DERS İÇERİĞİ

- Programlamaya giriş ve algoritma kavramları
- Basit ve karmaşık veri tipleri
- Program kontrol komutları (Döngü ve şart yapıları)
- Diziler ve karakterler
- Pointerler
- Fonksiyonlar
- Yapılar
- Ekran ve dosya giriş/çıkış işlemleri
- Matematik ve zaman fonksiyonları

Programlama dilleri

Programlama dili: İnsan-makina ve makina- makina arasındaki iletişimi sağlar.

- Programlama dili programcının programı yazarken kullandığı özel bir dildir.
- Programcının bilgisayara,
  - hangi veri üzerinde işlem yapacağını,
  - verinin nasıl depolanıp iletileceğini,
  - hangi koşullarda hangi işlemlerin yapılacağını tam olarak anlatmasını sağlar.

İlk Bilgisayar Programcısı

- ***Ada Augusto Lovelace*** (1815-1852)
- Analitik makinanın kullanımını sağlayan ilk bilgisayar programını yazmıştır.
- Ada Programlama diline (1970-) bu isim onu onurlandırmak için verilmiştir.

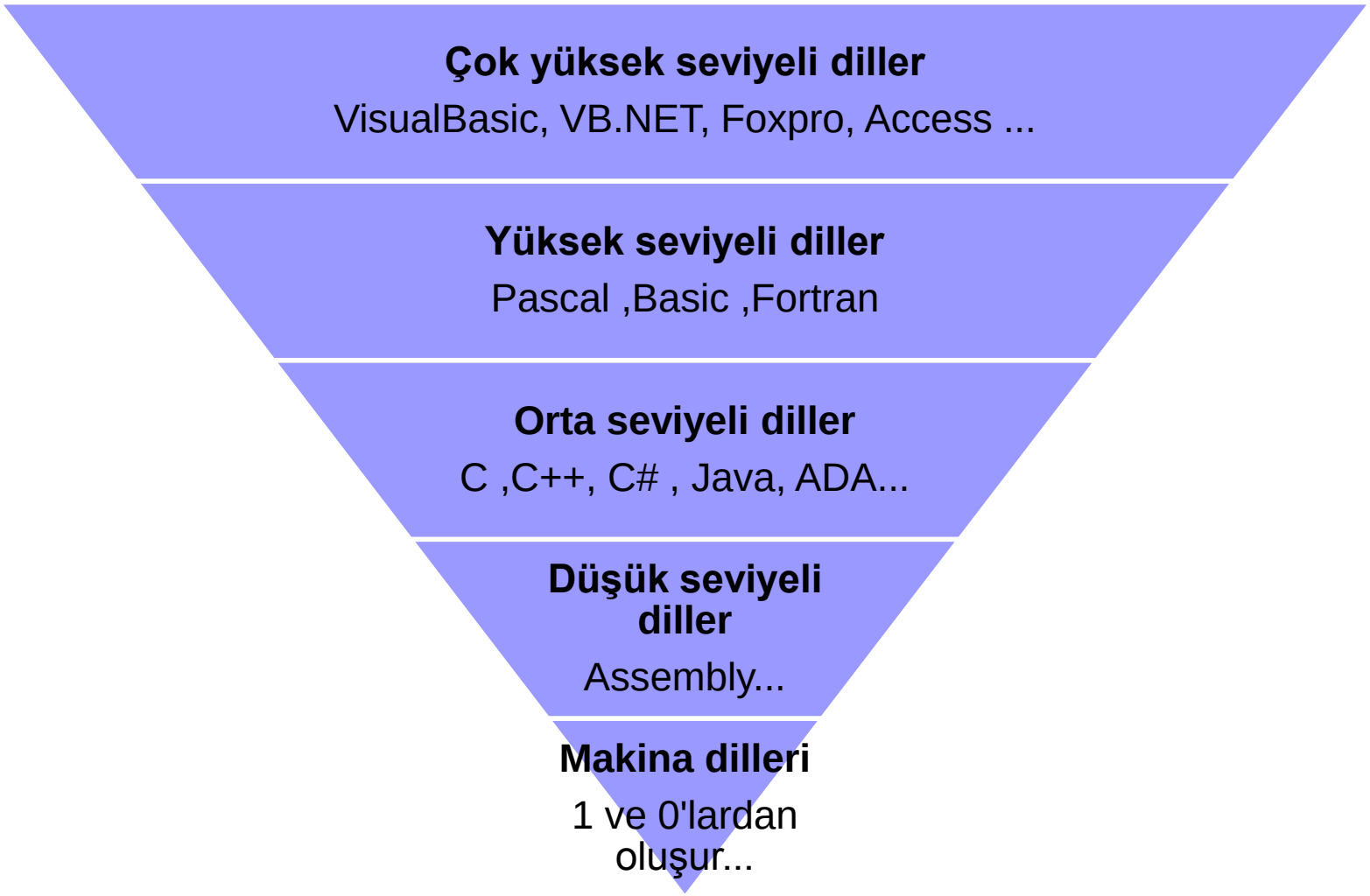
PROGRAMLAMA DİLLERİNİN SINIFLANDIRILMASI

1. Seviyelerine göre
2. Çalıştıkları ortama göre

- **Seviyelerine göre**

Makina kodlarına yakın diller *düşük seviyeli,* insanların kolay anlayıp kullanabileceği diller ise *yüksek seviyeli* programlama dilleridir

PROGRAMLAMA DİLLERİNİN SINIFLANDIRILMASI



- Makine Dili
- Makine dili bilgisayarın doğal dilidir ve bilgisayarın donanımsal tasarımına bağlıdır.
- Makine dilinde yazılan kodlar doğrudan makinanın işlemcisine, donanım parçalarına verilen komutlardır.
- Bilgisayarların geliştirilmesiyle birlikte onlara iş yaptırmak için kullanılan ilk diller de makine dilleri olmuştur.
- Bu yüzden makine dillerine 1. kuşak diller de denebilir.

- Sembolik Makine Dili
- Ardından sembolik makine dilleri geliştirilmiştir.
- Sembolik makine dilleri (Assembly languages) yalnızca 1 ve 0 dan oluşan makine dilleri yerine İngilizce bazı kısaltma sözcüklerden oluşuyordu.

- Yorumlayıcı
- Bu dillerle yazılan bir programın çalıştırılma aşamasında yorumlayıcı (interpreter) bir program yardımıyla sembolik dilin komutları, bilgisayar tarafından komut komut makine diline çevriliyor ve oluşan makine kodu çalıştırılıyordu.
- Ancak bu şekilde çalıştırılan programların hızı neredeyse 30 kat yavaşlıyordu.

- Derleyici
- Programların yavaşlamasını azaltmak için bir fikir ortaya atıldı
- Program her çalıştırılışında değil sadece ilk çalıştırılışında makina diline çevrilsin, sonra öyle kaydedilsin, böylece bilgisayar yavaşlamasın;
- Böylece insanın anlayabileceği basit bir algoritmik dili, makinanın anlayabileceği dile çeviren bir program yazmış ve bu programa derleyici denmiştir.

- Derleyici
- Bu fikiri geliştiren Grace Hopper isimli bir bayandır.
- *Grace Hopper* aynı zamanda Cobol dilini geliştiren ekipten biridir ve bug sözcüğünü ilk olarak Grace Hopper kullanmıştır.

- Böylece programcılar sembolik sözcüklerden oluşan Assembly programlarını kullanıyor,
- yazdıkları programlar derleyici tarafından makine koduna dönüştürülüyor
- ve makine kodu eski hızından birşey kaybetmeksizin tam hızla çalışıyordu.
- Assembly diller 2. kuşak diller olarak tarihte yerini aldı.

