

# Doğayı ve Bilimi Keşfet Hayallerini Kodla!

Tübitak 4004 Projesi

**Hazırlayan**

Ali Osman Çıbıkdiken

Konya 2019

Merhaba,

Tübitak 4004 “Doğayı ve Bilimi Keşfet Hayallerini Kodla” Projesi kapsamında hazırlanan bu kitapçık Kodlama Eğitimi için temel kavramları ve bilgileri aktarmayı amaçlamaktadır.

Kitapçık; bilgisayar, algoritma, kodlama ile ilgili bilgileri vermekte ve Python programlama dili ile uygulamalarını kapsamaktadır.

Herkese faydalı olması dileğiyle...

*Ali Osman Çıbıkdiken, Konya, 2019*

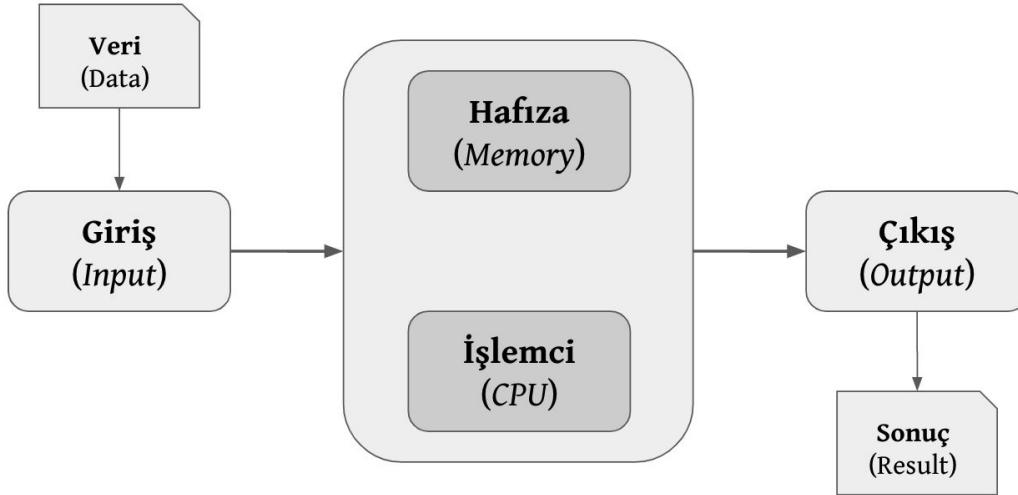
*aocdiken@gmail.com*

# Bilgisayar Nedir, Nasıl Çalışır?

Bilgisayar; önceden belleğine yüklenmiş bir yazılıma göre komuta edilerek, çok sayıda karmaşık mantıksal ve aritmetiksel işlemlerden oluşan bir işi, çok kısa sürede yapabilen ve yaptığı işlemlerin sonucunu saklayabilen, istenildiğinde geri getirebilen bir aygıttır.

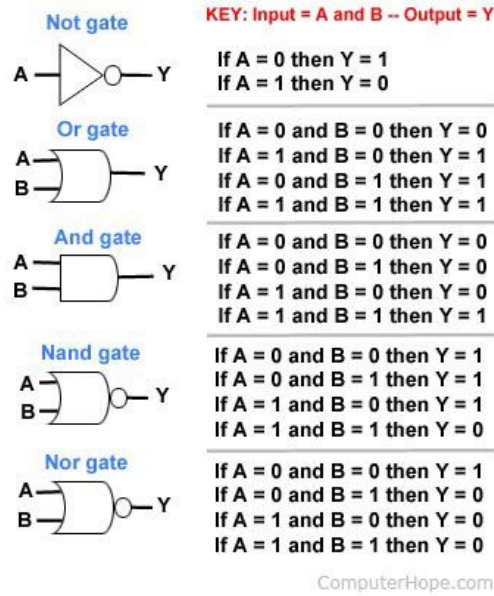
Günümüzde bilgisayarların temel yapısını elektronik parçalar oluşturmaktadır. Bu elektronik parçalar, farklı görevler için tasarlanmış olan ve bilgisayarın donanım (*hardware*) adını verdiğimiz kısmını oluşturur. Donanım üzerinde komutların çalıştırılmasını sağlayan kısmı ise yazılım (*software*) olarak adlandırılır.

Bilgisayarlar veriyi (*data*) giriş ünitelerinden (*input*) alır ve onlar üzerinde çeşitli matematiksel ve mantıksal işlemleri gerçekleştirip çıktı ünitelerinden (*output*) verir. İşlemler hafıza (*memory*) ve işlemci (*CPU-Central Processing Unit*) kısmında gerçekleştirilir. Modern bilgisayarların temel çalışma prensibi Şekil 1 de verilmiştir.



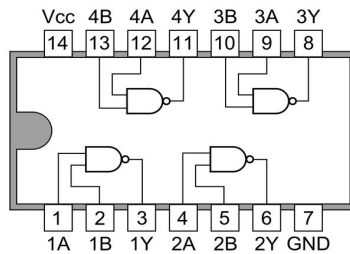
**Şekil 1.** Modern bilgisayarların çalışma prensibi

Bilgisayarların elektronik yapılardan oluştuğunu belirtmiştik. Bu yapılar basit anahtar devreleri ile çalışır. Bu anahtar devrelerine mantıksal kapı (*logical gate*) denir. Kapılar ikili durum dediğimiz bir yaklaşımla çalışırlar. Yani durumları; akım var/akım yok, açık/kapalı, 1/0 gibi değerlerle çalışır. Temel mantıksal kapılar Şekil 2 de gösterilmiştir.



