

PROGRAM YAZMAK SÜRECİ

1. Problemin farkına varmak,
2. Problemi analiz etmek,
3. Çözüm yolları düşünmek,
4. İyi çözüm yolları seçip algoritma oluşturmak,
5. Akış diyagramı çizmek,
6. Uygun bir dilde kodlamak,
7. Programı test etmek,
8. Programı dağıtmak.

ALGORİTMA NEDİR?

(Adım adım işlem basamaklarının yazılmasıdır.)

Programlamaya Giriş

Program : Belirli bir problemi çözmek için bir bilgisayar dili kullanılarak yazılmış deyimler dizisi. Önceki bölümde bir problemin çözümü ile ilgili teknikler sunmuştuk. Bir problemi bilgisayar ile çözmek için geliştireceğimiz programın yazımında izleyeceğimiz adımlar:

- i) Problemin ne olduğunu kavra. Çözüm için gereksinimleri belirle.
- ii) Problemin girdilerini, çıktılarını ve diğer kısıtlama ve gereksinimleri belirle (bilgilerin giriş ve çıkış biçimlerinin nasıl olacağına kadar).
- iii) Problemin çözümünü veren algoritmayı yaz.
- iv) Algoritmayı bir programla dili ile yaz.
- v) Programın doğru çalışıp çalışmadığını test et. Bu testi değişik veriler (girdiler) için tekrarla.

Problem Çözme ve Algoritmalar

1.1 Problem Çözme

Problem çözmeye, soruna hemen girişmek yerine, dikkatli ve sistematik yaklaşım ilke olmalıdır. Problem iyice anlaşılmalı ve mümkün olduğu kadar küçük parçalara ayrılmalıdır. Descartes tarafından "Discourse on Method" isimli kitabında anlatılan problem çözme teknikleri;[2]

1. Doğruluğu kesin olarak kanıtlanmadıkça, hiçbir şeyi doğru olarak kabul etmeyin; tahmin ve önyargılardan kaçının.
2. Karşılaştığınız her güçlüğü mümkün olduğu kadar çok parçaya bölün.
3. Düzenli bir biçimde düşünün; anlaşılması en kolay olan şeylerle başlayıp yavaş yavaş daha zor ve karmaşık olanlara doğru ilerleyiniz.
4. Olaya bakışınız çok genel, hazırladığınız ayrıntılı liste ise hiçbir şeyi dışarıda bırakmayacak kadar kusursuz ve eksiksiz olsun.

1.2 Algoritmalar

Belirli bir görevi yerine getiren sonlu sayıdaki işlemler dizisidir. İ.S. 9.yy da İranlı Musaoğlu Horzumlu Mehmet (Alharezmi adını araplar takmıştır) problemlerin çözümü için genel kurallar oluşturdu. Algoritma Alharezmi'nin Latince okunuşu. Her algoritma aşağıdaki kriterleri sağlamalıdır.

1. *Girdi:* Sıfır veya daha fazla değer dışarıdan verilmeli.
2. *Çıktı:* En azından bir değer üretilmeli.
3. *Açıklık:* Her işlem (komut) açık olmalı ve farklı anlamlar içermemeli.
4. *Sonluluk:* Her türlü olasılık için algoritma sonlu adımda bitmeli.

5. *Etkinlik*: Her komut kişinin kalem ve kağıt ile yürütebileceği kadar basit olmalıdır.

Not: Bir program için 4. özellik geçerli değil. İşletim sistemleri gibi program sonsuza dek çalışırlar .

Örnek: İki sayının toplamını yazan algoritmayı yazınız.

Çözüm: Adım 1 --> İlk sayıyı oku, Sembolik olarak A1 S1=?

Adım 2 --> İkinci sayıyı oku A2 S2=?

Adım 3 --> Sayıları topla A3 T S1+S2

Adım 4 --> Sonucu görüntüle. A4 T'yi göster.

1.Soru : Kullanıcının girdiği iki sayının karelerinin toplamını görüntüleyen algoritmayı yazınız.

Çözüm

A0 --> Başla

A1 --> Sayaç=0 (Sayaç'ın ilk sayısı 0 olarak başlar.)

