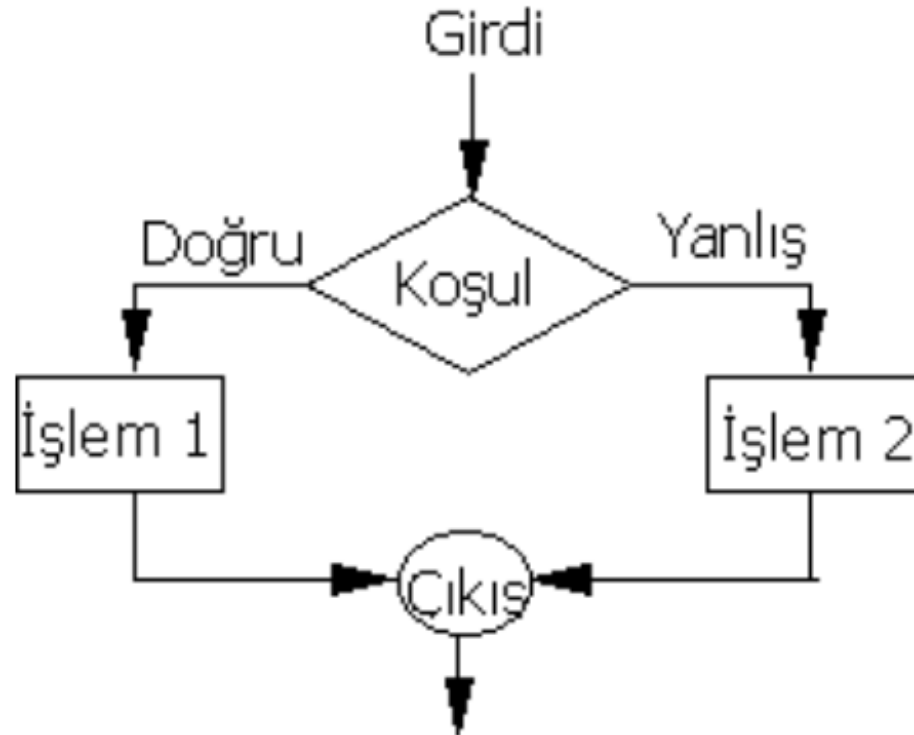
A4 : $Ort = (X+Y)/2$

A5 : Ort deęerini yaz

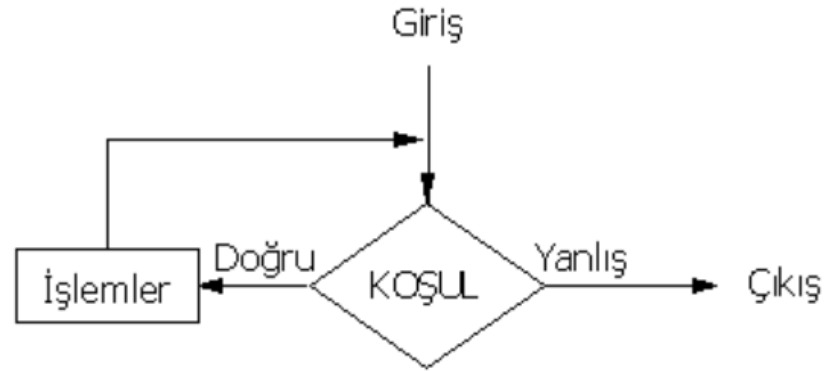
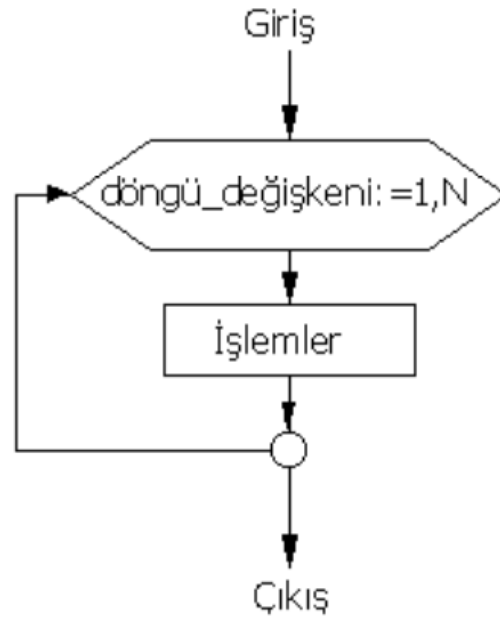
A6 : Bitir



KARAR VERME DURUMLARI



TEKRARLI DURUMLAR



Sayaç kavramı

Sayaç: Programlarda, bazı işlemlerin belirli sayıda yaptırılması gerekebilir. Bu tür sayma işlemlerine, algoritmada sayaç veya sayıcı adı verilir.

