

ALGORİTMA VE AKIŞ ŞEMALARI

İster bilgisayarla ister bilgisayarsız soru çözmek için belirli bir yol vardır. Ancak bu yol ile sağlıklı bir çözüme ulaşılabilir. Bilgisayar kullanarak soru çözmek için sonuca giden yolun tam olarak belirlenmesi gerekir. Doğru bir yol izleyebilmek için, çıkılan ve ulaşılan yer tanımlanmalıdır. Aynı soru için değişik çözüm yolları geliştirilebilir. Eğer bilgisayara verilen çözüm yanlışsa, çıkan sonuç yanlış çözüm doğru ise çıkan sonuç da doğrudur.

Soru Çözme Adımları :

Bilgisayar ortamında bir problem çözülürken aşağıdaki adımlara dikkat edilmelidir.

a-) Soru Tanımlama: Her şeyden önce çözülecek soru tam olarak anlaşılmalıdır. Yanlış anlaşılmış bir sorunun çözümü yanlış olacak ve istenileni vermeyecektir. Bu adımda yapılacak en ufak bir hata daha sonraki adımların yeni baştan yapılmasını gerektirebilir. Sorunun tanımı yapılırken var olan bilgiler, anlamları ve birbirleri ile ilişkileri tanımlanmalıdır. Daha sonra istenenler belirlenmeli ve bunların var olan bilgiler ile ilişkileri öğrenilmelidir. Son olarak yapılacak işlemler belirlenir. Mümkün ise örnek veriler ile elde edilen sonuçlar değerlendirilmelidir.

b-) Algoritma Geliştirme: Algoritma bir sorunun çözümü için izlenecek yolun tanımıdır. Kısaca algoritma mevcut bilgilerden istenilenlere erişme yöntemidir. Soru tanımını tam olarak yaptıktan sonra, çözüm için yol aramak gerekir. Genellikle bir sorunun birden fazla çözüm yolu olabilir. Bunlardan en uygunu seçilmeye çalışılır. Soru ne kadar karışık olursa olsun, alt birimlere bölünür. Her birimin çözümü ayrı, ayrı yapılır. Bu yapılırken birimler arası ilişki sürekli olarak korunur.

c-) Girdi ve Çıktı Biçimi Belirleme: Sonuçların dış ortama, dolayısıyla insana aktarımı düzgün bir biçimde yapılmalıdır. Programcı program çıktısı olarak almak istediği dökümün biçimini tasarlar. Bir döküm biçimi tasarlanırken anlaşılır ve kullanılabilir olmasına özen gösterilmelidir. Genellikle programa, çözdüğü soruna ilişkin bazı verilerin dışarıdan verilmesi gerekir. Örneğin bir denklem takımının kökleri bulunacaksa, ilgili katsayıların programa verilmesi gibi.

d-) Akış Şemasını Çizme: Akış şeması belirli bir işin yapılabilmesi için, basit işlemlerle şema halinde gösterilmesidir. Kısaca algoritmanın şemalarla gösterilmesidir. Algoritma geliştirildikten sonra, daha iyi anlaşılabilir olması ve programlama dillerine aktarımı daha kolay olması nedeniyle, akış şeması haline getirilir. Böylece sorunun çözüm basamakları, birbirleri ile ilişkileri ve bilgi akışı daha kolay görülebilir ve yanlışlıklar düzeltilebilir.

e-) Kodlama: Akış şemaları çizildikten sonra, sorunu yapısına uygun bir programlama dili seçilir. Bu dil ile akış şemaları dilin kurallarına uygun olarak bilgisayarın anlayabileceği duruma getirilir.

f-) Programı Sınama: Program yazıldıktan sonra, sonuçları daha önceden bilinen veriler girilerek, elde edilen sonuçlarla çıkan sonuçlar karşılaştırılır. Programın doğru çalışıp çalışmadığı sınanır.

