

IEEE

METUNCC

C Programlama

```
11011000010101100100010000111000100111
0100110010110100110110100111101111011110
00011010001000111010001101000011010
010010011010001010001110
10001001int main()
010101001{
1110011000    printf("Hello World");
0010000011    return 42;
0001101000}
01001001101111010111011110000001010001110
```

Her Şeyden Önce

- Veri: Hesaplama ya da manipülasyon amacı ile kullanılır.
- Input: Girdi, programa dışarıdan bir veri girişi yapılmasıdır.
- Output: Programdan alınan çıktıdır.
- Derleyici: Programcılıkta, bir programlama dilinde yazılmış olan kaynak kodunu başka bir dile (genellikle makine koduna) çeviren yazılım.

Bilgisayar Ve Programlama Nedir?

Bilgisayar en basit açıklamayla,

- Girilen veriyi alma(INPUT),
 - O veriyi işleme(PROCESSING)
 - Bu işlenmiş veriden bir sonuç verme(OUTPUT)
- olarak 3 temel göreve sahip bir makinedir.

Bilgisayar Ve Programlama Nedir?

Fakat bilgisayar sadece donanım olarak çalışamayıp, bu işlemleri yapabilmek için donanıma komut veren programlara ihtiyaç duyar.

Yani programlama ise ,bilgisayar donanımına ne yapmasını gerektiğini ,bilgisayarların anlayabileceği bir şekilde söylemektir.

Neden Programlama Öğrenmeliyiz?

- Esnek bir kariyer şansı
- Karlı bir yatırım
- Güvenli ve rahat bir meslek
- Evden çalışma imkanı
- Bir adım önde olmak
- Programlama bilgileri kendiniz için kullanılabilir






















Neden Programlama Öğrenmeliyiz?



Nedir bu C ?

- C, 1972 yılında Amerika'da AT & T Bell Laboratuvarlarında **Dennis Ritchie** tarafından dizayn edilmiş ve yazılmış bir programlama dilidir.
- Ayrıca C günümüzde hala kullanım alanı ve popülerlik olarak pek çok programlama dilinin üstünde yer almaktadır.

Nedir bu C?

Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java	  	100.0	100.0
2. C	  	99.9	99.3
3. C++	  	99.4	95.5
4. Python	 	96.5	93.5
5. C#	  	91.3	92.4
6. R		84.8	84.8
7. PHP		84.5	84.5
8. JavaScript	 	83.0	78.9
9. Ruby	 	76.2	74.3
10. Matlab		72.4	72.8

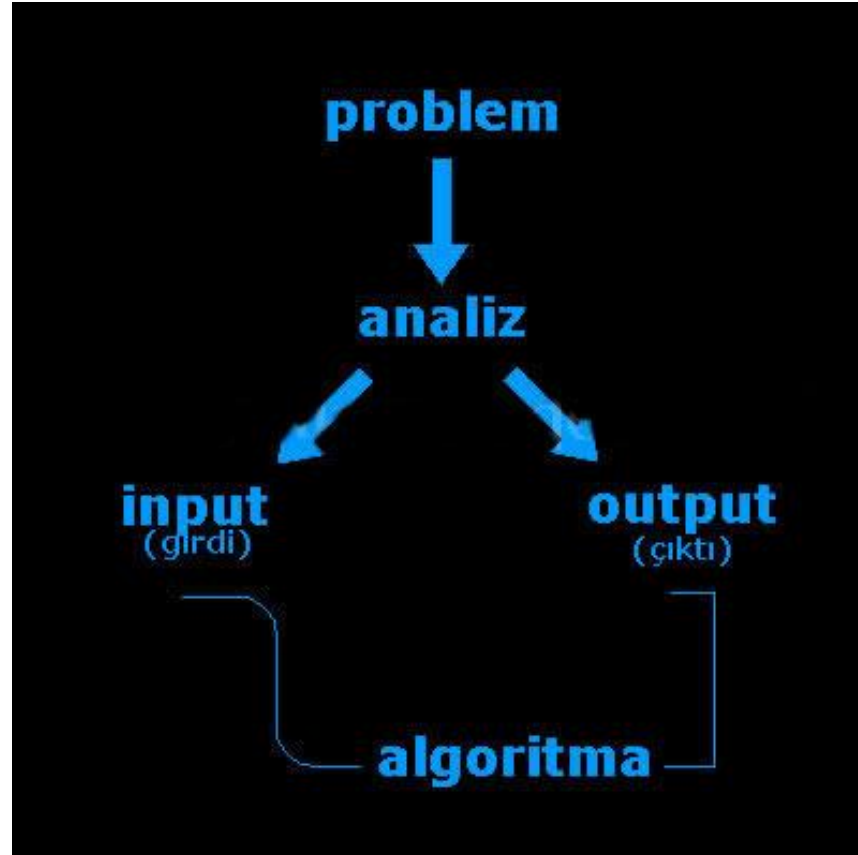
Programlama dili diyoruz ama..