Sayaç, işlem akışı kendisine her geldiğinde, belirtilen adım değeri kadar artan/azalan değişkendir.

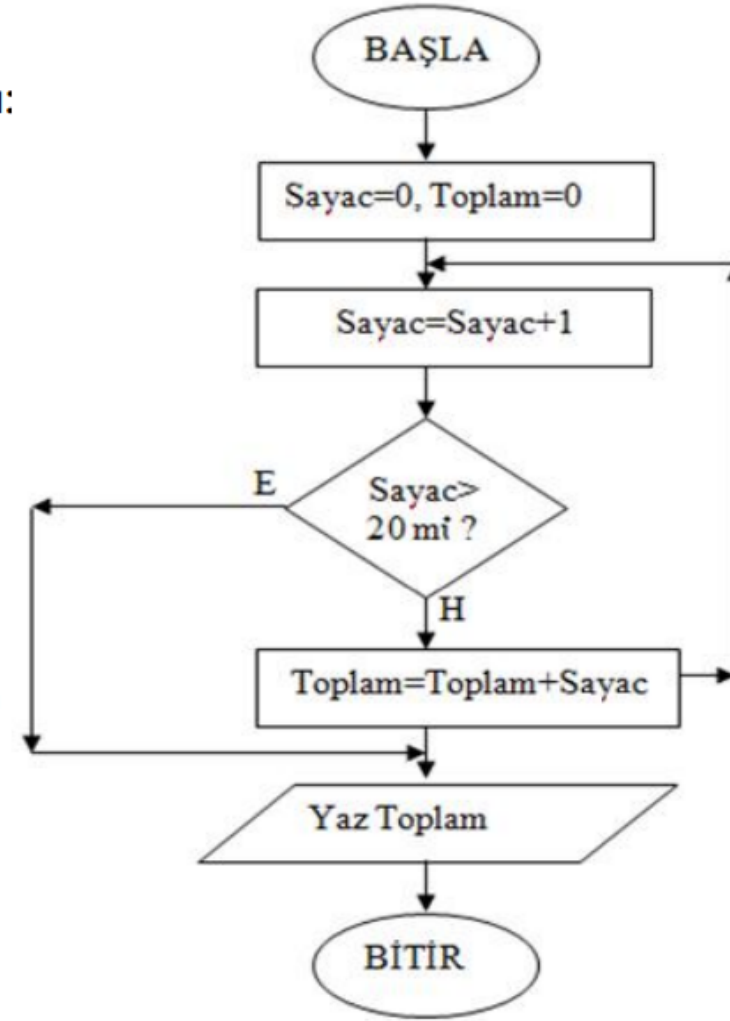
Genel kullanım (syntax)

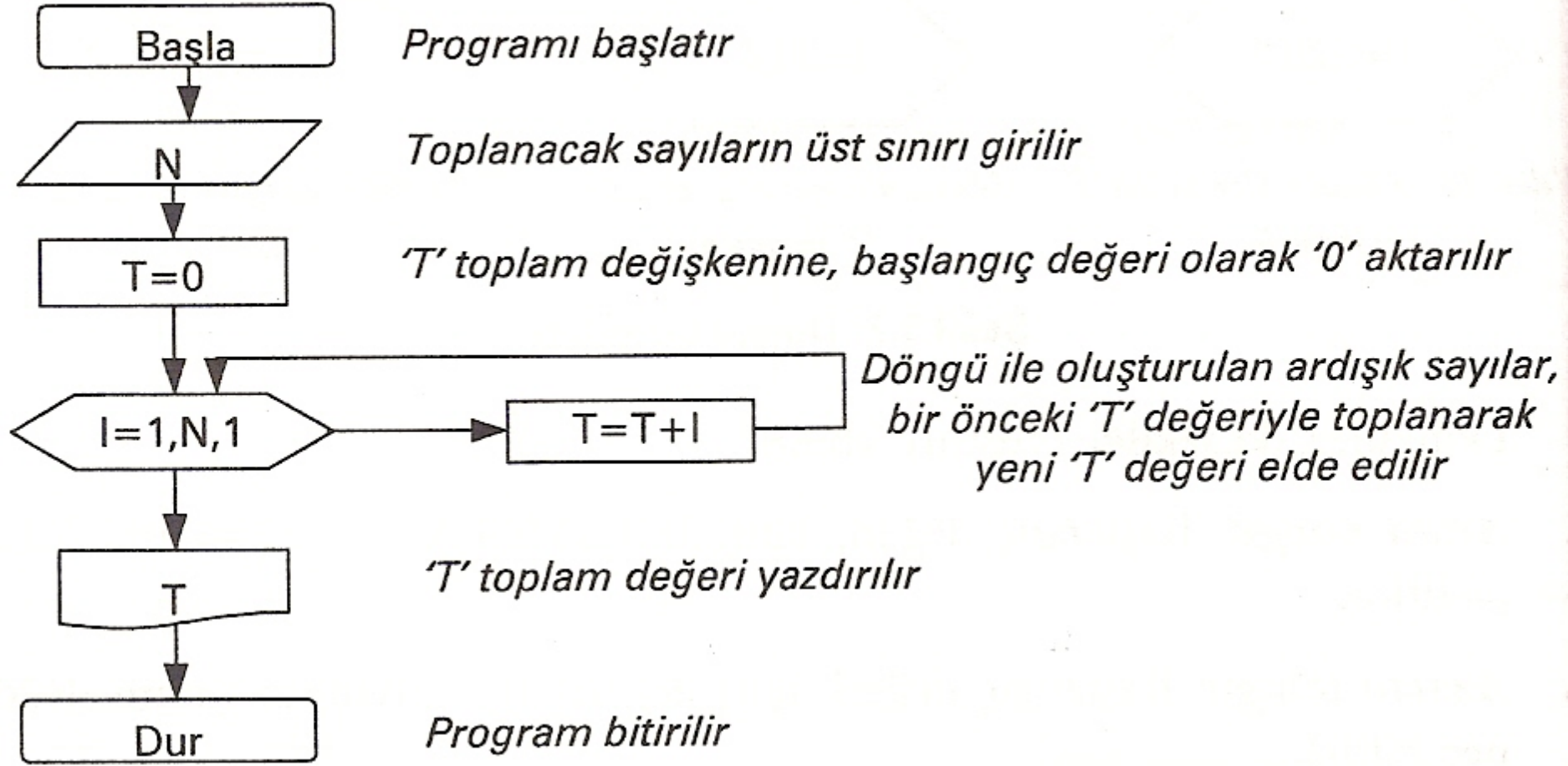
Sayac değişkeni=sayac değişkeni \pm adım