A2 --> Sayı=? : T β T+Sayı (Sayıyı giriniz. T'ye sayıyı ekle ve T'yi göster.)

A3 --> Sayaç= Sayaç+1 (Sayaç'a bir ekle ve sayacı göster.)

A4 --> Sayaç<4 ise A2'ye git. (Eğer sayaç 4'ten küçükse Adım 2'ye git.)

A5 --> O=T/4 (Ortalama için T değerini 4'e böl)

A6 --> O'yu göster. (Ortalamayı göster.)

A7 --> Dur

2.Soru : 20'den 50'ye kadar olan sayıların toplamını bulan algoritma yazınız. BAŞLA

Çözüm A0 --> Başla

A1 --> S=20 : T=0 (Sayı 20 T değeri 0 ile başla)

A2 --> T β T+S (T'ye sayıyı ekle T'yi göster.)

A3 --> S=S+1 (Sayıyı bir artır.)

A4 --> S<50 ise A2'ye git. (Eğer sayı 50'den küçük ise Adım 2'ye git)

A5 --> T'yi göster. (T'nin değerini göster.)

A6 --> Dur Akış Şeması à

3.Soru : 30 ile 90 arasındaki çift sayıların toplamını bulan algoritma yazınız.

Çözüm:

A0 --> Başla

A1 --> S=0 : T=0 (Sayı 0, Toplam değeri 0 olsun)

A2 --> S=S+1 (Sayıyı bir artır, sayıyı göster.)

A3 --> T=T+2S+28 (28 sayısına çift sayı olması için sayıyı 2 ile çarp ve toplama ekle, toplamı göster)

A4 --> 2S+28<90 ise git A2 (Eğer Çift sayı 90 değerinden küçük ise Adım 2'ye git)

A5 --> O=T/S (Toplam değeri sayıya böl ve ortalamayı bul)

A6 --> O'yu yaz. (Ortalamayı göster)

A7 --> Dur

4.Soru : Klavyeden girilen 10 adet sayıdan çift sayıların toplamının tek sayıların toplamına oranını bulan algoritma yazınız.

Çözüm:

A0 --> Başla

A1 --> S=0 : T=0 : Ç=0 (Sayı, tek ve çift sayılar 0 değerini alsın)

A2 --> Sayı? (Sayıyı giriniz)

A3 --> (-1)sayı=1 ise Ç=Ç+Sayı (Eğer -1'in sayı üssü 1 sayısına eşitse sayıyı çift sayıya ekle ve çift sayıyı bul.)

A4 --> (-1)sayı=+1 ise T=T+Sayı (Eğer -1'in sayı üssü +1 sayısına eşitse sayıyı tek sayıya ekle ve tek sayıyı bul.)

A5 --> S=S+1 (Sayaça 1 ekle ve sayacı göster)

A6 --> S<10 ise A2'ye git. (Eğer Sayaç 10'dan küçük ise Adım 2'ye git.)

A7 --> V β Ç/T (Çift sayıların toplamını tek sayıların toplamına böl.)

A8 --> V'yi göster. (Bölme sonucunu göster.)
A9 --> Dur

5.Soru : Klavyeden girilen iki sayıdan en büyüğünü bulup gösteren algoritmayı yazınız.

Çözüm :

A0 --> Başla

A1 --> S1=? : S2=? (İlk sayıyı gir ; ;İkinci sayıyı gir.)

A2 --> S1>S2 ise git A4 (Sayı 1 sayı 2'den küçükse Adım 4'e git.)

A3 --> S2>S1 ise git A5 (Sayı 2 sayı 1'den küçükse Adım 5'e git.)

A4 --> S1'i göster git A6 (sayı 1 değerini göster ve işlemi durdur)

A5 --> S2'yi göster. (Sayı 2 değerini göster)

A6 --> Dur

6.Soru : Klavyeden girilen üç sayıdan en büyüğünü bulup gösteren algoritmayı yazınız.