- **Programlama dili**, yazılımcının bir bilgisayara ne yapmasını istediğini anlatmasının tek tiplleştirilmiş yoludur.
- Programlama dili iki önemli kısımda incelenir. **Syntax** ve **Semantics**: Biçim ve Anlam.
- Bilgisayarlar, sadece 0'lar ve 1'ler ile işlem yapabilir. C gibi diğer programlama dilleri bizim bilgisayara daha kolay hükmetmemiz için arayüzlerdir diyebiliriz.

Peki Neden C?

- C derleyicisi taşınabilirdir, pek çok işletim sisteminde çalışabilir.
- İşlem gerçekleştirme konusunda en hızlı dillerdendir.
- Güçlü ve etkili komutlar barındırır.
- Hardware (Donanım) erişimi oldukça kolay ve hızlıdır.

Algoritma



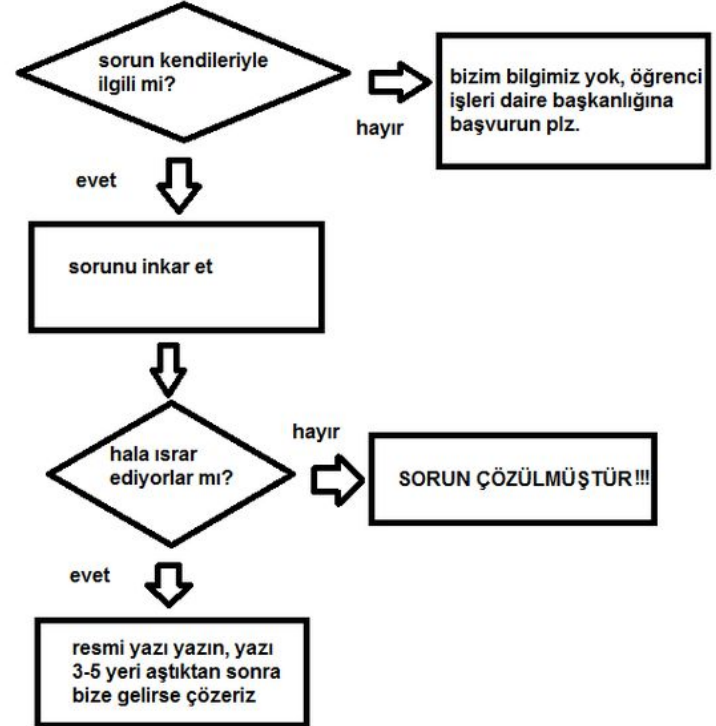
Sözde Kod(Pseudocode)

- Sözde Kod(Pseudocode): Uygulanacak işlemleri anlaşılabilir, belirli bir programlama dili syntaxına bağlı kalmadan yazmadır. Kısacası algoritma tasviridir.

Algoritma

- Algoritma, bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur.
- Algoritma bir programlama dili değildir. (Programlama dillerine yol gösteren bir yöntem dizisidir.)

Bilgi İşlem Algoritması



Algoritma

Algoritmada 3 temel bileşenimiz vardır.