**Şekil 2.** Temel mantıksal kapılar

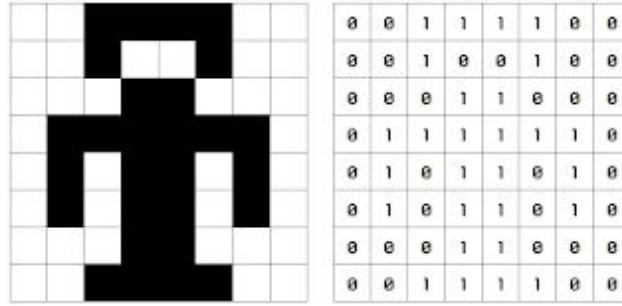
İşlemci dediğimiz bütün elektronik birimler aslında bu kapıların farklı görevlere sahip kombinasyonlarından oluşan yapılardan oluşmaktadır. Örneğin 7400 Quad işlemcisi içerisinde bulunan kapıların şematik gösterimi Şekil 3 de verilmiştir.



**Şekil 3.** 7400 Quad işlemcisinin mantıksal kapılarla şematik gösterimi

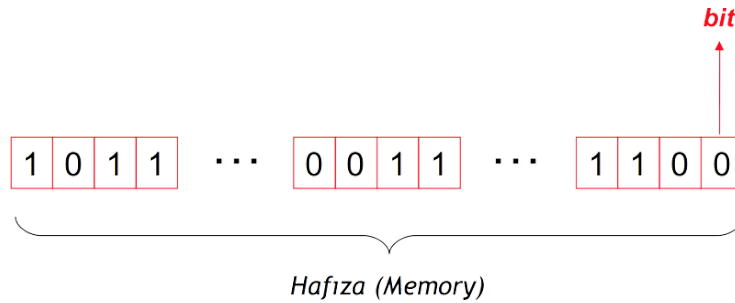
Matematiksel olarak ikili durumu ifade etmek için ikili sayı sistemi (*binary system*) kullanılır. Bu sisteme göre bir değer ya 1 (bir) ya da 0 (sıfır) değerini alır. Bu en küçük 1 ya da 0 dan oluşan bilgi parçasına ise bit (*binary digit*) denir.

Bilgisayarda kullanılan bütün sayı, yazı, resim, müzik, video, animasyon vb veri ikili sisteme çevrilerek hafızaya yüklenir işleme alınır. Örneğin bir oyundaki adam karakterinin 1 ve 0 ile hafızaya alınmış hali Şekil 4 de gösterilmiştir. Burada karakter; 00111100 00100100 00011000 01111110 01011010 01011010 00011000 00111100 olarak bilgisayar tarafından işleme alınır ve bu hafızada 64 bit yer kaplar.



**Şekil 4.** Bir oyun karakterinin 1 ve 0 ile gösterimi

Hafıza bilgisayarın işleme almak için veriyi sakladığı alandır (Şekil 5).



**Şekil 5.** Bilgisayar hafızası ve bit

Her türlü bilgi bit'lerle ifade edildiğinden herhangi bir verinin büyüklüğü bit'in katları ile gösterilir. Kapasite büyüklüğünün bit'lerin katları ile gösterilmesi Şekil 6 da verilmiştir.

kapasite	sembol		deger
1 bit	Bit	=	0 veya 1
1 byte	Byte	=	8 bit
1 Kilobyte	KB	=	1024 bytes
1 Megabyte	MB	=	1024 KB
1 Gigabyte	GB	=	1024 MB
1 Terabyte	TB	=	1024 GB
1 Petabyte	PB	=	1.024 TB

**Şekil 6.** Kapasite büyüklüğünün bit'lerle ifade edilmesi

Şekil 6 da verilen bilgilerle eğer bilgisayar hafızası 4 GB ise;

$$4 \text{ GB} \times 1024 = 4096 \text{ MB}$$

$$4096 \text{ MB} \times 1024 = 4\,194\,304 \text{ KB}$$

$$4\,194\,304 \text{ KB} \times 1024 = 4\,294\,967\,296 \text{ byte}$$

$$4\,294\,967\,296 \text{ byte} \times 8 = 34\,359\,738\,368 \text{ bit}$$

olarak bulunur.

## Bilgisayar Hafızası ve Sayı Sistemleri

Bilgisayarların hafıza kapasiteleri bit ve onun katları ile belirleniyordu. Her bir bit değerini bir hücre gibi düşünebiliriz. O halde hafıza aslında hücrelerden oluşan bir yapıdır. Hücrelerden oluşan bu yapı bizim verdiğimiz her türlü veriyi ikili sisteme çevirip kullanır. Bu çevirme işlemi matematikte anlatılan taban aritmetiği ile kolayca yapılabilir. Örneğin 127 sayısını bilgisayarın hafızasına almak istersek ikili tabana göre çevirme işlemi yapılması gerekir. Taban aritmetiğine göre, sağdan sola doğru olmak üzere her bir basamak 2 ve katlarına göre değer alır. Şekil 7 de 43 sayısının ikili tabandaki yani ikili sistemdeki karşılığı görülmektedir.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$									
0	0	1	0	1	0	1	1									
x	x	x	x	x	x	x	x									
128	64	32	16	8	4	2	1									
↓	↓	↓	↓	↓	↓	↓	↓									
0	+	0	+	32	+	0	+	8	+	0	+	2	+	1	=	43

Şekil 7. 43 sayısının ikili sistemdeki karşılığı

43 sayısının bilgisayar hafızasındaki yeri ise Şekil 8’de verilmiştir.