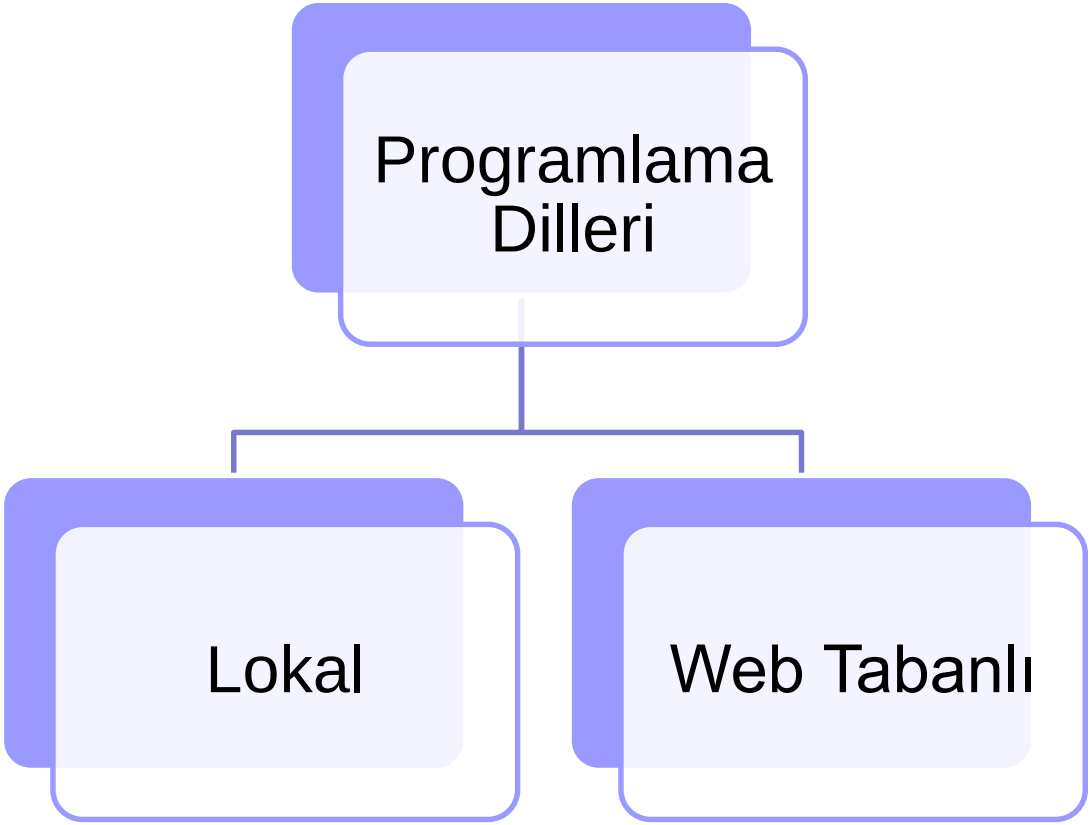
- Ancak en basit işlemlerin bile bilgisayara yaptırılması için bir çok komut gerekmesi,
- programlama sürecini daha hızlı bir hale getirmek için arayışları başlatmış,
- bunun sonucunda da daha yüksek seviyeli programlama dilleri geliştirilmeye başlanmıştır

- Tarihsel süreç içinde Assembly dillerinden daha sonra geliştirilmiş ve daha yüksek seviyeli diller 3. kuşak diller sayılmaktadır.
- Bu dillerin hepsi algoritmik dillerdir.
- Bugüne kadar geliştirilmiş olan yüzlerce yüksek seviyeli programlama dilinden yalnızca pek azı bugüne kadar varlıklarını sürdürebilmiştir:

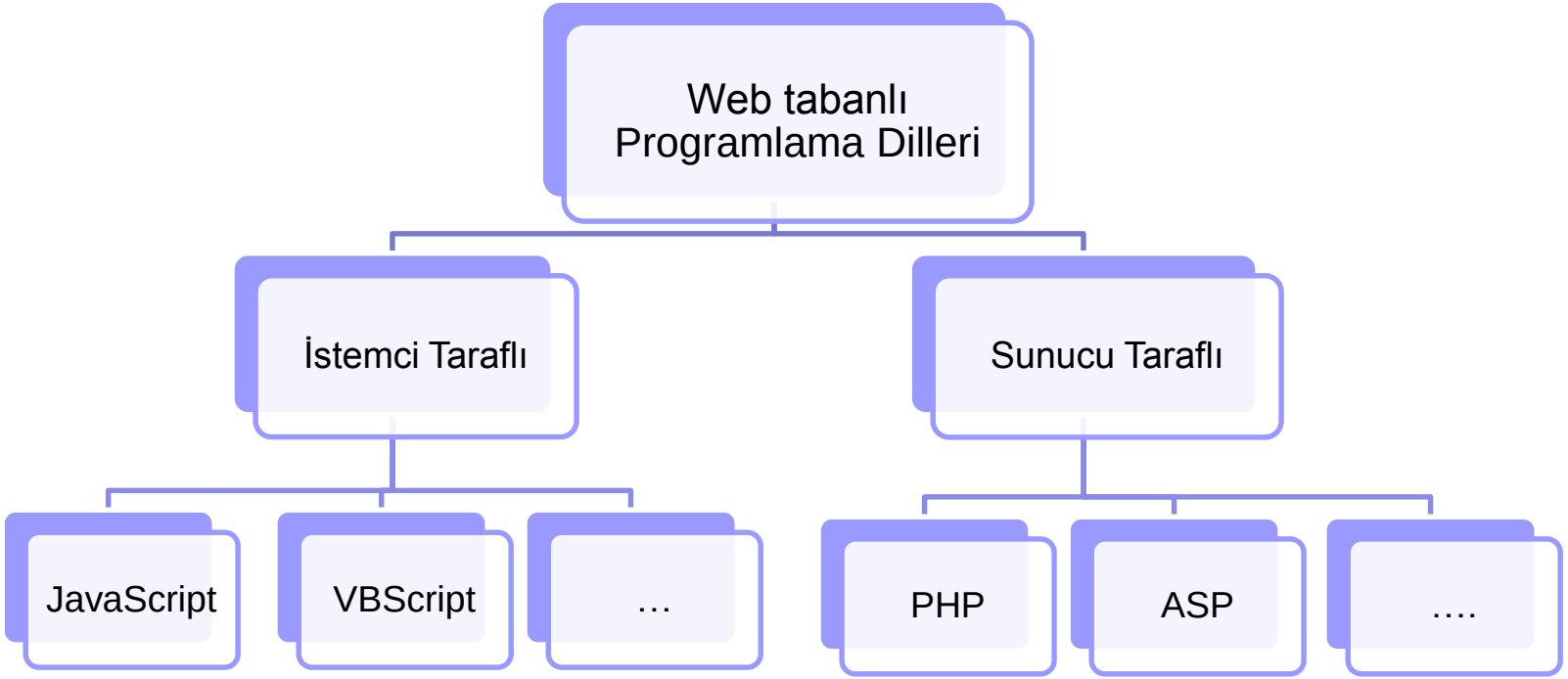
- Çok yüksek seviyeli ve genellikle algoritmik yapı içermeyen programların görsel bir ortamda yazıldığı diller ise 4. kuşak diller olarak isimlendirilirler.
- Özellikle küçük IBM makinalarının kullanıcıları olan şirketlerin, rapor üretimi için basit bir dil istemeleri üzerine IBM firması tarafından geliştirilmiştir.

PROGRAMLAMA DİLLERİNİN SINIFLANDIRILMASI

Çalıştıkları ortama göre


Programlama Dilleri
Lokal
Web Tabanlı

- Lokal programlama dilleri, bilgisayara yükleyerek exe'si ile çalıştırdığımız masaüstü uygulamalarını geliştirmeye imkan verir,
- Web tabanlı programlama dilleri ise istemci-sunucu mimarisine göre tasarlanan web sayfalarını geliştirmeye imkan verir,


Web tabanlı
Programlama Dilleri
İstemci Taraflı
Sunucu Taraflı
JavaScript
VBScript
…
PHP
ASP
….

WEB TABANLI PROGRAMLAMA DİLLERİ

- İstemci Taraflı Programlama dilleri, HTML dilinin karşılayamadığı bazı ihtiyaçlara çözüm üretmek için kullanılır.
- İstemci Taraflı Programlama dilleri, kullanıcı ile veri alış verişi içerisindedir.
- Her işlem istemci üzerinde gerçekleştirilir.
- Script dilleri

- Script dilleri sayfa ile kullanıcının etkileşimli olarak çalışmasını sağlar:
  - bir nesneye tıklamak,
  - bir nesnenin üzerine gelmek,
  - bir nesnenin üzerinde dolaşmak gibi

- Sunucu Taraflı Programlama dilleri, sunucu ile veri alış verişi içerisindedir.
- Her işlem sunucu üzerinde gerçekleştirilir.
- Örneğin bir dosya yüklersin, sunucudaki dosyayı düzenlersin.

- Webde istemci –sunucu iletişimi
- İstemci bilgisayarda internet explorer veya firefox gibi bir web tarayıcısına bir adres girilir,
- Tarayıcı ilgili web sunucusunu bulur ve sayfayı ister,
- Web sunucusu ilgili sayfayı komutlar topluluğu şeklinde istemci makineye gönderir,
- İstemci makinedeki tarayıcı bu sayfaları alır, gelen komutları yorumlar ve web sayfasını anlaşılır bir şekilde gösterir

Programlama dillerinin uygulama alanlarına göre sınıflandırılması:

. **Bilimsel ve Mühendislik Diller**: Bu diller daha çok bilimsel ve mühendislik problemlerinin çözümünde tercih edilirler. PASCAL ve C dillerini , birde geleceği pek parlak olmayan ve hala ısrarla kullanılan 90 canlı, dünyanın ilk yüksek seviyeli dili FORTRAN' ı buna örnek verebiliriz.