Şimdi algoritma hazırlama işlemini daha ayrıntılı olarak inceleyelim. **Cünkü programlamanın en önemli kısmı algoritma hazırlayabilmektir. Algoritma hazırlandıktan sonra hazırlanan algoritmanın herhangi bir programlama dilinde kodlanması işin en basit kısmıdır. Bu yüzden kullanılan dilin basic ya da başka bir programlama dili olması hiçbir şeyi değiştirmez. Yani burada önemli olan programlama dili değil problemin çözümü için algoritma geliştirebilmektir. Bu yüzden kullanılan programlama dilinin eski yada yeni bir programlama dili olması hiç önemli değildir.**

Örnek bir Algoritma

Örneğimiz bir insanın evden çıkıp işe giderken izleyeceği yolu ve işyerine girişinde ilk yapacaklarını tanımlamaktadır.

Çözüm 1:

Evden dışarıya çık
 Otobüs durağına yürü
 Durakta gideceğin yöndeki otobüsü bekle
 Otobüsün geldiğinde otobüse bin
 Biletini bilet kumbarasına at
 İneceğin yere yakınlaştığında arkaya yürü
 İneceğini belirten ikaz lambasına bas
 Otobüs durunca in
 İşyerine doğru yürü
 İş yeri giriş kapısından içeriye gir
 Mesai arkadaşlarınla selamlaş
 İş giysini giy
 İşini yapmaya başla.

1. Algoritma Hazırlanması:

Algoritmanın bir sorunun çözümü için izlenecek yolun tanımı olduğunu belirtmiştik. Bu yolun açıklanabilmesi için, algoritmada kullanılan bazı tanım ve kurallar vardır. Şimdi bu tanım ve kuralları inceleyelim.

1.1. Değişken Kavramı:

Farklı zamanlarda farklı değerler alabilen bilgi sahalarına verilen sembolik adlardır. Bilgisayar işlem yaparken RAM belleği(geçici bellek) kullanır. İşte program yazılırken programcının Ram belleği kullanmasını sağlayan değişkenlerdir. Değişkenler Ram bellekte tahsis edilmiş odacıklar olarak düşünülebilir. Yani bir değişken tanımlandığında ram bellekte bir odacık (bir bölüm) açılır ve bu bölüme değişken ismiyle ulaşılır. Program içinde kullanılacak olan değişkenler problemin tanımı ve girdi-çıkıtlı belirleme aşamalarında belirlenmelidir.

Örneğin klavyeden girilen iki sayının toplamını bulan program yazılırken 3 tane değişken tanımlanmalıdır. Çünkü klavyeden 2 tane sayı girilecek ve bu sayılar toplanarak 3. bir değişkene aktarılacaktır.

Soru: Klavyeden girilen 3 sayının aritmetik ortalaması bulunurken kaç değişken tanımlanmalıdır.

1.2. Aktarma Deyimi:

Aktarma deyimi yada operatörü değişkenlere değer aktarmak için kullanılır. **A=5** yada **A=A+1** şeklindeki bir yazılımda “=” sembolü aktarma deyimi adını alır. Aktarma deyiminin sağ tarafındaki değer yada matematiksel ifadenin sonucu, sol tarafındaki değişkene aktarılır.

Aktarma yapılırken değişkenin aldığı bir önceki değer kaybolur. Bu işlem matematiksel mantıkla karıştırılmamalıdır. Matematikte $A=A+1$ yanlış olduğu halde, bilgisayar mantığında doğrudur.

Sayı1=9

Sayı2=6

Toplam=Sayı1+Sayı2

Toplam=Toplam*2

Yandaki işlemlerin sonucunda bellekteki değişkenlerin değerleri şu şekilde değişir.		
Sayı1	Sayı2	Toplam
9	6	15 30
RAM BELLEK		

Soru 1:

Sayı1=1

Sayı2=1

Sayı3=Sayı1+Sayı2

Sayı4=Sayı3+Sayı2

Sayı5=Sayı4+Sayı3

Yukarıdaki aktarma işlemlerinin sonucunda değişkenlerin değişimini şema halinde gösteriniz.