1. Değişkenler: Dışarıdan girilen ve bizim oluşturduğumuz değerleri tutan elemanlardır.

2. Algoritma: Kısaca gerekli adımların mantıksal bir sıra ile yazılmasıdır.

3. Akış Diyagramı: Birbirine oklar gösterilerek algoritmaları ve şemaları birbirine bağlayan kutulara denir.

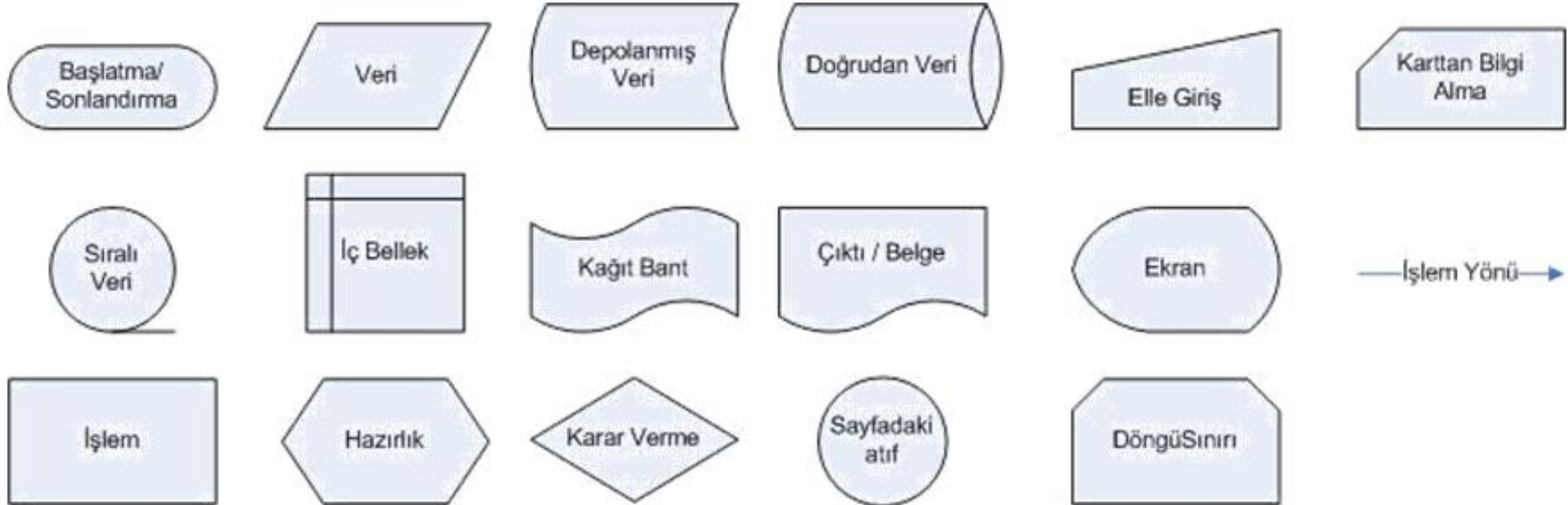
Algoritma

Her algoritma aşağıdaki kriterleri sağlamalıdır.