. **Veritabanı Programlama Dilleri:** Bu diller veritabanlarının genel olarak yönetiminde kullanılan dillerdir: DBASE, PARADOX, FOXPRO, SQL.. kişisel bilgisayarlarda yaygın olarak kullanılanlardan bazıları.

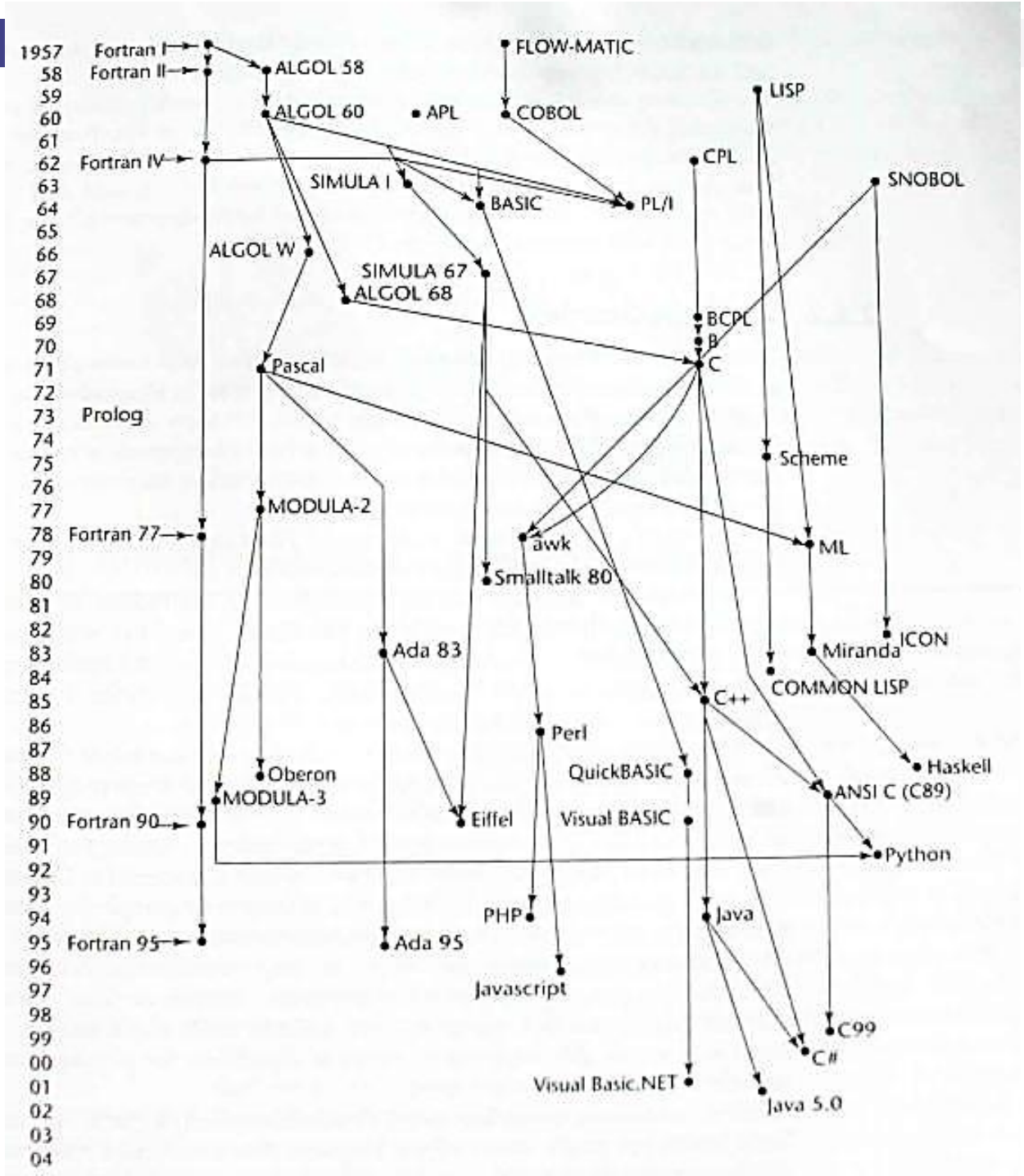
. **Yapay Zeka Dilleri:** bu diller insan davranışını taklit etmeye yönelik yapay zeka içeren programların yazımında kullanılan mantıksal dillerdir. En ünlüleri : LISP ve PROLOG.

. **Genel Amaçlı Diller:** Çok çeşitli konularda uygulama geliştirmek amacıyla kullanılan dillerdir. C ve PASCAL' ı örnek verebiliriz.

. **Sistem Programlama Dilleri:** Sistem programlarının yazımında kullanılan dillerdir. C ' yi sembolik makine dillerini bu grup içinde ele alabiliriz.

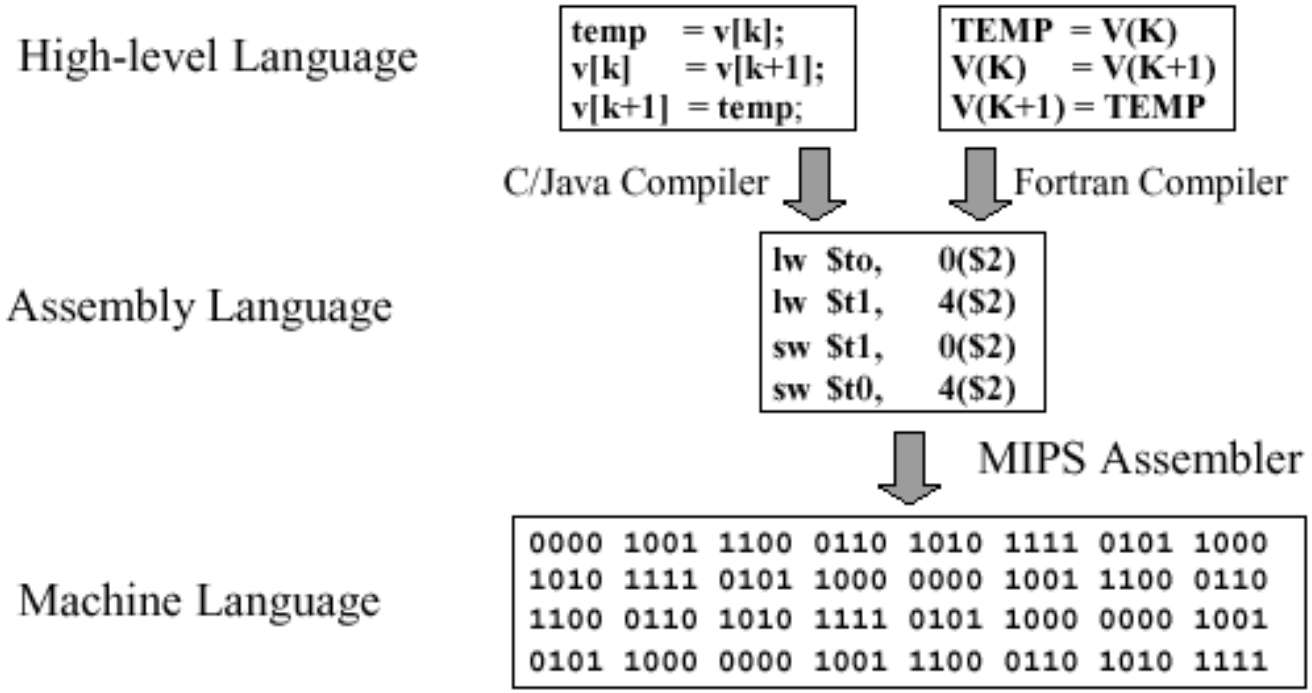
Genel olarak bugüne kadar kullanılan ***programlama dilleri*** şunlardır:

- Basic
- Fortran
- Pl/1
- Rpg
- Cobol
- Modular2,3
- Turbo C++
- Pascal
- Cecil
- Ada
- Blue
- Dylan
- Eiffel
- Erlang
- Avenue
- Forth
- Haskell
- Icon
- Logo
- Lua
- Delphi
- VBasic
- PHP
- Perl
- SGML
- XML
- HTML
- ASP
- Jawa
- ABAP
- Beta
- Mercury
- Miranda
- ML
- Oberon
- Prolog
- Rexx
- Ruby
- Sather
- SETL
- Lisp
- C
- V C++
- VFocus Pro
- Occam
- Power Builder
- Smalltalk
- Snobol
- ABC
- APL
- Flash
- Python
- Tcl
- Apple Script
- Shell Script
- VB Script
- Metlap
- CSH
- SH
- TSCH
- .....


