1. Bir problem çözümü için mevcut tercih edilen terim olan algoritma , günümüzde bir makinenin (ve özellikle bir bilgisayarın) belirli bir hedefe ulaşmak için izlediği kurallar dizisi için yaygın olarak kullanılmaktadır. Bununla birlikte, her zaman bilgisayar aracılı etkinlik için geçerli değildir. Terim, bilgisayar destekli veri analizi için olduğu kadar, pizza yapımında veya Rubik Küpünü çözmede izlenen adımlar için de doğru bir şekilde kullanılabilir. Hazırlanan algoritmalar hem metin şeklinde, hem sözde kodlar hem de geometrik şekillerde gösterilmektedir. Algoritma nedir, sorusunun cevabının iyi bir şekilde anlaşılabilmesi için algoritmanın ne işe yaradığı konusunda bilgi sahibi olunması gerekir. Hazırlanan başarılı bir algoritma sayesinde;

Program yazmak kolaylaşır.Hatalı kodlama oranı asgari düzeye iner.Programın işlem akışı görülür. Böylece program kontrolü kolaylaşır.Program yazımı pratik hale geldiği için zaman sarfiyatı olmaz.Sonradan programa ekstra özellik eklenmesi kolaylaşır.Program hızlı bir şekilde çalışır.Yazılan program bellekte fazla yer kaplamamalıdır.Algoritmanın başarılı olabilmesi için tüm ihtimallerin belirtilmesi gerekir ve bu ihtimallerin şansa bırakılmayacak şekilde net olması önemlidir. Ayrıca kağıt halinde belirtilmesi için yazılışı basit olmalıdır.

1. Önce akış diyagramı çizilmelidir.
2. Algoritma oluşturulurken girdiler, yani kullanılacak değerler belirlenmeli.

Algoritmanın sonunda çıktı, yani bir değer, sonuç üretilmeli.

Algoritmanın işlem sırası ve komutları açık olmalı, farklı sonuçlara yönlendirmemeli.

Farklı olasılıklar ve sonuçlar için Algoritmalar sonlu adımlardan oluşmalı.

Her işlem kullanıcının kağıt üzerinde gösterebileceği kadar basit olmalı.

1. Değişken herhangi bir yazılım dilinde bir veri tipinin değerini tutan bir kapsayıcıdır. Bir değişken herhangi bir veri tipinde yaratılabilir. Değişkenin kapladığı boyut içerdiği veri tipine göre değişkenlik gösterir.

Eğer değişkenler olmasaydı programlama yaparken tüm olasılıkları tek tek yazmak zorunda kalırdık. Değişkenler sayesinde hem zaman kazanabiliriz hem de hata olasılığı daha düşük seviyelerde olur.

1. Bazı işlemlerin belirli sayıda yaptırılması ve üretilen değerlerin sayılması gerekebilir.  
   Bu tür sayma işlemlerine algoritmada Sayaç adı veriler.Sayaçlar da birer değişkendir.
2. Bu algoritma, S değişkenini 2'şer 2'şer artırarak T değişkenini günceller. S değişkeni 10'dan büyük olduğunda T'yi yazdırır ve sona erer. Algoritmanın başlangıç değerleri (T=0 ve S=0) ve S'nin artış hızı (S=S+2) dikkate alındığında, S değişkeni 10'a ulaşana kadar T değişkeni bir değer almaz. Ancak S 10 veya daha büyükse, T değeri hesaplanır ve yazdırılır.
3. Bu algoritma, S değişkenini 3'er 3'er azaltarak F değişkenini günceller. S değişkeni 1'den küçük olduğunda F değeri hesaplanır ve yazdırılır. Başlangıç değerleri (F=1 ve S=20) ve S'nin azaltma hızı (S=S-3) dikkate alındığında, S 1'den küçük olduğunda F'nin değeri hesaplanır ve yazdırılır.
4. # Kullanıcıdan üç sayıyı girmesini isteyin

A = float(input("Birinci sayıyı girin: "))

B = float(input("İkinci sayıyı girin: "))

C = float(input("Üçüncü sayıyı girin: "))

# En büyük sayıyı bulma

if A >= B and A >= C:

en\_buyuk = A

elif B >= A and B >= C:

en\_buyuk = B

else:

en\_buyuk = C

# Sonucu ekrana yazdırın

print("En büyük sayı:", en\_buyuk)

1. # Kullanıcıdan üç sayıyı girmesini isteyin

A = float(input("Birinci sayıyı girin: "))

B = float(input("İkinci sayıyı girin: "))

C = float(input("Üçüncü sayıyı girin: "))

# Sayıları sıralama

if A > B:

A, B = B, A

if A > C:

A, C = C, A

if B > C:

B, C = C, B

# Sıralanmış sayıları ekrana yazdırın

print("Küçükten büyüğe sıralanmış sayılar:", A, B, C)

1. # Toplam ve çarpım değişkenlerini başlatın

tek\_toplam = 0

cift\_toplam = 0

tek\_carpim = 1

cift\_carpim = 1

# 1 ile 99 arasındaki sayıları işleyin

for sayi in range(1, 100):

if sayi % 2 == 0: # Çift sayı

cift\_toplam += sayi

cift\_carpim \*= sayi

else: # Tek sayı

tek\_toplam += sayi

tek\_carpim \*= sayi

# Sonuçları yazdırın

print("Tek sayıların toplamı:", tek\_toplam)

print("Çift sayıların toplamı:", cift\_toplam)

print("Tek sayıların çarpımı:", tek\_carpim)

print("Çift sayıların çarpımı:", cift\_carpim)