Soru2:

Vize=60

Final=70

Toplam=(Vize*40/100)+(Final*60/100)

Ortalama=Toplam/2

Yukarıdaki aktarma işlemlerinin sonucunda değişkenlerin değişimini şema halinde gösteriniz.

1.3. Matematiksel Mantık ve Karar Sembolleri:

Algoritmada kullanılan karar sembolleri aşağıdaki tabloda belirtilmiştir.

-	+	*	/	=
Çıkarma	Toplama	Çarpma	Bölme	Aktarma
<>	<	>	<=	>=
Eşit değil	Küçüktür	Büyüktür	Küçük eşit	Büyük eşit

Örnekler:

Dyili=1977

Yas=2003-1977

1.Sayı=-5

2.Eğer Sayı<0 ise Yaz “Sayı Negatif”

3.Eğer Sayı>0 ise Yaz “Sayı Pozitif”

4.Eğer Sayı=0 ise Yaz “Sayı Sıfırdır”

Yandaki örnekte Sayı değişkenine -5 değeri aktarılmıştır. Diğer satırlarda ise Sayı değişkenin içeriğine bakılarak bir mesaj verilmektedir. Bu örnekte yalnızca 2. satırdaki şart sağlanır ve ekrana “Sayı Negatiftir” mesajı yazılır.

Bir veya birden fazla koşul bazı bağlaçlarla bir araya getirilerek daha karmaşık sorular sorulabilir. Örneğin yas (kişinin yaşı) değişkeninin içeriği kontrol edilmek istendiğinde şöyle bir şart cümlesi kullanılabilir:

Yas=-5

Eğer yas<0 Veya Yas=0 ise Yaz “Yanlış Değer Girildi”

Algoritmada kullanılabilecek bağlaçlar VE, VEYA, DEĞİL bağlaçlarıdır.

VE Bağlacı: Ve bağlacı ile söylenmek istenen her iki koşulun da sağlanmasıdır. VE bağlacı ile bağlanmış önermelerden en az birinin yanlış olması sonucu yanlış yapar.

VEYA Bağlacı: VEYA bağlacı ile bağlanan koşullardan en az birisi doğru ise sonuç doğrudur. İki den fazla önermeler için, önermelerden en az birinin doğru olması sonucu doğru yapar.

DEĞİL Bağlacı: DEĞİL bağlacı doğruyu yanlış, yanlış doğru yapar. DEĞİL tek bir önerme veya koşul üzerinde uygulanır. VE, VEYA ise iki önerme veya koşul üzerinde uygulanır.

Doğru=1 ve **Yanlış=0** tanımıyla, aşağıdaki tabloda bağlaçların x ve y'nin alacağı değerlere göre sonuçları gösterilmiştir.

X	Y	X VE Y	X VEYA Z	Z	DEĞİL Z
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1		
1	1	1	1		

Örnek:

DEĞİL A=1 VE B=C VEYA (D=2 VE ADI="AHMET") önermesini A=5, B=4, C=4, D=3 ve ADI="AHMET" için bulalım.

DEĞİL A=1 VE B=C VEYA (D=2 VE ADI="AHMET")
 1 VE 1 VEYA (0 VE 1)
 1 VEYA 0
 0

1.4. Matematiksel İşlemler:

Matematiksel işlemleri algoritmada aynen kullanamayız. Bilgisayar mantığına göre matematiksel ifadelerin yeniden yazılmaları gerekir. Algoritmada işlem öncelik sırası kuralları aşağıda verilmiştir. Parantez kullanılarak işlem öncelik sıraları değiştirilir. İç içe kullanılan parantezlerde öncelik en içtekindedir. Aynı işlem önceliğine sahip elemanlarda işlem soldan sağa doğrudur.