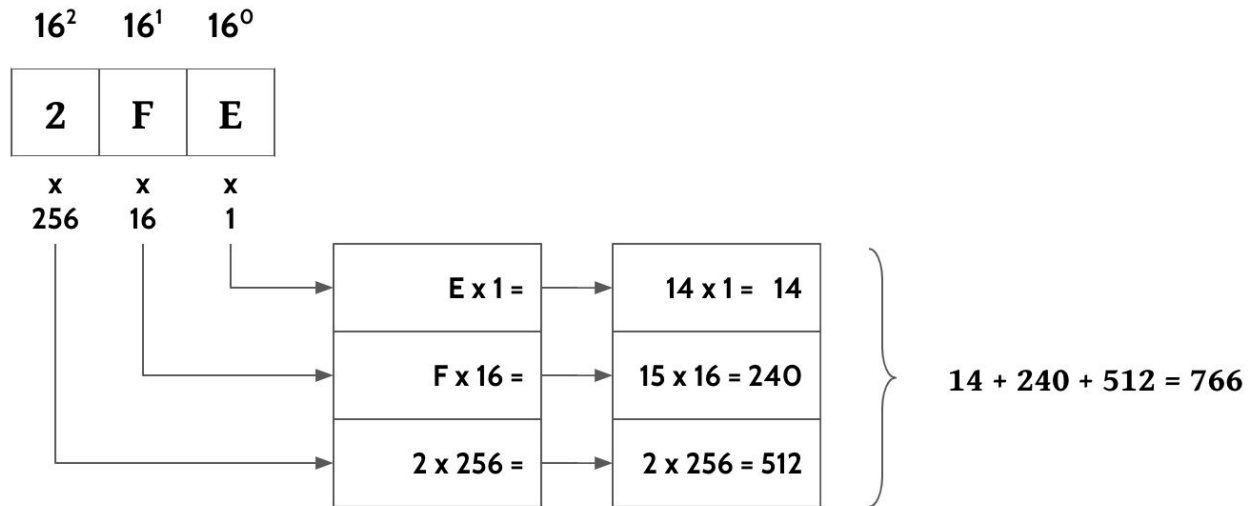
Şekil 8. 43 sayısının bilgisayar hafızasına yerleştirilmesi

Bilgisayar dünyasında kullanılan bir diğer sayı sistemi de onaltılı (hexadecimal) sayı sistemidir. Örneğin, bilgisayardaki renk kodları onaltılı sayı sistemi ile ifade edilir. Tıpkı ikili sistem gibi onaltılı sistemde de taban aritmetiği ile dönüşümler kolayca yapılabilir. Onaltılı sistemde sayıların dönüşümünde sembolize ederken rakamlar yeterli olmadığı için harfler de kullanılmaktadır. Sayı sembol dönüşümleri Şekil 9 da gösterilmiştir.

Sayı	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Onaltılı sembol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

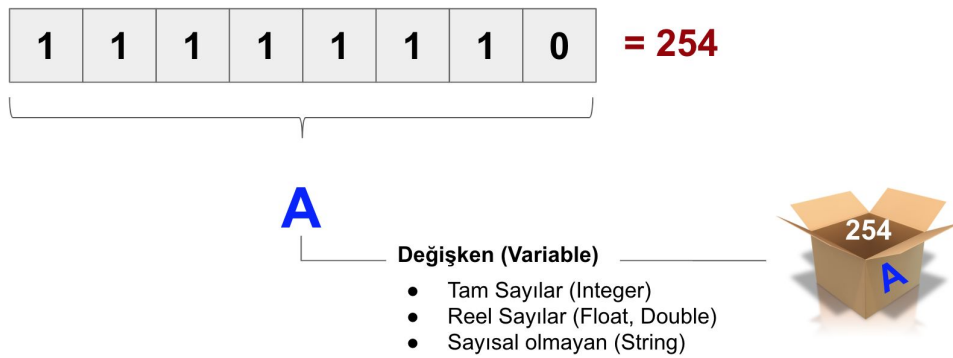
**Şekil 9.** Onaltılı sayı sisteminde kullanılan değerler

Şekil 9 da verilen tabloya göre 2FE sayısı 766 olarak aşağıdaki şekilde bulunur.