Çözüm :

A0 --> Başla : EB=-∞ : S=0 (Başla olabilecek EB:en küçük sayı olsun, Sayı 0 olsun)

A1 --> Sayı? (Sayıyı giriniz.)

A2 --> Sayı>EB ise EB=Sayı (Eğer Sayı EB'den küçükse EB sayıyla aynı değeri alsın.)

A3 --> S=S+1 (Sayıya 1 ekle ve sayacı göster)

A4 --> S<3 ise git A1 (Eğer sayac 3'ten küçükse adım 1'e git.)

A5 --> EB'yi göster (En büyük sayıyı göster.)

A6 --> Dur

Örnek 1.2.1 : 1'den 100'e kadar olan sayıların toplamını veren algoritma.

1. Toplam T, sayılar da i diye çağırılın.
2. Başlangıçta T'nin değeri 0 ve i'nin değeri 1 olsun.
3. i'nin değerini T'ye ekle.
4. i'nin değerini 1 arttır.
5. Eğer i'nin değeri 100'den büyük değil ise 3. adıma git.
6. T'nin değerini yaz.

Algoritmaların yazım dili değişik olabilir. Günlük konuşma diline yakın bir dil olabileceği gibi simgelere dayalı da olabilir. Akış şeması eskiden beri kullanıla gelen bir yapıdır. Algoritmayı yazarken farklı anlamlar taşıyan değişik şekildeki kutulardan yararlanılır. Yine aynı amaç için kullanılan programlama diline yakın bir (sözde kod = pseudo code) dil , bu kendimize özgü de olabilir, kullanılabilir.

Aynı algoritmayı aşağıdaki gibi yazabiliriz.

1. T=0 ve i=0
2. i'nin değerini T'ye ekle.
3. i'yi 1 arttır.
4. i<101 ise 2.adıma git.
5. T'nin değerini yaz.

Algoritmayı bir de akış şeması ile gerçekleyelim.

T=0
i=0

i'nin Değrini T'ye ekle

i'yi bir arttır

i<101

T'yi yaz

Örnek 1.2.2 : $ax^2+bx+c=0$ tipi bir denklemin köklerini veren algoritma.

Girdi : a, b ve c katsayıları Çıktı : denklemin kökleri

1. a, b ve c katsayılarını al.
2. $D = b^2-4ac$ değerini hesapla.
3. $D < 0$ ise gerçel kök yok. 7. adıma git.
- 4.
- 5.
6. değerlerini yaz.
7. Dur.

Döngü Gösterimi

Tekrarlanan adımlar

n. Koşul sağlandığı sürece

n.1 ...

n.2 ... tekrarlanan adımlar

n.3 ...

Örnek 1.2.3 : İki tamsayının çarpma işlemini sadece toplama işlemi kullanarak gerçekleyin.

Girdi : iki tamsayı

Çıktı : sayıların çarpımı

1. a ve b sayılarını oku
2. $c = 0$
3. $b > 0$ olduğu sürece tekrarla
- .3.1. $c = c + a$
- 3.2. $b = b - 1$
4. c değerini yaz ve dur

Örnek 1.2.4 : Bir tamsayının faktoriyelini hesaplayınız.

Girdi : Bir tamsayı

Çıktı : sayının faktoriyel

İlgili formül: Faktoriyel(n) = $1 * 2 * \dots * n$

1. n değerini oku
2. $F = 1$
3. $n > 1$ olduğu sürece tekrarla
- .3.1. $F = F * n$
- 3.2. $n = n - 1$
4. F değerini yaz

Örnek 1.2.5 : İki tamsayının bölme işlemini sadece çıkarma işlemi kullanarak gerçekleyin. Bölüm ve kalanın ne olduğu bulunacak.