İşlem öncelik sırası kuralları			
Sıra	Tanım	Matematik	Bilgisayar
1	Parantezler	(())	(())
2	Üs Almak	a ⁿ	a^n
3	Çarpma ve Bölme	ab a/b	a*b a/b
4	Toplama ve Çıkarma	a+b a-b	a+b a-b


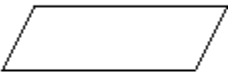




Örnek:

$$(C*D/(A*D))+B+C*D/A \longrightarrow \frac{CD}{AD} + B + \frac{CD}{A}$$

$$C+B*A/A-B^C*(B-C)^B \longrightarrow C + \frac{BA}{A} - B^C (B - C)^B$$

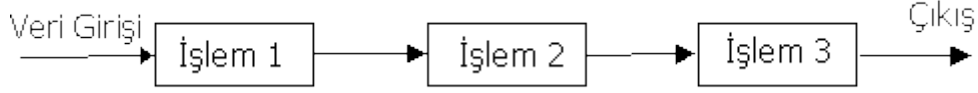
2. Akış Şeması Hazırlama:Geliştirilecek olan yazılımın genel yapısının şematik gösterimine akış şeması veya blok diyagramı adı verilir. Akış diyagramları, yazılımı oluşturacak program parçalarını ve bu parçaların birbirleri ile olan ilişkilerini belirler. ***Bir bilgisayar programının oluşturulmasında akış diyagramlarının hazırlanması, algoritma oluşturma aşamasından sonra gelmektedir.*** Bilgisayar programının oluşturulması sırasında algoritma aşaması atlanarak, doğrudan akış diyagramlarının hazırlanmasına başlanabilir. Programlama tekniğinde önemli ölçüde yol almış kişiler bu aşamayı da atlayarak direkt olarak programın yazımına geçebilirler.Akış şemalarının algoritmadan farkı, adımların simgeler şeklinde kutular içinde yazılmış olması ve adımlar arasındaki ilişkilerin (iş akışı) oklar ile gösterilmesidir.Akış şemalarında kullanılan semboller, anlamları ve kullanım amaçları aşağıdaki tabloda verilmiştir.

Tablo 1. İş akış şemalarında kullanılan semboller ve anlamları

Simge	Simgenin Adı ve Anlamı
	Elips Akış şemasının başlangıç ve bitiş yerlerini gösterir. Başlangıç simgesinden çıkış oku vardır. Bitiş simgesinde giriş oku vardır.
	Paralel Kenar: Programa veri girişi ve programdan elde edilen sonuçların çıkış işlemlerini gösterir.(Oku, Yaz)
	Dikdörtgen Aritmetik işlemler ve değişik atama işlemlerinin temsil edilmesi için kullanılır.(A=A+1, Final=100 vb..)
	Eşkenar Dörtgen Bir karar verme işlemi temsil eder. (Eğer sayı<0 ise Yaz "Sayı negatif" vb...)
	Altıgen Program içinde belirli blokların ard arda tekrar edileceğini gösterir.(Döngü kurmak için kullanılır)
	Oklar Diyagramın akış yönünü ,yani her hangi bir adımdaki işlem tamamlandıktan sonra hangi adıma gidileceğini gösterir.

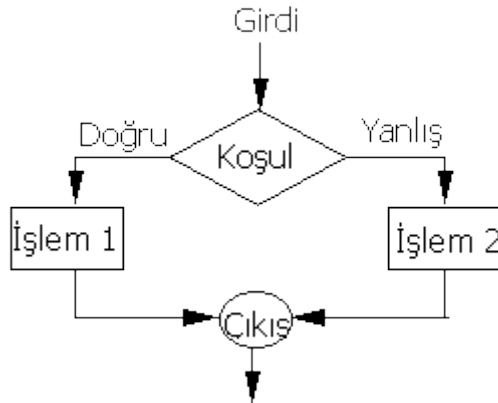
Ayrıntılı bir akış şeması, yazılımı oluşturan işlemleri ve ilişkilerini en küçük detayına kadar belirler.