Bilgisayar hafızası bit hücrelerinden oluşan bir yapı olduğundan yönetimi bilgisayar tarafından kolayca yapılıyor olmasına rağmen program yazarken takip etmek oldukça zordur. Bu yüzden hücrelerden oluşan kutular şeklinde tarifleyerek kullanmak daha kolay olacaktır. Bu kutular bilgisayarın teknolojisine göre 8 bitlik, 16 bitlik, 32 bitlik veya 64 bitlik olarak kullanılabilir. Bu kitapçık boyunca kolaylık olması açısından 8 bitlik tercih edilmiştir. Kutu büyüklüğüne belirlenmesi ile artık bu kutulara istenilen değer saklanabilir. Kutulara programlama dünyasında **değişken** (*variable*) adı verilir. Kutu isimlerinin, saklanacak değer ne amaçla kullanılıyor ise genellikle onu anımsatacak şekilde verilmesi uygun olur. Kutu isimlerini verirken mümkün olduğunca İngiliz alfabesinde olmayan Türkçe harfler kullanılmaması ihtiyaç olursa eşleniklerinin kullanılması uygun olacaktır. Bu eşlenik harfler “ı=i, ö=o, ü=u, ğ=g, ç=c, ş=s” şeklindedir. Kutular içindeki değerler tam sayılar (integer), reel sayılar (floating point) ve metin (string) şeklinde olabilir. Metin şeklinde saklanacak değerlerin ise sayısal karşılıkları kutularda saklanır. Metin değerlerinin sayısal karşılığı ASCII kod tablosu ile bulunur. 254 değerinin hafızada A değişkeni ile saklanması Şekil 10 da verilmiştir. A değişkeni herhangi bir şekilde kullanıldığında bilgisayar tarafından hafızadan A ya karşılık gelen 254 değeri işleme alınacaktır. Herhangi bir değişkene yeni bir değer atanmak istenildiğinde eski değer silinir.

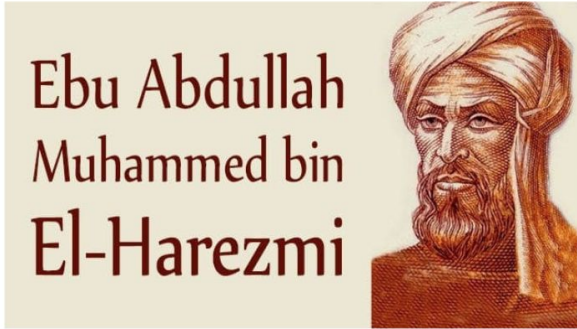


**Şekil 10.** 254 sayısının hafızada A değişkeni ile saklanması

## Problem Çözümü ve Algoritma

Bilgisayara yaptırmak istediğimiz her şey aslında onun çözmesini istediğimiz bir problemdir. Oyun oynatması, hesap yapması, video göstermesi, internete bağlanması, bir veri indirmesi vb bütün olaylar bilgisayar için yapılması gereken ama nasıl yapıldığını bizim söylediğimiz problemlerdir.

Bir problemin çözümü için yapılması gereken sonlu ve sıralı adımların listesine **algoritma** (*algorithm*) denir. Algoritma yaklaşımı ilk defa bir müslüman bilim adamı olan Ebu Abdullah Muhammed bin El-Harezmi tarafından ortaya konmuştur. Algoritma ismi de batılıların El-Harezmi'nin ismini Latince söylenmesinden ortaya çıkmıştır. Cebirin babası olarak bilinen El-Harezmi yazmış olduğu “El’Kitab’ül-Muhtasar fi Hıساب’il Cebri ve’lMukabele (Cebir ve Eşitlik Üzerine Özet Kitap)” kitabı ile meşhur olmuştur. Batı kaynaklarında tıpkı algoritma gibi cebire de isim babalığı yapan El-Harezmi olmuştur.



**Arapça** : El-Harezmi  
**Latince** : Al-Khwarizmi  
**İngilizce** : Algorithm

**Arapça** : El-Cebri  
**Latince** : Al-Gebra  
**İngilizce** : Algebra

**El’Kitab’ül-Muhtasar fi Hıساب’il Cebri ve’lMukabele”  
(Cebir ve Eşitlik Üzerine Özet Kitap)**

Algoritma bir problemin çözüm adımları olduğuna göre bu adımları oluşturan yapılarla herhangi bir çözüm kolaylıkla tanıtılabilir. Algoritma çözümünde kullanılan 5 temel yapı kullanılır. Bu yapıların hepsi birarada kullanılmak zorunda değildir. Tek tek ya da bir tanesi çok kere ihtiyaç oluyorsa kullanılabilir.

Yaygın bir örnek olarak kullanılan ay demleme algoritması Őekil 11 de verilmiřtir. Verilen bu algoritma sıralı ve sonlu adımlarla nasıl ay demlenir problemini özmektedir.