1957
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00
01
02
03
04
Fortran I
Fortran II
ALGOL 58
FLOW-MATIC
ALGOL 60
APL
COBOL
LISP
Fortran IV
CPL
SNOBOL
SIMULA I
BASIC
PL/I
ALGOL W
SIMULA 67
ALGOL 68
BCPL
B
C
Pascal
Prolog
Scheme
MODULA-2
Fortran 77
awk
ML
Smalltalk 80
Ada 83
Miranda
ICON
COMMON LISP
C++
Perl
Haskell
QuickBASIC
Oberon
ANSI C (C89)
MODULA-3
Fortran 90
Eiffel
Visual BASIC
Python
PHP
Java
Fortran 95
Ada 95
Javascript
C99
C#
Visual Basic.NET
Java 5.0

Makina, assembly ve C dillerinin karşılaştırılması

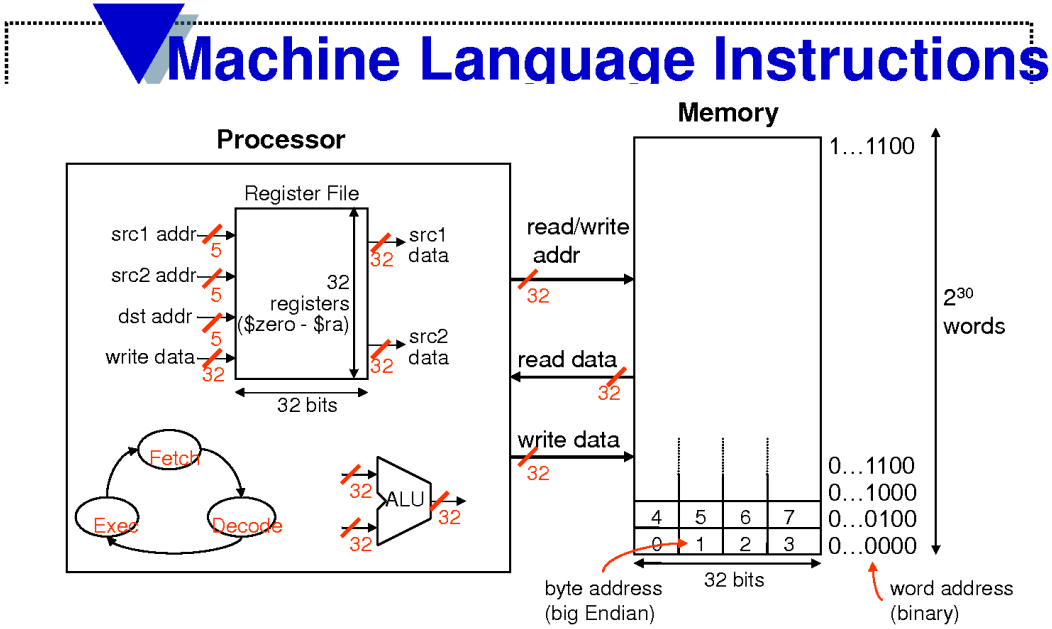
Bir işlemcide, iki farklı sayının toplanması için üç dilde yazılması gereken kodlar aşağıda verilmiştir.



Makina dili için;

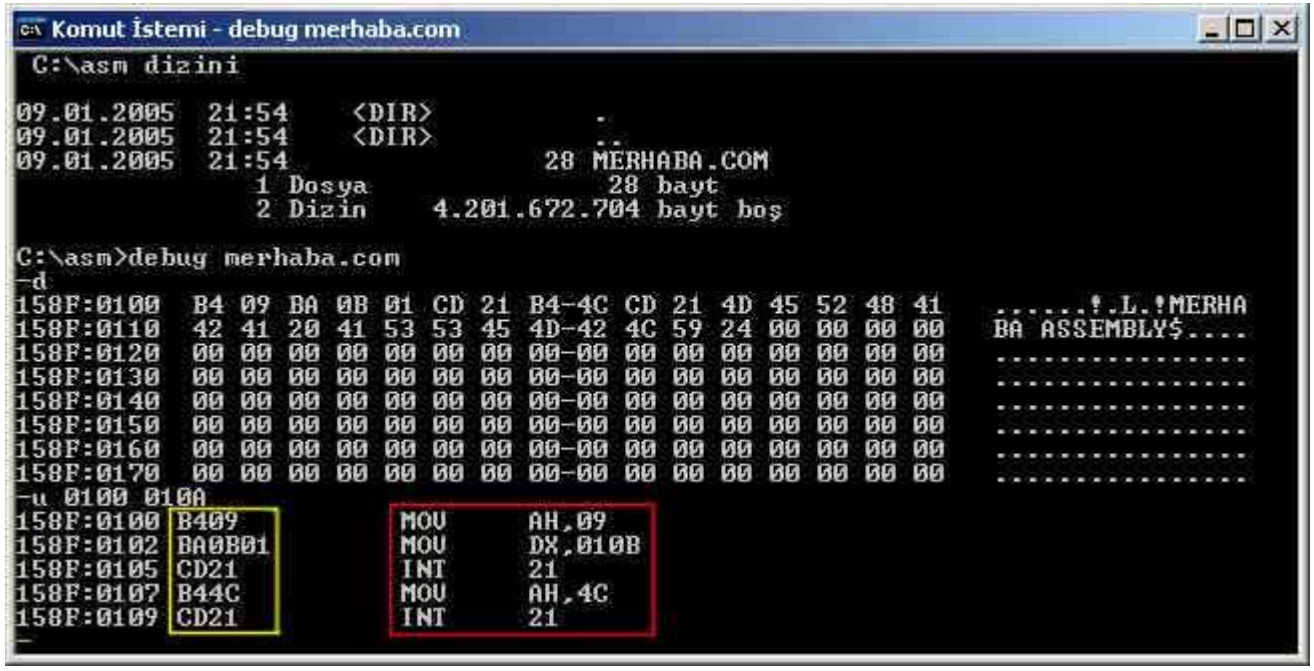
1081h	1. sayı
0322h	2.sayı
A003h	toplam

Verilen kod kullanılan makinaya özeldir. İşleme hızı hem assembly hem de C diline göre yüksektir. Yazımı oldukça zordur.



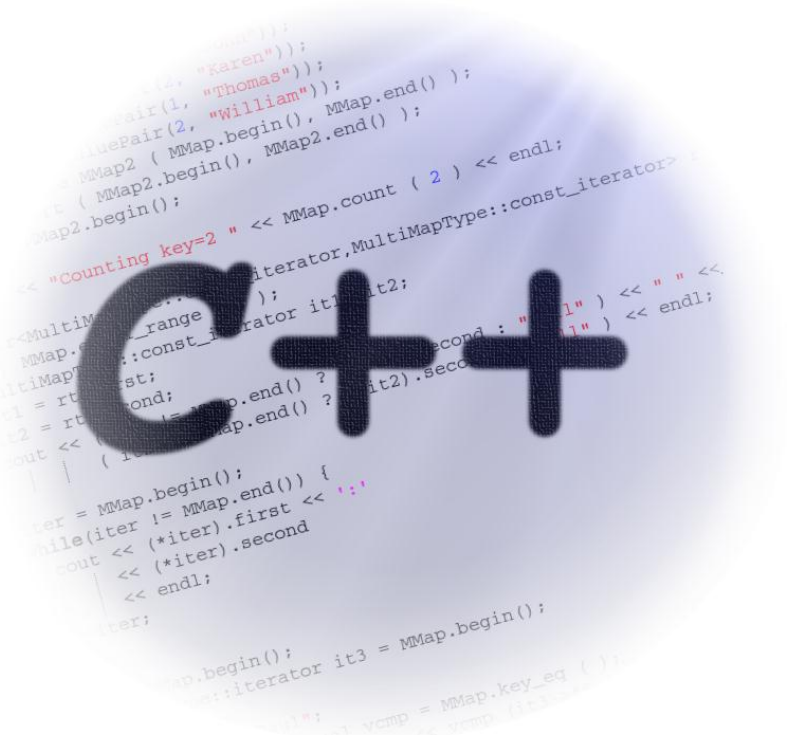
Assembly dili için ;
lacc a 1. sayı
add b 2.sayı
sacl c toplam

Verilen kod kullanılan makinaya özeldir. İşleme hızı C diline göre yüksektir. Yazımı C diline göre zordur.



C dili için;
c=a+b;

Verilen kod derleyicisi bulunan bütün makinalarda kullanılır. İşleme hızı diğerlerine göre yavaştır. Yazımı kolaydır.



Neden C++ ?