Bir bilgisayar programının geliştirilmesinde kullanılan programlama dili ne olursa olsun bu programların akış diyagramlarında genel olarak yalnız üç basit mantıksal yapı kullanılır. Bu mantıksal yapılardan en basiti sıralı yapıdır. Sıralı yapı, hazırlanacak programdaki her işlemin mantık sırasına göre nerede yer alması gerektiğini vurgular. Bu yapı sona erinceye kadar ikinci bir işlem başlayamaz.



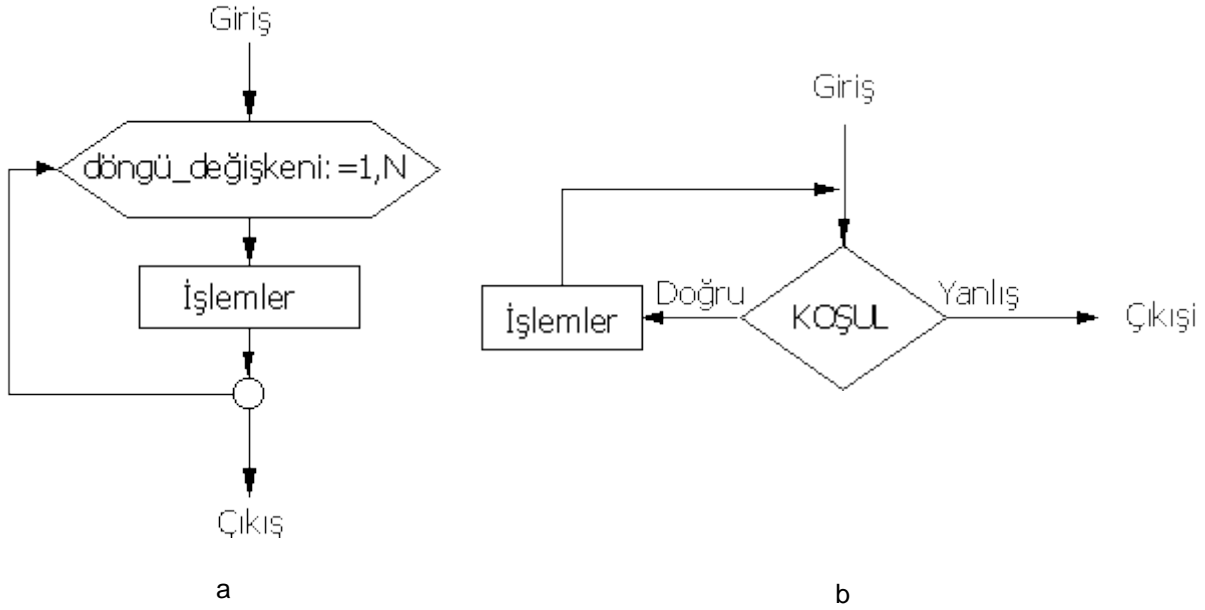
Şekil 1.2 Sıralı Yapı

Mantıksal yapılardan ikincisi Karar Verme yapısıdır (Şekil 1.3). Programlama sırasında If...Then... Else (Eğer.... <şart>İse....)yapısı ile tanıyacağımız bu mantıksal yapılar, birden fazla sıralı yapı seçeneğini kapsayan modüllerde, hangi şartlarda hangi sıralı yapının seçileceğini belirler.



Şekil 1.3 Karar Verme Yapısı

Üçüncü mantıksal yapı çeşidini tekrarlı yapılar oluşturmaktadır. Yani döngü oluşturmak için kullanılan yapıdır. Döngüler aynı işlemin bir çok kez yapılmasını sağlar.Söz konusu üç değişik yapı, değişik kombinezonlarda kullanılarak istenilen işlevleri yerine getirecek programlar hazırlanabilir. Programların bu üç basit yapı ile sınırlandırılması program modüllerinin daha kolay tasarlanmasını sağlar.



Şekil 1.4. Tekrarlı Yapı

ÖRNEK ALGORİTMA VE AKIŞ ŞEMALARI

Örnek 1: Klavyeden girilen 2 sayının toplamını bulan programın algoritma ve akış şemasını yapınız.