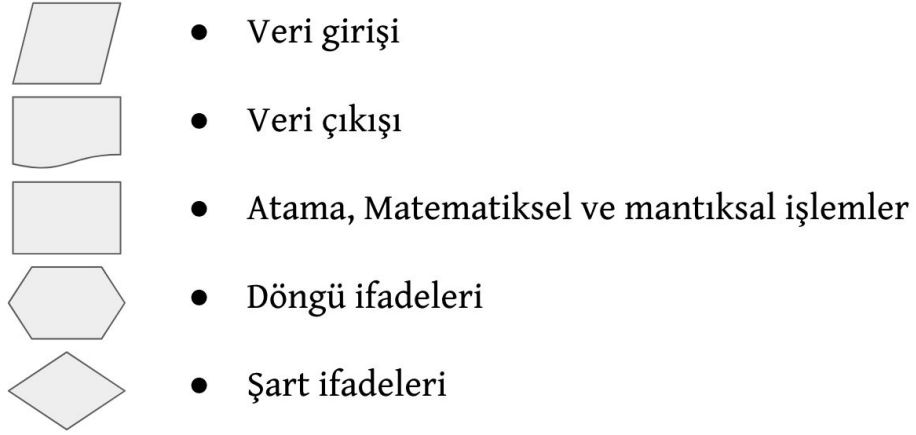
1. Bařla
2. aydanlıęa su doldur.
3. Ocaęı yak.
4. aydanlıęı ocaęa koy.
5. Bir süre suyun kaynamasını bekle.
6. Su kaynadı mı? Kaynamadıysa 5.adıma git. Kaynadıysa 7.adıma git.
7. ayı demle.
8. aydanlıęa su ilave et.
9. aydanlıęı tekrar ocaęa koy.
10. Bir süre suyun kaynamasını bekle.
11. Su kaynadı mı? Kaynamadıysa 10.adıma git. Kaynadıysa 12.adıma git.
12. Bir süre ayın demlenmesini bekle.
13. ay demlendi mi? Demlenmediyse 12.adıma git. Demlendiyse 14.adıma git.
14. ayı bardaklara servis yap.
15. Dur

**Őekil 11.** ay demleme algoritması

Algoritma yazmada kullanılan 5 temel yapı ařaęıda verilmiřtir:

- Veri giriři (*input*),
- Veri ıkışı (*output*),
- Deęiřkenler için atama, matematiksel ve mantıksal iřlemler (*assign, math and logical operations*),
- Döngü ifadeleri (*loop*),
- Őart ifadeleri (*if*)

Algoritma adımları için kullanılabilecek 5 temel yapı genellikle İngilizce olarak ifade edilmesine raęmen, algoritma hangi dilde kullanılıyor ise o dilde yazılabilir. Fakat bütün herkesin dil farkı olmadan 5 temel yapıyı algılayabilmesi için řematik gösterim geliřtirilmiřtir. Buna akıř řeması (*flowchart*) denir. Algoritmanın akıř řeması ile ifadesinde kullanılan semboller Őekil 12 de verilmiřtir.



**Şekil 12.** Algoritmanın 5 temel yapısının akış şeması karşılıkları

Algoritmanın 5 temel yapısını kullanım detayları aşağıdaki şekilde verilmiştir:

### 1. Veri Girişı



#### Veri Girişı

Tam sayı, reel sayı ve string ifadelerin kullanıcı tarafından girişinin yapıldığı ve bir değişkende saklandığı durumları gösterir.



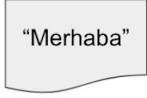
Kullanıcı tarafından girilen değeri **A** değişkeninde sakla

### 2. Veri Çıkışı



#### Veri Çıkışı

Herhangi bir mesajı ya da değişkende saklı olan değeri çıktı ünitesine (ekran, yazıcı vb) göndermek için kullanılır.



Çıktı olarak "Merhaba" mesajını gönderir.



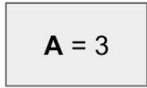
Z değişkeninin değerini çıktı olarak gönderir.

### 3. Atama, Matematiksel ve Mantıksal işlemler

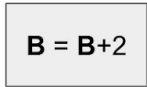


#### Atama, Matematiksel ve Mantıksal işlemler

Aritmetik işlemler ya da mantıksal işlemlerin gerçekleştirilmesi için kullanılır.

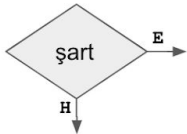


A değişkenine 3 değeri atar (saklar).



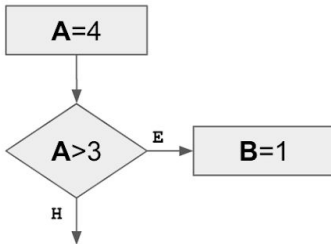
B değişkeninin mevcut değerine 2 ekler ve tekrar B değişkenine atar (saklar).

### 4. Şart İfadeleri



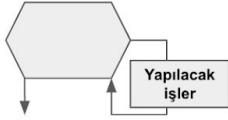
#### Şart İfadeleri

Verilen şarta göre (Evet, True, Doğru) değeri alıyorsa E kolundan, (Hayır, False, Yanlış) değeri alıyorsa H kolundan devam ederek gerekli adımları yerine getirir.



A değişkeninin değeri 3 den büyük olduğu için E (Evet) kolundan devam eder ve B değişkenine 1 değerini atar (saklar). H (Hayır) kolundan devam edildiğinde yapılacak bir iş yoktur.

### 5. Döngü İfadeleri



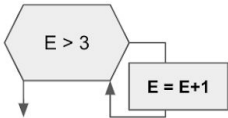
### Döngü ifadeleri

Birden fazla tekrarlanan aynı işleri yerine getirir. Yapılacak işler döngü içinde ifade edilir. Döngü sayısı belli olan ve döngü bir şarta bağlı olan şeklinde iki türü vardır.



### Döngü sayısı belli olan:

T değerini 1 den 5 kadar birer artırır ve A değişkenine atar (saklar).