1. a ve b değerlerini oku
2. $m = 0$
3. $a \geq b$ olduğu sürece tekrarla
- 3.1 $a = a - b$
- 3.2 $m = m + 1$
4. kalan a ve bölüm m 'yi yaz

Örnek 1.2.6 : 100 tane sayıyı okuyup, ortalamasını bul

1. $T = 0, i = 0$
2. $i < 101$ olduğu sürece tekrarla
- 2.1 m değerini oku
- 2.2 $T = T + m$
- 2.3 $i = i + 1$

3. $T = T / 100$
4. Ortalama T 'yi yaz
5. Dur

Örnek 1.2.7 : Bir sınava giren öğrencilerin not ortalamasının hesaplanması

1. Tüm sınav kağıtlarını inceleyip notların toplamını hesapla
2. Ortalamayı notların toplamını incelenen sınav kağıdına bölerek hesapla
3. Ortalamayı yaz.

1. Notların toplamını ve incelenen sınav kağıdı sayısını sıfır kabul et
2. Sıradaki sınav kağıdının notunu notların toplamına ekle
3. İncelenen sınav kağıdı sayısını Bir arttır
4. İncelenecek sınav kağıdı var ise 2. Adıma git
5. Ortalamayı notların toplamını incelenen sınav kağıdına bölerek hesapla
6. Ortalamayı yaz

1. Notların toplamını ve incelenen sınav kağıdı sayısını sıfır kabul et
2. Her bir sınav kağıdı için
 - 2.1. Sıradaki sınav kağıdının notunu notların toplamına ekle
 - 2.2. İncelenen sınav kağıdı sayısını bir arttır
3. Ortalamayı notların toplamını incelenen sınav kağıdına bölerek hesapla
4. Ortalamayı yaz

Koşul Gösterimi

- n. Koşul doğru ise
n.D.1
n.D.2 doğru olduğunda işlenen adımlar
n.D.3
aksi halde
n.Y.1
n.Y.2 yanlış olduğunda işlenen adımlar
n.Y.3

Kök bulma örneğinde 3. Adımı tekrar yazarsak

3. $D \geq 0$ ise
3.D.1
3.D.2
aksi halde
3.Y.1 Reel kök yoktur

Sorular:

- * Girilen üç sayıdan en büyüğünü bulan algoritmayı yazınız.
- * Tamsayılarda üs alma işlemini gerçekleştiren algoritmayı yazınız (ab).
- * 1-100 arasında tutulan bir sayıyı tahmin eden algoritmayı yazınız.

Örnek 1.2.8 : Aracın otopark ücretinin hesaplanması. Araçların en fazla 24 saat kaldığını varsayın.

- 0 - 2 saat 150 bin
2 - 8 saat 300 bin
8-24 saat 500 bin

1. Aracın kaç saat kaldığını öğren (t olsun).
2. $t \leq 2$ ise
 - 2.D.1. ücret = 150 bin

Aksi halde

2.Y.1. $t \leq 8$ ise

2.Y.1.D.1. ücret = 300 bin

Aksi halde

2.Y.1.Y.1. ücret = 500 bin

3. ücreti yaz

4. Dur

Örnek 1.2.9: Sınavdaki en büyük notun bulan algoritma.

1. En büyük = ilk sınav kağıdındaki not (ya da olabilecek en düşük değer kabul edilebilir).

2. İncelenecek sınav kağıdı var ise

2.1 Sınav kağıdındaki not > En büyük ise En büyük = Sınav kağıdındaki not

3. En büyük değerini yaz.

4. Dur

Algoritmanın yazımı daha simgesel olabilir. Ni i. Öğrencinin notu olsun.

1. EB = N1

2. i = 2

3. İncelenecek sınav kağıdı var ise

3.1 $N_i > EB \Rightarrow EB = N_i$

3.2 $i = i + 1$

4. EB' yi yaz.

5. Dur

Örnek 1.2.10 : Programın C dili ile yazılıp çalışır hale getirilmesi.

1. Programı bilgisayara gir

2. Kaynak dosya olarak kaydet

3. Kaynak dosyayı derle (compile)

4. Derleme sonucunda hata var ise

4.1 Hataları düzelt

4.2 3. Adıma git

5. Oluşan amaç dosyasına diğer dosyaları bağla (link)

6. Bağlama sonucunda hata var ise

6.1. Hataları düzelt

6.2. Hatalar kaynak dosya ile ilgili ise 2. adıma aksi halde 5. adıma git

7. Program çalıştırılmaya hazır