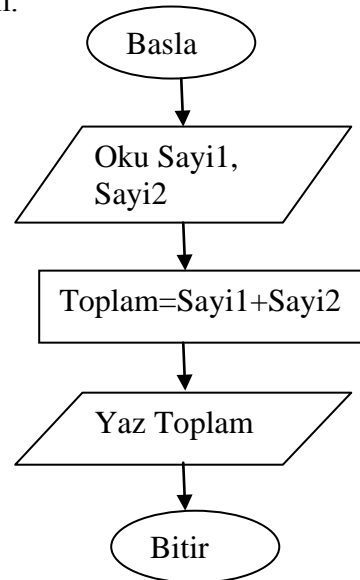
Bu problemi çözerken yapmamız gereken ilk iş problemi iyice anlamaktır. Problemi çözümlersek yapılacak olan işlemler şunlardır:

1. Klavyeden 2 adet sayı girilecek. O zaman bellekte 2 odacık açmalıyız. Yani veri girişi için 2 tane değişken kullanmak zorundayız.
2. Klavyeden girilen ve ram belleğe aktarılan bu iki değişken toplanacak. Toplam sonucunu yine ram belleğe aktarılmalıdır. Bu yüzden ram bellekte bir odacık daha açmalıyız.
3. Bulduğumuz toplam sonucunu ekrana yazdırmalıyız. Aksi takdirde toplam sonucu sadece ram bellekte bulunur ve sonucu kullanıcı göremez. Şimdi bu açıklamalar ışığında algoritma ve akış şemasını yazalım:

1. Oku Sayi1, Sayi2
2. Toplam=Sayi1+Sayi2
3. Yaz Toplam
4. Dur

9 ve 6 değerleri için bellekteki durum şöyledir:

Sayi1	Sayi2	Toplam
9	6	15



Örnek-2:

Klavyeden girilen 3 sayının aritmetik ortalamasını bulan programın algoritma ve akış şemasını yazın.

Değişkenler S1(1. sayı), S2, S3, Toplam, Ort (Ortalama) olmalıdır. S1, S2 ve S3 değişkenlerinin değerleri klavyeden okutulacaktır. Toplam ve Ort değişkenleri ise program içinde hesaplatılacaktır. Sonuç olarak ekranda Ort değişkeninin içeriği görüntülenecektir.

1. Oku S1, S2 ve S3

2. $Toplam = S1 + S2 + S3$

3. $Ort = Toplam / 3$

4. Yaz "Ortalama="; Ort

5. Dur

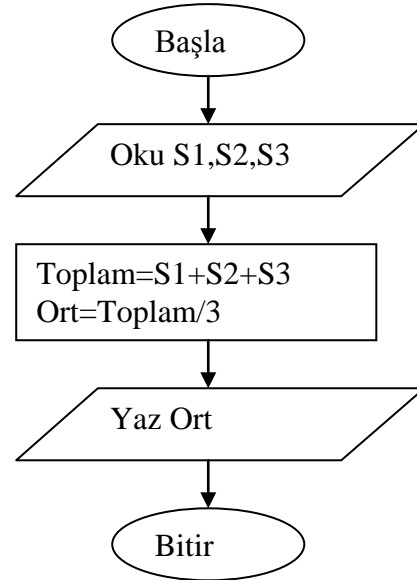
Yerine
 $Ort = (S1 + S2 + S3) / 3$
yazılabilir.

10, 15, 8 değerleri için bellekteki durum ve ekran çıktısı:

S1	S2	S3	Toplam	Ort
10	15	8	33	11

Ekran Çıktısı

Ortalama=11



Örnek 3: İsim ve soyadınızı ekrana 5 defa yazdıran programın algoritma ve akış şemasını yazın?