### Döngü bir şarta bağlı olan:

E değeri 3 ten büyük olduğu sürece E değişkeninin değerini 1 artırır ve E değişkenine atar (saklar).

Kullanıcı tarafından verilen iki sayının toplamı probleminin çözümünü algoritma olarak ifade edilmek istenirse aşağıdaki gibi yazılabilir:

Adım 0. Başla

Adım 1. Veri girişi, **A**

Adım 2. Veri girişi, **B**

Adım 3. **C = A + B**

Adım 4. Veri çıkışı, **C**

Adım 5. Son

Bu algoritma İngilizce olarak aşağıdaki şekilde yazılabilir:

Step 0. Start

Step 1. Input A

Step 2. Input B

Step 3. **C = A + B**

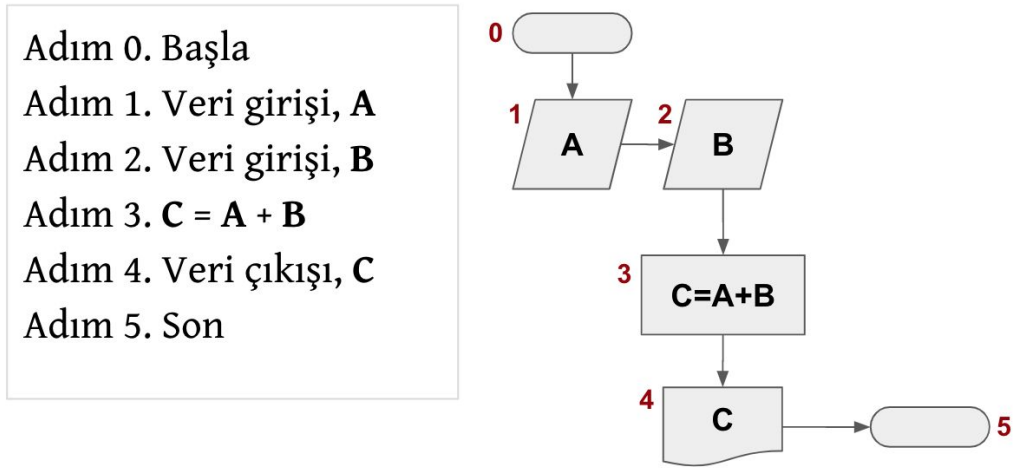
Step 4. Output C

Step 5. End

Aynı algoritma Çince olarak ise şu şekilde yazılır:

第0步。开始  
步骤1.输入A.  
第2步。输入B.  
步骤3.C = A + B.  
步骤4.输出C.  
第5步。结束

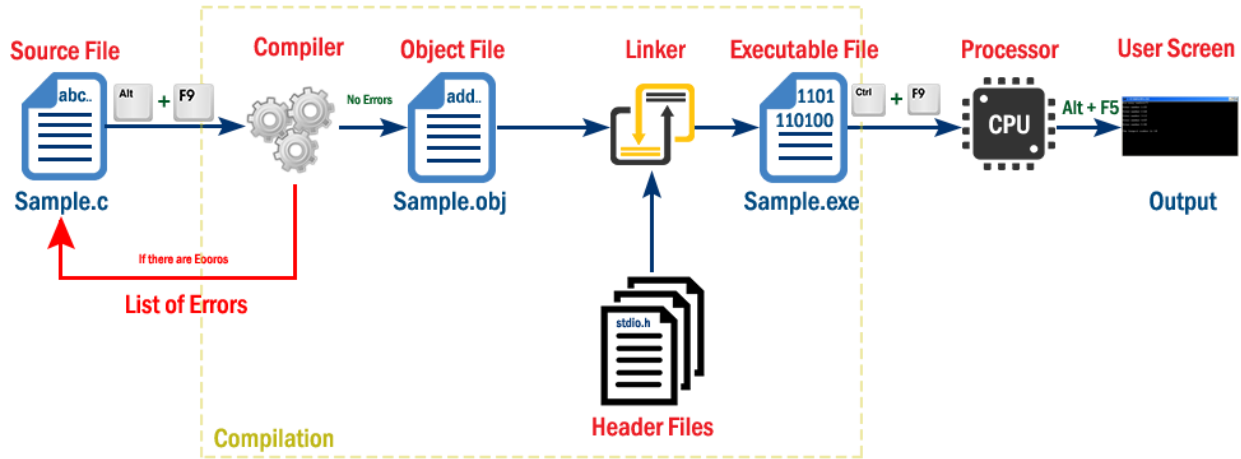
Görüldüğü gibi algoritma farklı dillerde ifade edilebilir ve o dili bilen kişi tarafından okunabilir. Aynı algoritmayı akış şeması ile gösterdiğimizde artık kullanılan dilin bir önemi yoktur. Algoritmanın akış şeması ile gösterimi Şekil 13 de verilmiştir. Akış şemasının her bileşeninin yanında gösterilen kırmızı sayılar adım numarasını vermektedir.