C Programlama dilinin kullanım alanları •

C programlama her tip programların yazımında kullanılmaktadır (işletim sistemleri, veri tabanları, editörler, derleyiciler v.b). Elektrik-Elektronik mühendisliğindeki genel kullanım alanları

- Elk-Elt devre analizlerinin yapılmasında
- Sistemlerin bilgisayar ile denetiminde
- Mikroişlemci (PIC, PLC vb)programlanmasında

PROGRAMCILIĞA GİRİŞ

Algoritma Ve Akış Şemaları

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:

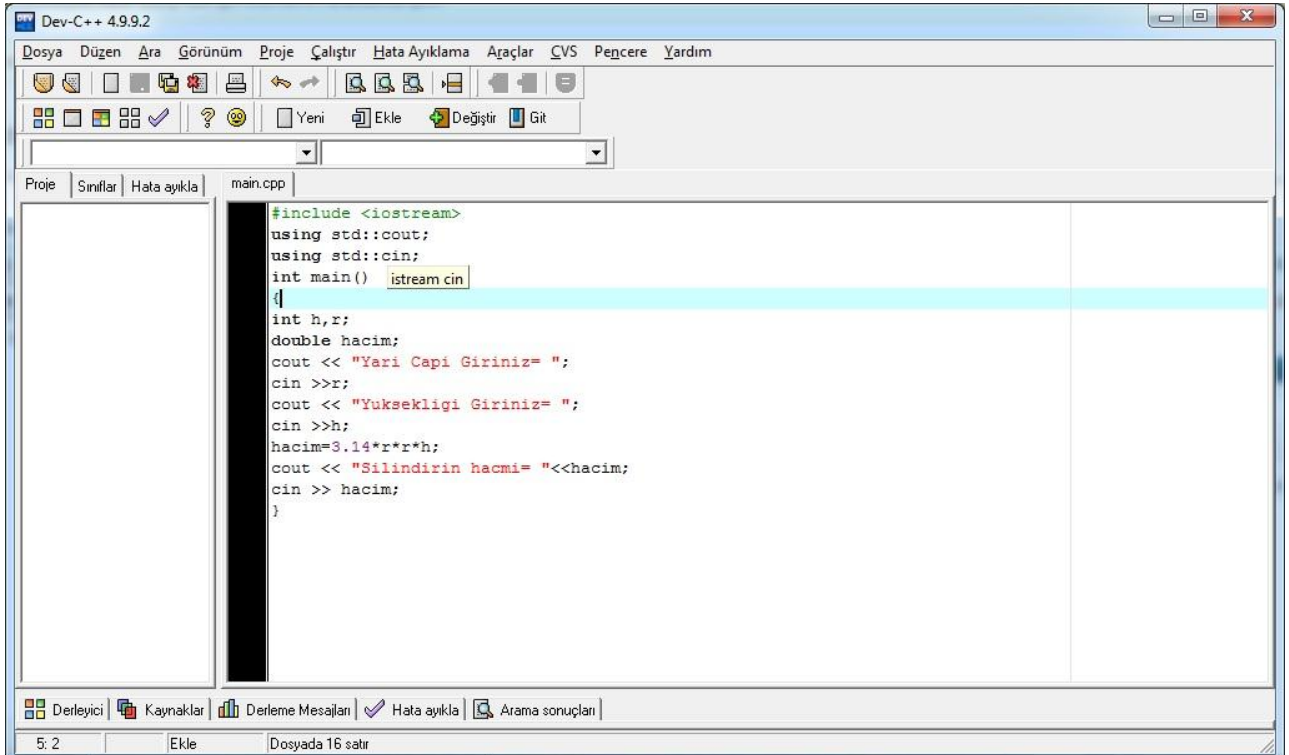
Akış şeması belirli bir işin yapılabilmesi için, basit işlemlerle şema halinde gösterilmesidir. Kısaca algoritmanın şemalarla gösterilmesidir.

Algortima 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, eldeki sonuçlarla çıkan sonuçlar karşılaştırılır. Programın doğru çalışıp çalışmadığı sınanır.

Örnek bir Algoritma

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

Çözüm 1:

1. Evden dışarıya çık
2. Otobüs durağına yürü
3. Durakta gideceğin yöndeki otobüsü bekle
4. Otobüsün geldiğinde otobüse bin
5. Biletini bilet kumbarasına at
6. İneceğin yere yakınlaştığında arkaya yürü
7. İneceğini belirten ikaz lambasına bas
8. Otobüs durunca in
9. İşyerine doğru yürü
10. İş yeri giriş kapısından içeriye gir
11. Mesai arkadaşlarınla selamlaş
12. İş giysini giy
13. 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-çıktı 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:

Sayi1=1
Sayi2=1
Sayi3=Sayi1+Sayi2
Sayi4=Sayi3+Sayi2
Sayi5=Sayi4+Sayi3

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.

AKIŞ ŞEMALARI

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.

Program Geliştirme Metodu

- Problemi anlama
- Çözüm metodu geliştirme (Analiz)
- Metodun adımlanması (Tasarım)
- Programı kodlama (Uygulama)
- Programın Testi

Niçin kullanılır?

Akış şemaları

- Programcı tarafından hazırlanır
- Programın genel görünümü
- Planını
- Akış yönünü
- Çözümleri adım adım gösteren

şemalardır.

Akış komutları

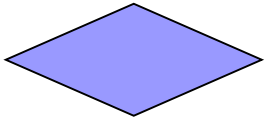
Başla-Bitir(sonlandırıcı) →

Input (girişler) →

İşlem →

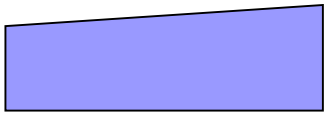
Görüntüleme →

Karar →



Tekrarlı işlem →

Elle girilen değer →



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

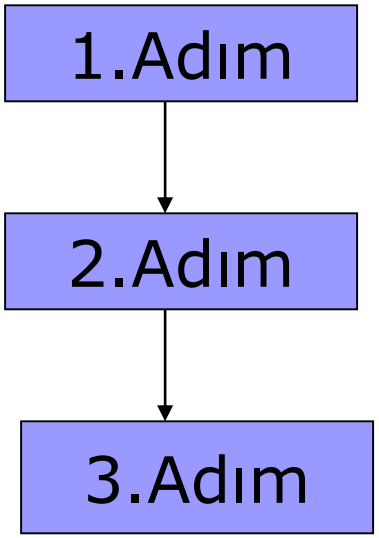
Bir bilgisayar programının geliştirilmesinde kullanılan programlama dili ne olursa olsun bu programların akış diyagramlarında genel olarak yalnız üç basit mantıksal yapı kullanılır.

Akış şemaları

- Sıralı akış
- Şartlı akış
- Tekrarlı akış

Sıralı Akış

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.



- Bütün işlemlerin sırayla biribirini takip ettiği akış
- Daha önceki işlemlere geri dönülmez
- Karşılaştırma yapılmaz

Örnek

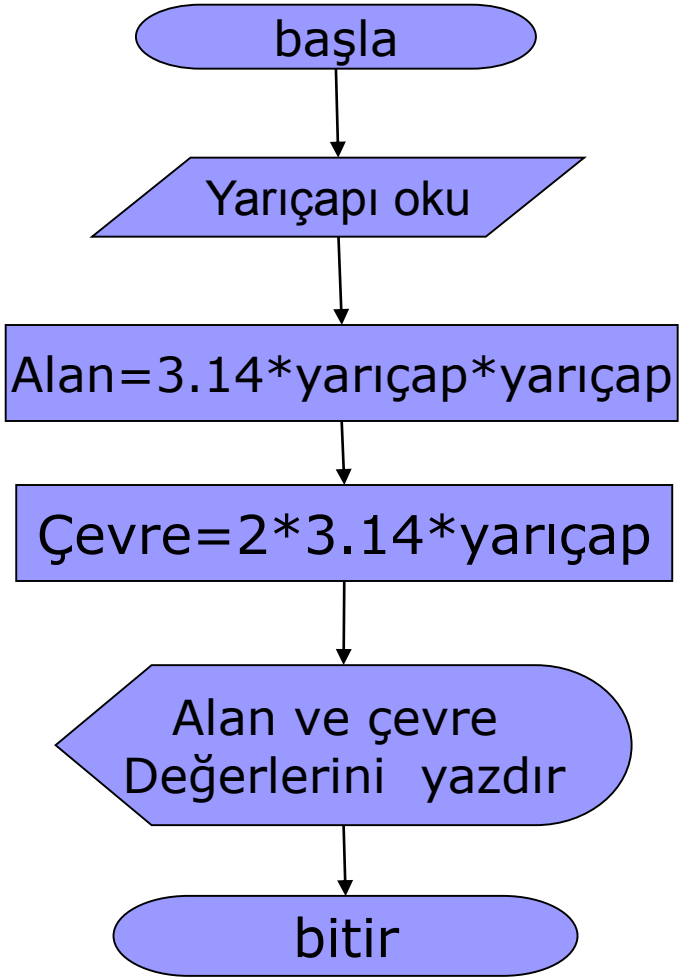
SORU:Yarıçapı verilen çemberin alanını ve çevresini hesaplayan program akışını çiziniz.