1. Basla

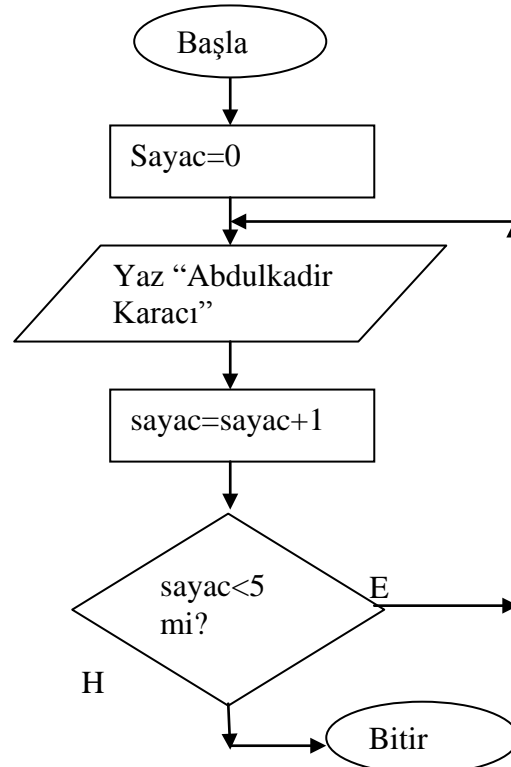
2. sayac=0

3. YAZ "Abdulkadir KARACI", sayac

4. sayac=sayac+1

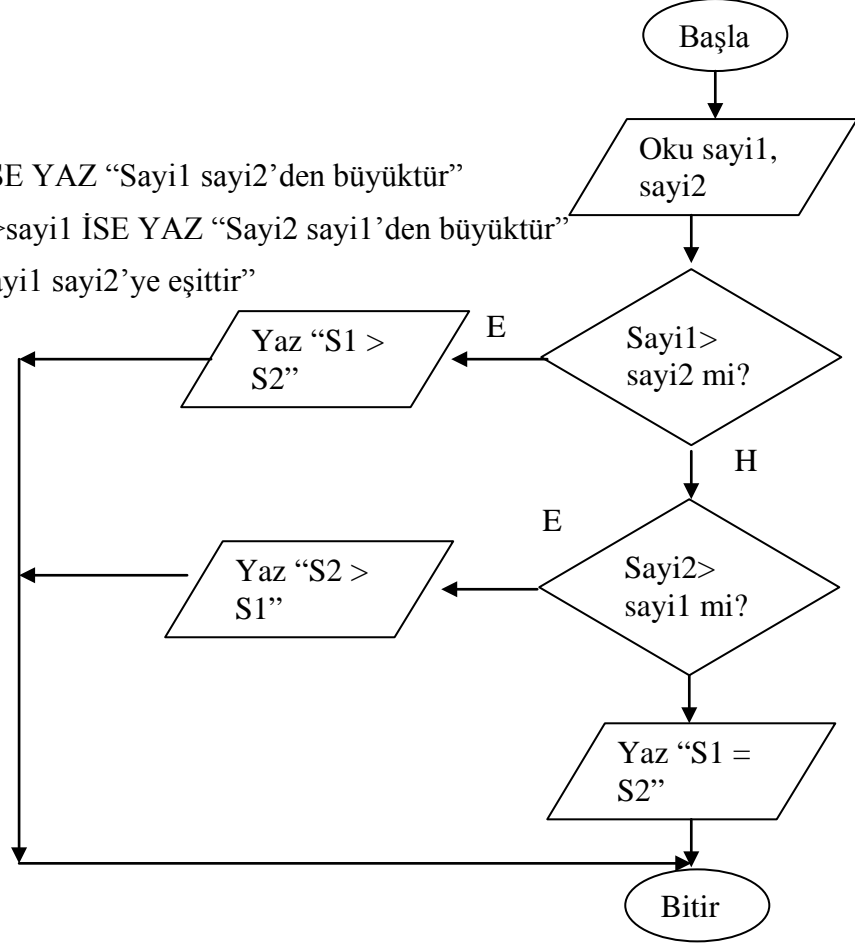
5. Eğer sayac<5 GİT 3

6. DUR



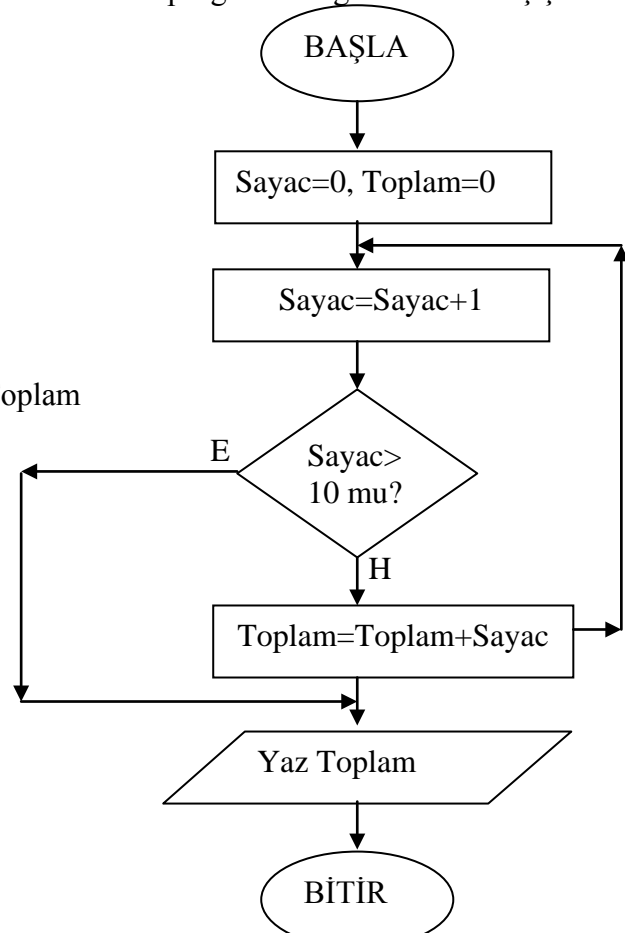
Örnek 4: Klavyeden girilen 2 sayıyı karşılaştırıp sonucu ekrana yazdıran algoritma ve akış şemasını yazın?

1. BAŞLA
2. OKU sayı1,sayı2
3. EĞER sayı1>sayı2 İSE YAZ “Sayı1 sayı2’den büyüktür”
4. Değilse EĞER sayı2>sayı1 İSE YAZ “Sayı2 sayı1’den büyüktür”
5. DEĞİL İSE YAZ “Sayı1 sayı2’ye eşittir”
6. BİTİR



Örnek 5: 1-10 arasındaki tamsayıların toplamını bulan programın algoritma ve akış şemasını yazın?

1. BAŞLA
2. Sayac=0, Toplam=0
3. Sayac=Sayac+1
4. EĞER Sayac>10 İSE GİT 7
5. Toplam=Toplam+Sayac
6. GİT 3
7. YAZ “1-10 Arası Sayıların Toplamı=”,Toplam
8. BİTİR



Örnek 6: 1-100 arasındaki çift sayıların toplamını bulan programın algoritmasını ve yazın?

1. BAŞLA
2. Sayac=2, Toplam=0
3. Toplam=Toplam+Sayac
4. Sayac=Sayac+2
5. EĞER Sayac<=100 İSE GİT 3
6. YAZ “1-100 Arası Çift Sayıların Toplamı=”, Toplam
7. BİTİR

Örnek 7: Klavyeden girilen 10 tabanındaki sayıyı ikilik tabana çeviren programın algoritmasını ve akış şemasını yazın.

1. Basla
2. Oku Sayi
3. Bolum=Sayi tamböl 2
4. Kalan=Sayi-Bolum*2
5. Yaz kalan
6. Sayi=Bolum
7. Eger Sayi>=2 İse GİT 3
8. Yaz Sayi
9. Bitir