**Şekil 13.** Kullanıcı tarafından verilen iki sayının toplamını bulan algoritmanın akış şeması

## Programlama ve Python Programlama Dili

Bir problemin çözümü için algoritma yazıldıktan sonra onu bilgisayarın anlayacağı hale getirmek için bir programlama diline ihtiyaç vardır. Bunun için bir metin editörü yazılımı (*text editor*) ile program yazılır ve dosya olarak saklanır. Algoritmanın herhangi bir programlama diline uygun yazılmasına **programlama** (*programming*) denir. Herhangi bir programlama dili ile yazılmış dosya **kaynak kod** (*source code*) olarak adlandırılır. Kaynak kodun bilgisayar tarafından anlaşılır hale getirilmesine **derlemek** (*compiling*) denir. Derlenen bir dosya artık **çalıştırılabilir bir dosya** (*executable file*) olarak tanımlanır. Çalıştırılabilir dosya artık işlemci tarafından işleme alınır ve sonuç kullanıcıya verilir. Tanımlanan bütün bu süreçler Şekil 14 de verilmiştir.



**Şekil 14.** Kaynak kodun derlenip çalıştırılarak kullanıcıya gösterilmesi süreci

Python pek çok uygulama geliştirmek için popüler olarak kullanılan bir programlama dilidir. Python, Guido Van Rossum adlı Hollandalı bir programcı tarafından geliştirilmiştir. Python programlama dili ismini, Guido Van Rossum'un çok sevdiği Monty Python adlı altı kişilik bir İngiliz komedi grubun Monty Python's Flying Circus adlı gösterisinden almaktadır. Python, yazım dili

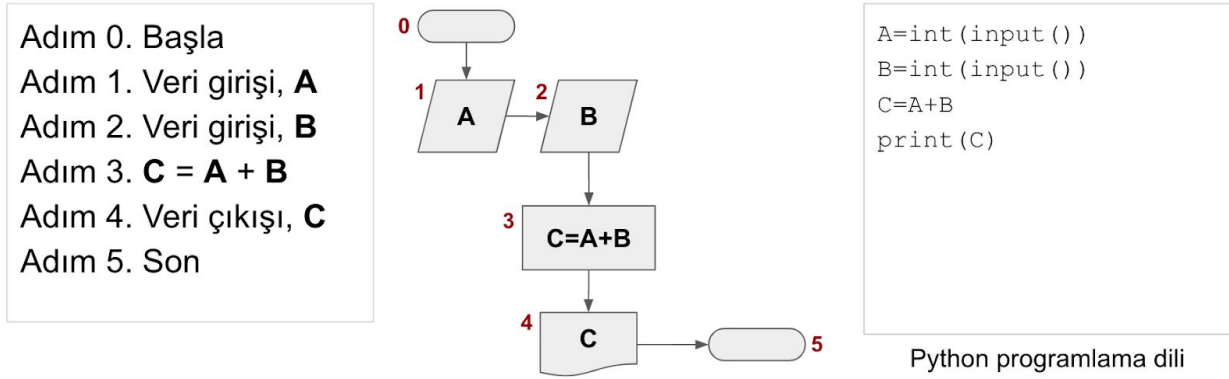


anlaşılabilen, kolay öğrenilebilen ve uygulama geliştirme sürecini kısaltan özelliklere sahiptir. Windows, Linux, MacOS gibi pek çok platformda rahatlıkla çalışabilir. Açık kaynak kodlu (open source) ve ücretsiz (free) bir yazılımdır.

Python, Linux tabanlı işletim sistemlerinde kurulu olarak hazır çalışır durumdadır. Windows kullanıcıları ise <http://www.python.org/downloads> web sitesinden Python yazılımını indirmeli (*download*) ve kurulum (*installation*) yapmalıdır. Kurulum ile ilgili talimatlara siteden ulaşabilirsiniz.

Python günümüzde yaygın olarak Python 2 ve Python 3 versiyonları ile kullanılmaktadır. Uygulama geliştirme sürecinde hangi versiyon tercih edilecekse karar verip ona göre kurulum yapılmalıdır. Bu kitapçık boyunca Linux işletim sistemi ve Python 2 versiyonu tercih edilmiştir.

Kullanıcı tarafından verilen iki sayının toplamı algoritmasının Python programlama dili ile yazılmış hali Şekil 15 te verilmiştir.



**Şekil 15.** Kullanıcı tarafından verilen iki sayının toplamını bulan algoritma, akış şeması ve Python programlama dilinde yazılmış hali

## Örnekler

**Örnek 1.** Kullanıcı tarafından girilen maaşın 5000TL den fazla ise %20 az ise %15 vergi kesintisi yaparak net maaşı hesaplayan algoritmayı yazınız, akış şemasını çizin ve Python programlama dilinde yazınız?