$i=i+1$, $j=j-2$, $s=s+3$

1-20 arasındaki tamsayıların toplamının bulunması:

- Adım 1. BAŞLA
- Adım 2. Sayac=0, Toplam=0
- Adım 3. Sayac=Sayac+1
- Adım 4. EĞER Sayac>20 İSE GİT 7
- Adım 5. Toplam=Toplam+Sayac
- Adım 6. GİT 3
- Adım 7. YAZ "1-20 Arası Sayıların Toplamı=", Toplam
- Adım 8. BİTİR





1. Basla

2. $T=0$

3. $J=1$

4. Eğer $J > 10$ ise git 8

5. $T=T+J$

6. $J=J+2$

7. Git 4

8. Yaz T

9. Dur

Döngü

Basla

$T=0$

$S=0$

Eğer $S>4$ ise git 9

$S=S+1$

Sayıyı (A) gir

$T=T+A$

Git 4

$\text{Ortalama} = T / (S - 1)$

Yaz Ortalama

Dur

1. Basla

2. $T=1$

3. $S=0$

4. Eğer $S>4$ ise git 9

5. $S=S+1$

6. Sayıyı (A) gir

7. $T=T+A$

8. Git 4

9. $\text{Ortalama} = T / S$

10. Yaz Ortalama

11. Dur

Kullanıcının girdiği bir sayının tek mi çift mi olduğunu ekrana yazan programın algoritması

Girilen sayı :G

Ara değişken: M

Ara değişken: cevap

Algoritma:

A1 :Basla

A2 : G'yi oku

A3 : $M = \text{mod}2(G)$

A4 : $M=0$ ise SAYI ÇİFTTİR yaz

A5 : $M=1$ ise SAYI TEKTİR yaz

A6 : Bitir

Klavyeden girilen 2 tamsayıyı, büyükten küçüğe doğru ekrana yazdıran bir program yazınız.

Problemin tanımı

Klavyeden girilen 2 tamsayı, büyük önce, küçük sonra olacak şekilde ekrana yazdırılacak.

Çözümleme

Giriş verileri:

birinciSayi /* Girilecek 1. tamsayı */
ikinciSayi /* Girilecek 2. tamsayı */

Çıkış verisi:

Giriş-çıkış verisi ilişkisi:

Çıktı görünümü:

<buyukSayi> > <küçükSayi>

Sistem çizgesi:



Tasarım

Algoritma:

1. Kullanıcıdan birinci sayıyı al.
2. Kullanıcıdan ikinci sayıyı al.
3. birinciSayı > ikinciSayı ise
 - 3.1. Ekrana birinciSayı > ikinciSayı yazdır.
4. Değilse
 - 4.1. Ekrana ikinciSayı > birinciSayı yazdır.

Akış çizgesi