ANALİZ:

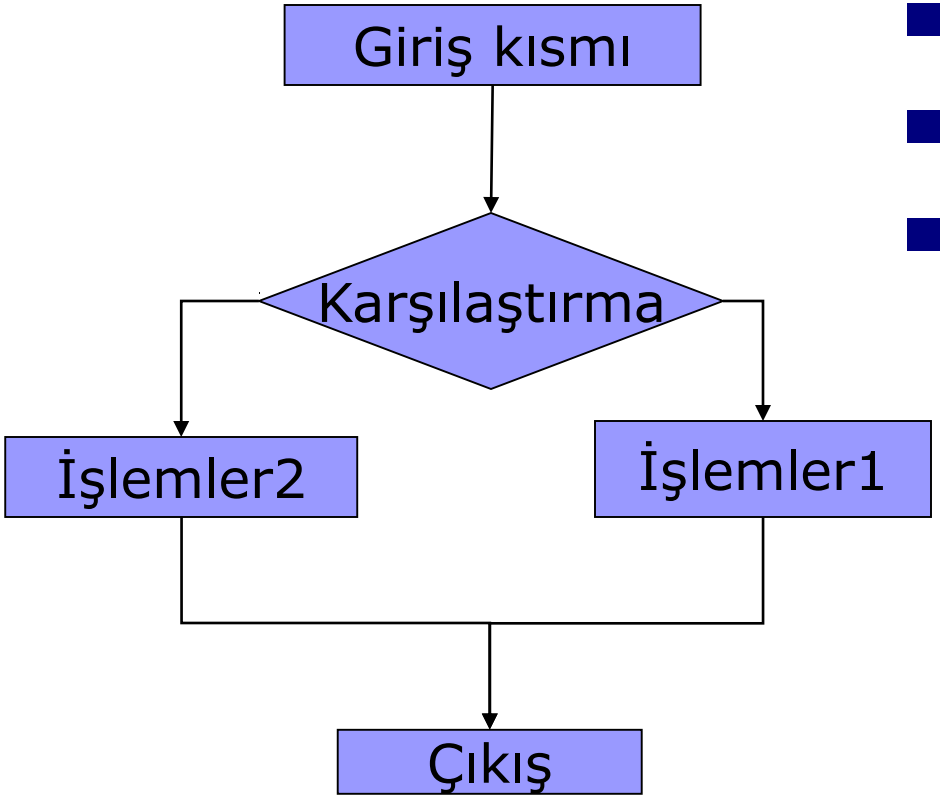
Çemberin yarıçapı okunur

Alan=pi*r^2 Çevre=2*pi*r

Sonuçlar ekrana yazılır


başla
Yarıçapı oku
Alan=3.14*yarıçap*yarıçap
Çevre=2*3.14*yarıçap
Alan ve çevre
Değerlerini yazdır
bitir

Şartlı Akış

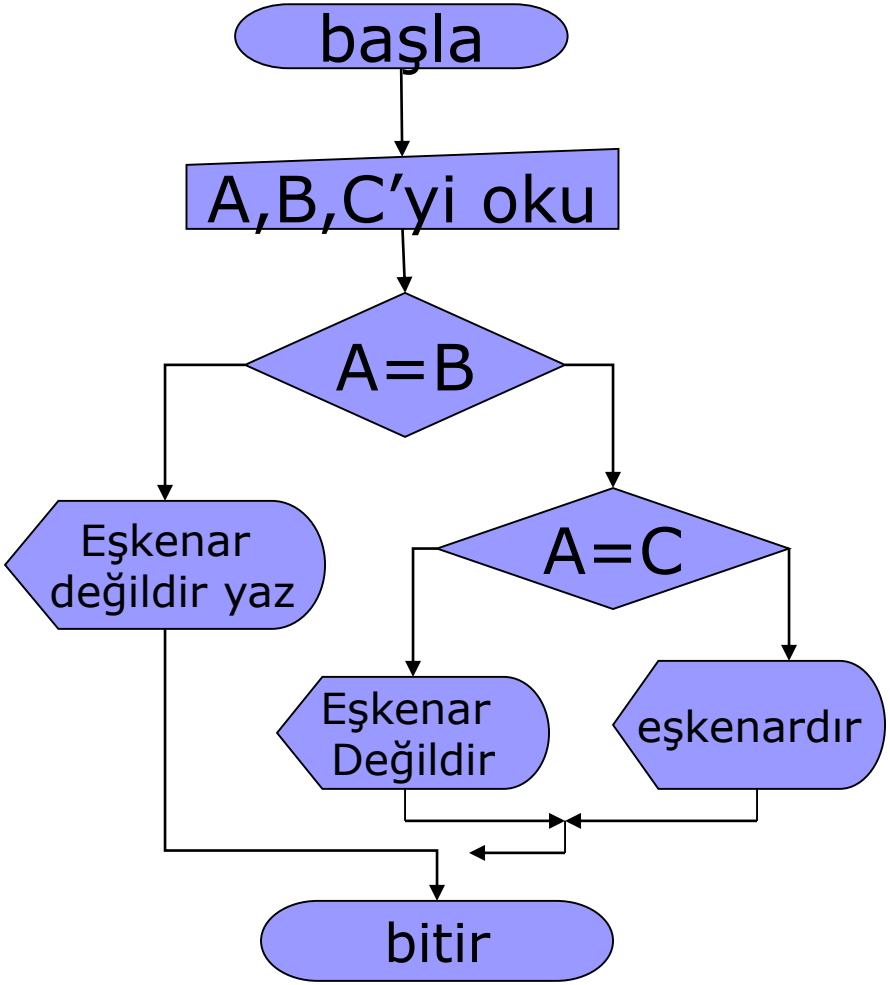


- Karşılaştırma ifadesi
- Doğru→Bir koldan
- Yanlış→Diğer koldan

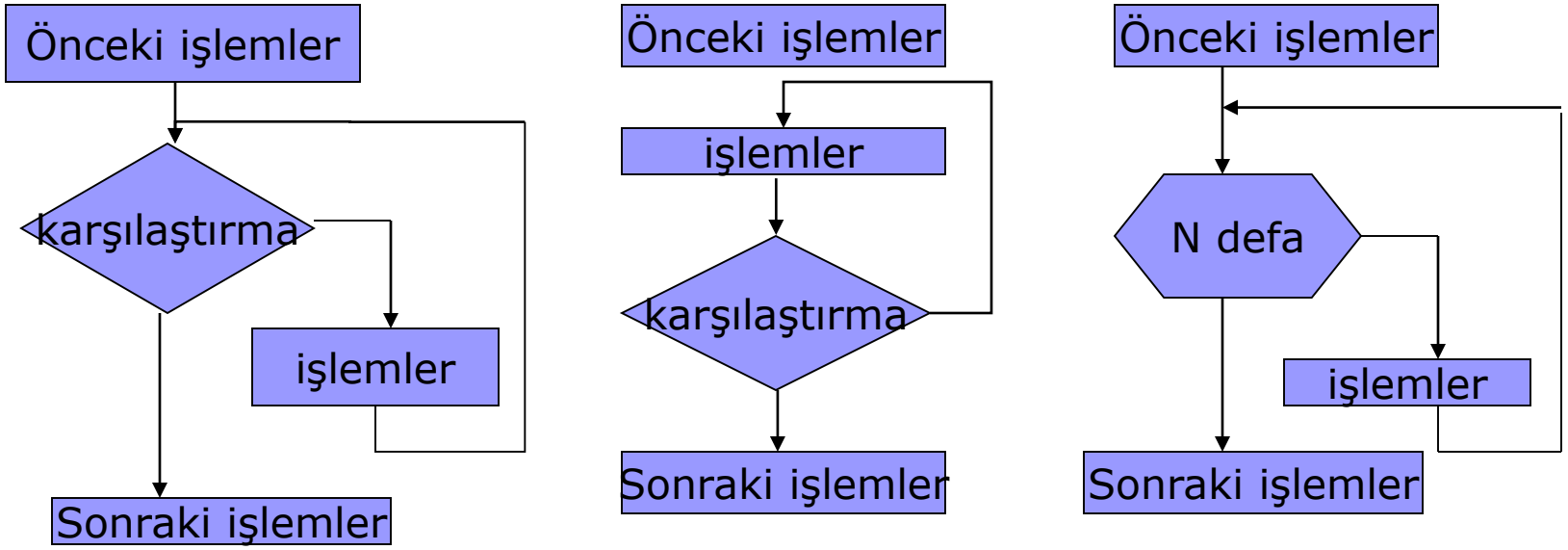
Örnek

SORU:

Üçkenarının uzunluğu girilen bir üçgenin eşkenar olup olmadığını test edecek program akışı şemasını geliştiriniz.


başla
A,B,C'yi oku
A=B
Eşkenar değildir yaz
A=C
Eşkenar Değildir
eşkenardır
bitir

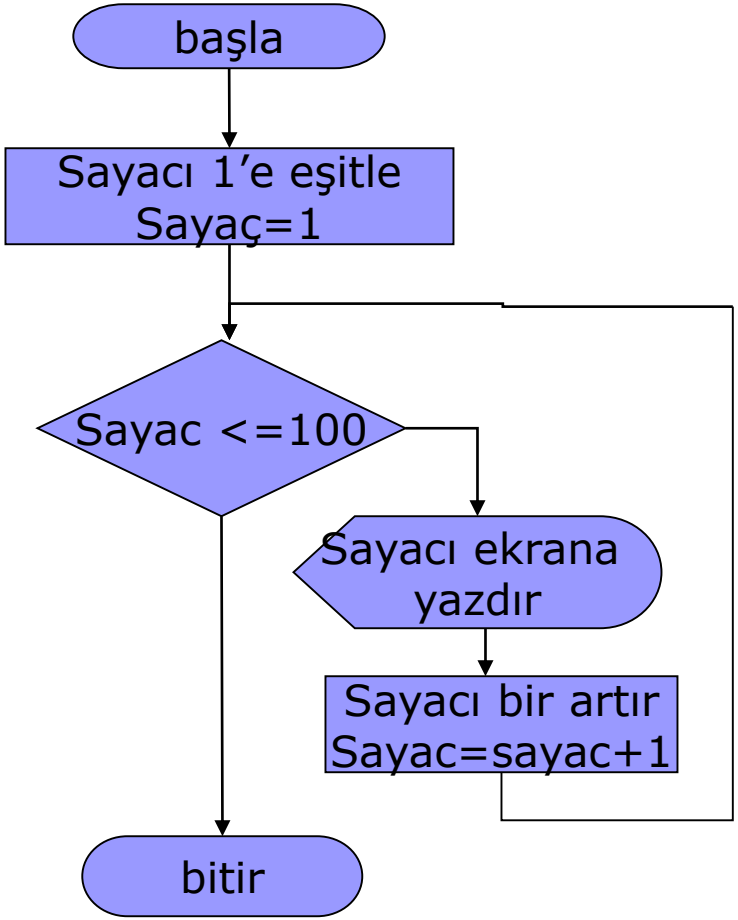
Tekrarlı Akış



- Aynı işlemleri birçok defa tekrar eden akış şemalarıdır.

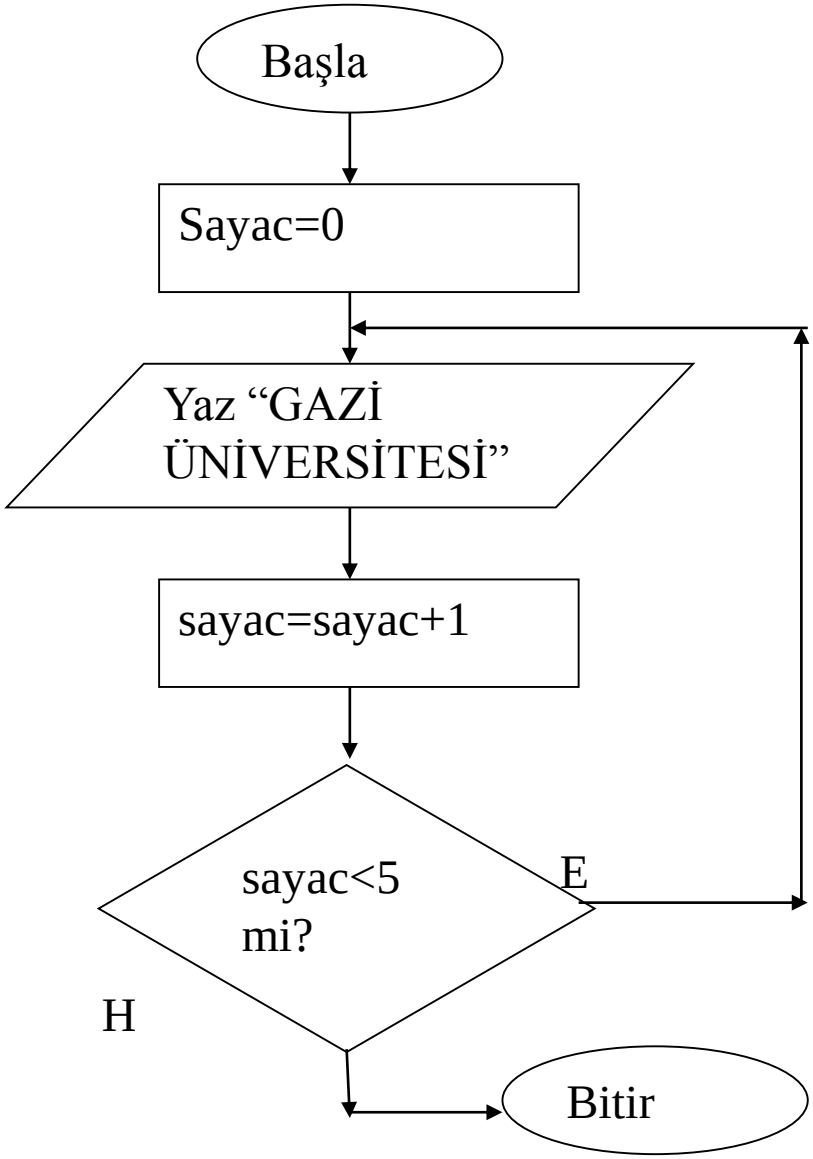
Örnek

SORU:1den 100 kadar sayıları yazdıran program akışını çiziniz


başla
Sayacı 1'e eşitle
Sayaç=1
Sayac <=100
Sayacı ekrana
yazdır
Sayacı bir artır
Sayac=sayac+1
bitir

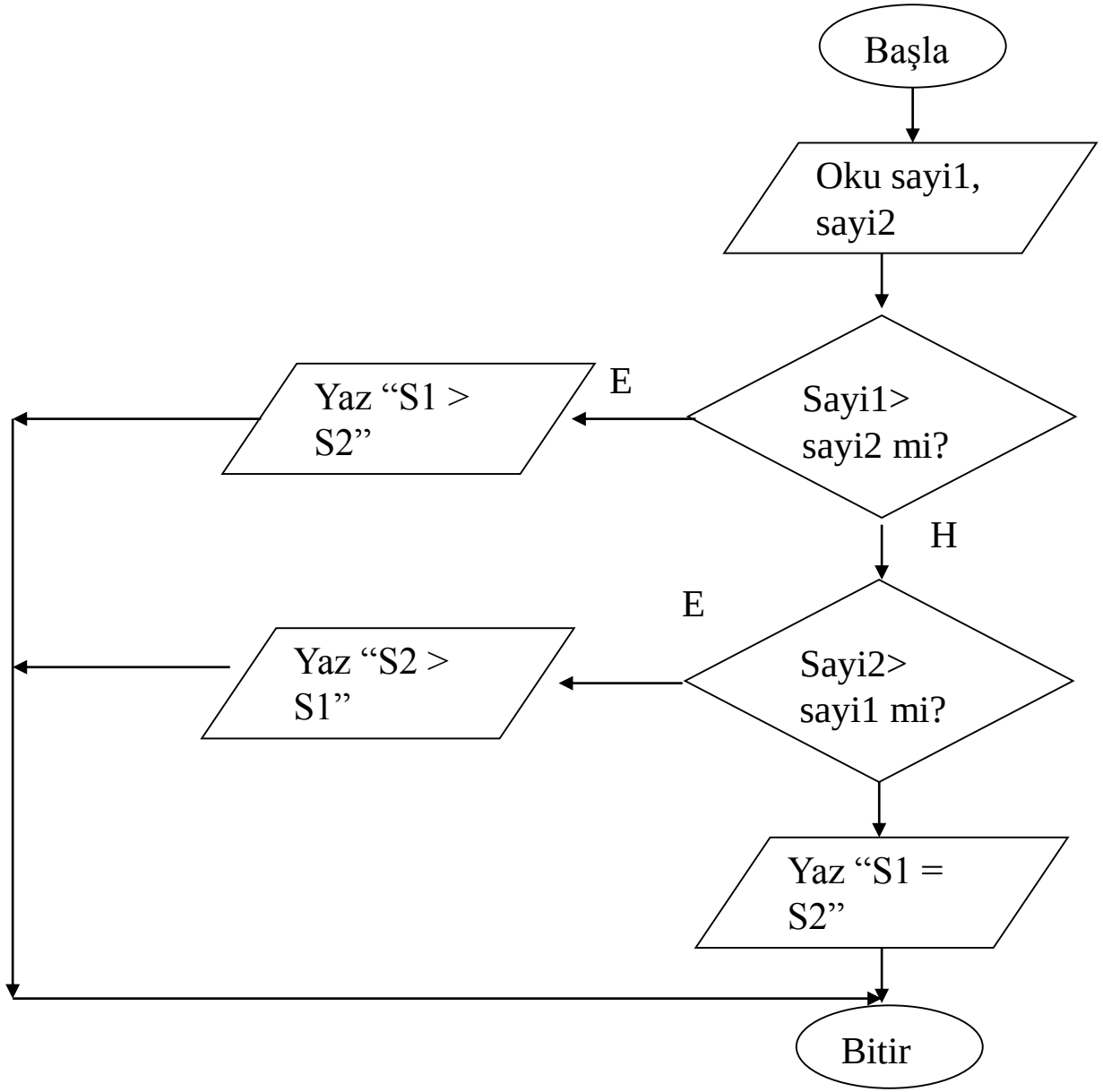
İsim ve soyadınızı ekrana 5 defa yazdıran programın algoritma ve akış şemasını yazın?