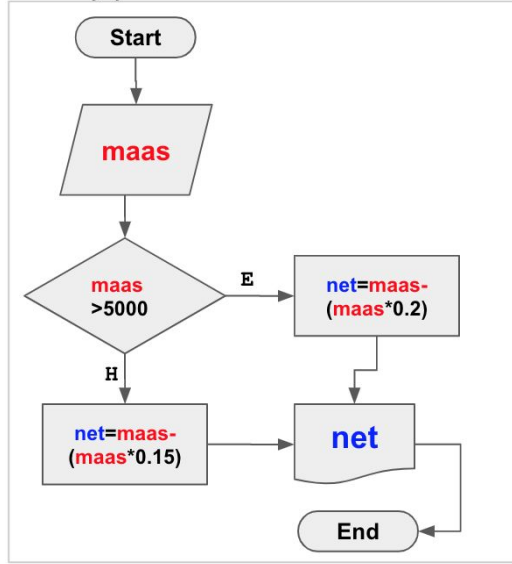
### Algoritma

Adım 0. Start  
Adım 1. Input **maas**  
Adım 2. if **maas**>5000 then  
**net**=**maas** - (**maas**\*0.20) else  
**net**=**maas** - (**maas**\*0.15)  
Adım 4. Output **net**  
Adım 5. End

### Kaynak kod

```
maas=int(input())  
if maas>5000:  
    net=maas - (maas*0.20)  
else:  
    net=maas - (maas*0.15)  
print(net)
```

### Akış Şeması



**Örnek 2.** 1 den 5 kadar olan tam sayıların 2 katını bulan algoritmayı yazınız, akış şemasını çizin ve Python programlama dilinde yazınız?

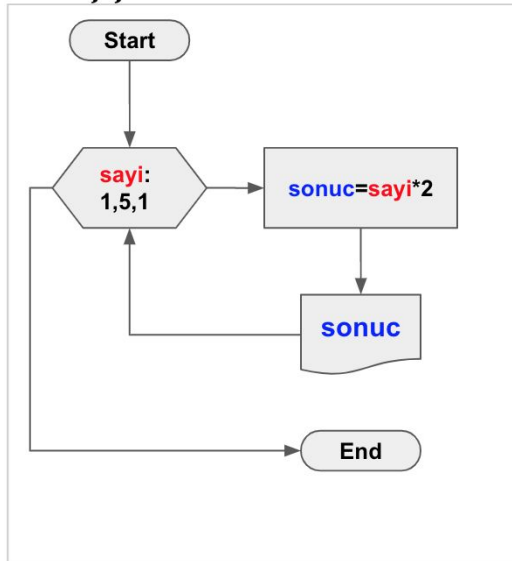
### Algoritma

Adım 0. Start  
Adım 1. for **sayi**:1,5,1  
    **sonuc**=**sayi**\*2  
    Output **sonuc**  
Adım 2. End

### Kaynak kod

```
for sayi in range(1,5):  
    sonuc=sayi*2  
    print(sonuc)
```

### Akış Şeması



**Örnek 3.** Kullanıcı tarafından girilen 4 tane sayının pozitif olanlarının toplamını bulan algoritmayı yazınız, akış şemasını çizin ve Python programlama dilinde yazınız?

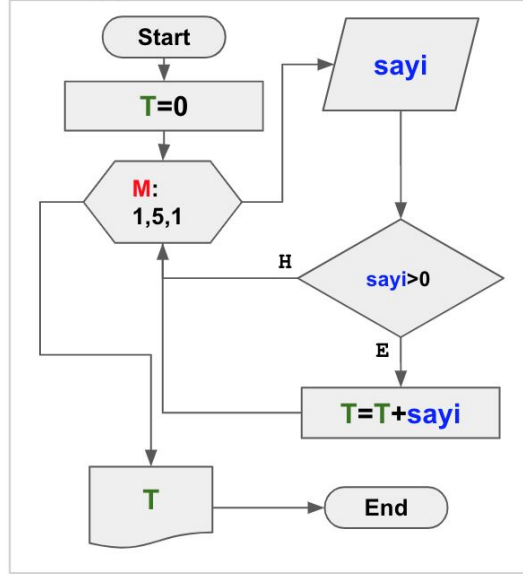
#### Algoritma

Adım 0. Start  
Adım 1.  $T=0$   
Adım 2. for  $M:1,5,1$   
    Input  $sayi$   
    if  $sayi>0$  then  $T=T+sayi$   
Adım 3. Output  $T$   
Adım 4. End

#### Kaynak kod

```
T=0
for M in range(1,5):
    sayi=int(input())
    if sayi>0:
        T=T+sayi
print(T)
```

#### Akış Şeması



**Örnek 4.** Bir öğrencinin yazılı ve sözlü notunun ortalamasını bularak, ortalama 1-3 arasında ise “D”, 4-6 arasında ise “C”, 7-8 arasında ise “B”, 9 dan büyük ise “A” notu ile değerlendiren algoritmayı yazınız ve Python programlama dilinde yazınız?

#### Algoritma

Adım 0. Start  
Adım 1. Input  $S$   
Adım 2. Input  $Y$   
Adım 3.  $F=(S+Y)/2$   
Adım 4. If  $F \geq 1$  and  $F \leq 3$  then  $sonuc="D"$   
Adım 5. If  $F \geq 4$  and  $F \leq 6$  then  $sonuc="C"$   
Adım 6. If  $F \geq 7$  and  $F \leq 8$  then  $sonuc="B"$   
Adım 7. If  $F \geq 9$  then  $sonuc="A"$   
Adım 8. Output  $sonuc$   
Adım 9. End

#### Kaynak kod

```
S=int(input())
Y=int(input())
F=(S+Y)/2
if F>=1 and F<=3:
    sonuc="D"
if F>=4 and F<=6:
    sonuc="C"
if F>=7 and F<=8:
    sonuc="B"
if F>=9:
    sonuc="A"
print(sonuc)
```