1. Basla
2. sayac=0
3. YAZ “GAZİ ÜNİVERSİTESİ”
4. sayac=sayac+1
5. Eğer sayac<5 GİT 3
6. DUR


Başla
Sayac=0
Yaz “GAZİ ÜNİVERSİTESİ”
sayac=sayac+1
sayac<5 mi?
E
H
Bitir

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

1. BAŞLA
2. OKU sayi1,sayi2
3. EĞER sayi1>sayi2 İSE YAZ “Sayi1 büyüktür”
4. Değilse EĞER sayi2>sayi1 İSE YAZ “Sayi2 büyüktür”
5. DEĞİL İSE YAZ “Sayi1 sayi2’ye eşittir”
6. BİTİR

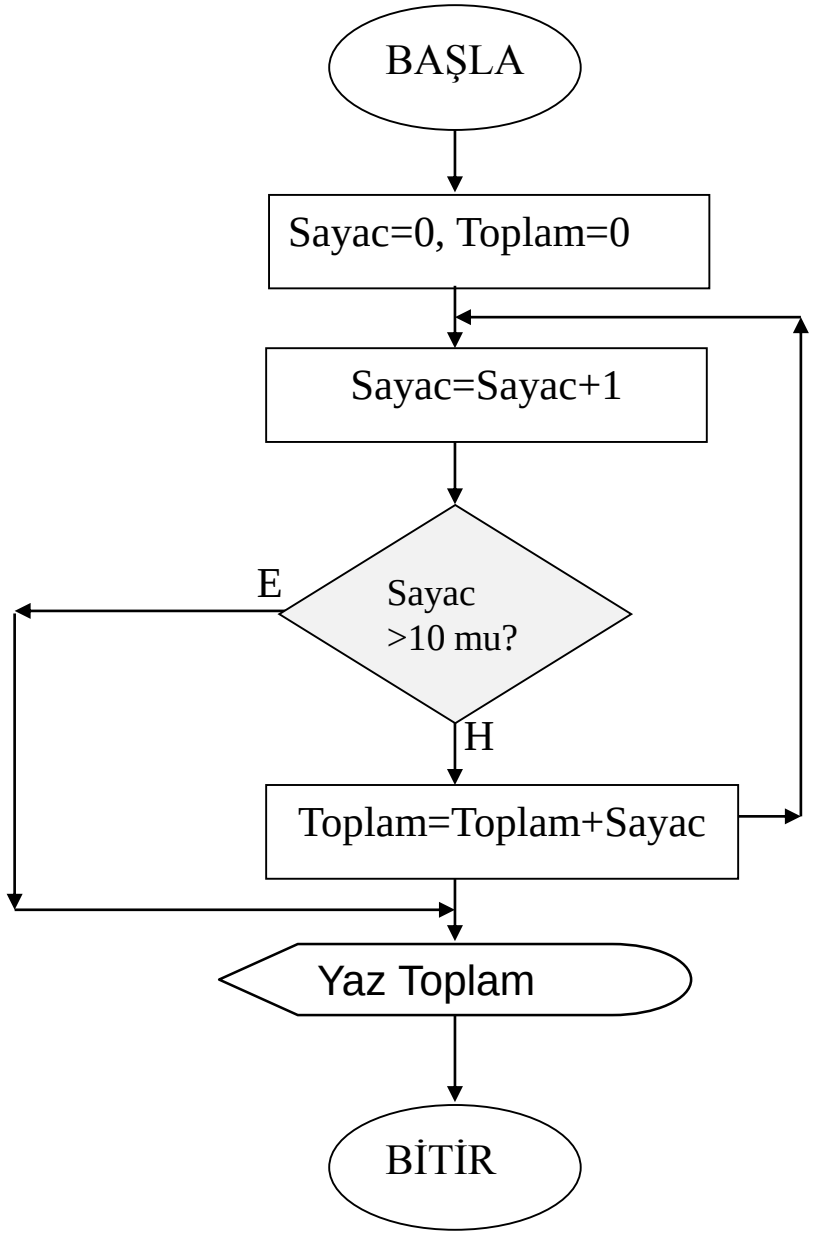

Başla
Oku sayi1, sayi2
Sayi1> sayi2 mi?
E
Yaz "S1 > S2"
H
Sayi2> sayi1 mi?
E
Yaz "S2 > S1"
Yaz "S1 = S2"
Bitir

Aşağıda verilen algoritmanın akış şemasını çizin ve programı izleyerek ne iş yaptığını belirtin?

1. BAŞLA
2. Sayi1=15
3. Sayi2=30
4. Yaz Sayi1, Sayi2
5. Gecici=Sayi1
6. Sayi1=Sayi2
7. Sayi2=Gecici
8. Yaz Sayi1, Sayi2

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


BAŞLA
Sayac=0, Toplam=0
Sayac=Sayac+1
Sayac
>10 mu?
E
H
Toplam=Toplam+Sayac
Yaz Toplam
BİTİR

ÖDEV 1 (gönderdiğiniz mailde ödev 1 olduğunu belirtiniz)

1. Klavyeden girilen 1-25 arasındaki bir tamsayının faktöriyelini alan programın algoritma ve akış diyagramını yazınız.
2. Klavyeden ardı ardına sayı girişi isteyen ve bu sayı 10 ile 15 arasında olmadığı sürece bu işleme devam eden programın algoritma ve akış diyagramını yazınız.
3. 1den 25 e kadar olan sayıların kareleri toplamını bulan programın algoritma ve akış diyagramını yazınız.
4. Klavyeden 10 tane tamsayı girilmesini isteyen ve bu girilen tamsayılardan kaç tanesinin negatif olduğunu bulan programın algoritma ve akış diyagramını yazınız.
5. a,b,ve c klavyeden girilmek üzere, ax2+bx+c=0 şeklindeki bir denklemin köklerini bulan programın algoritma ve akış diyagramını yazınız.
6. Klavyeden girilen 1-12 arasındaki tamsayıların hangi aya denk geldiğini bulup ekrana yazan programın algoritma ve akış diyagramını yazınız.
7. Dört işleme birer kod numarası vererek, klavyeden girilen iki sayıyı yine klavyeden girilen işlem koduna göre toplayan, çıkaran, çarpan veya bölen programın algoritma ve akış diyagramını yazınız.
8. Klavyeden ardı ardına girilen sayıları toplayan ve girilen sayı negatif olduğunda duran programın algoritma ve akış diyagramını yazınız.
9. Klavyeden bir not girilmesini isteyen ve bu not 0-49 arasındaysa “Başarısız”, 50-64 arasındaysa “Orta”, 65-84 arasındaysa “İyi”, 85-100 arasındaysa “Çok iyi “ Yazan programın algoritma ve akış diyagramını yazınız.
10. Klavyeden girilen iki tamsayıdan büyük olanı bulup ekrana yazdıran programın algoritma ve akış diyagramını yazınız.
11. Klavyeden girilen iki pozitif tamsayıdan birincisinin ikincisi cinsinden kuvvetini alan programın algoritma ve akış diyagramını hazır fonksiyon kullanmadan yazınız.
12. n! değerini hesaplayan programın algoritma ve akış diyagramını yazınız.
13. 1+4+9+ … +100= değerini hesaplayan programın algoritma ve akış diyagramını yazınız.
14. Toplama, çıkarama, çarpma ve bölme işlemi yapan programın algoritma ve akış diyagramını yazınız.
15. Saatte ortalama 60 km yol giden bir aracın, klavyeden girilen mesafeyi kaç saatte gideceğini hesaplayan programın algoritma ve akış diyagramını yazınız.