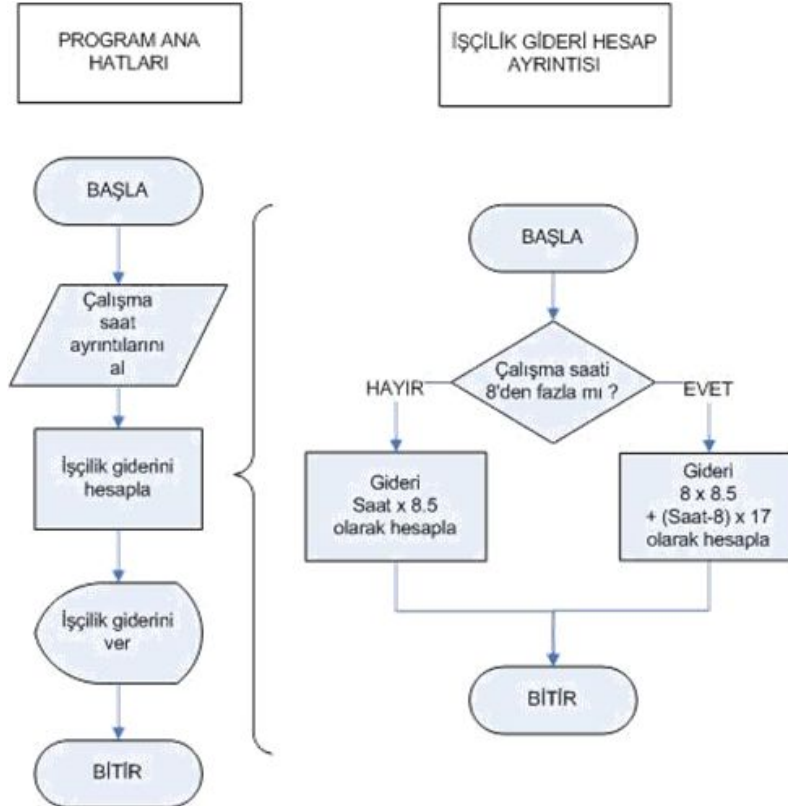
1. **Girdi** : Sıfır veya daha fazla değer dışarıdan verilmeli.
2. **Çıktı** : En azından bir değer üretilmeli.
3. **Açıklık** : Her işlem (komut) açık olmalı ve farklı anlamlar içermemeli.
4. **Sonluluk**: Her türlü olasılık için algoritma sonlu adımda bitmeli.
5. **Etkinlik** : Her komut kişinin kalem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

Algoritma ve Akış Şeması (Flow Chart)

- Bir programlama probleminin çözümü için gerekli ayrıntılı aşamaları grafiksel olarak gösteren bir çizimdir.
- Akım şemalarında her simge belirli bir işlemi ifade etmektedir.



Akış Şeması (Flow Chart)



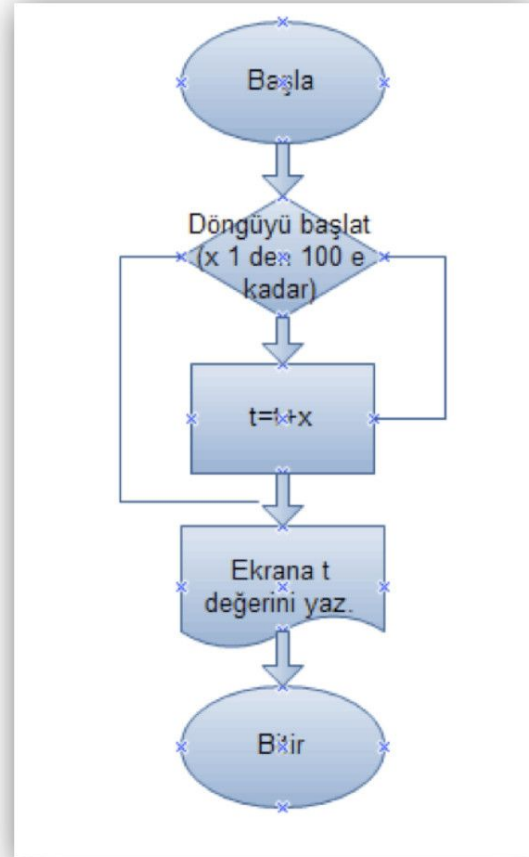
Algoritma

Örnek : 1 den 100 e kadar olan sayıların toplamını bulup sonucu ekrana yazan işlemin algoritmasını ve akış diyagramını tasarlayınız.

Değişkenler

sayıcımız: x

toplam değeri :t



Algoritma

Algoritma

Adım 1: Başla

Adım 2: Döngüyü başlat ($x=1$ den 100 e kadar)

Adım 3: $t=t+x$ işlemini yap.

Adım 4:Döngüyü sonlandır.(x değeri 100 e ulaştığı zaman)

Adım 5: Toplam değerini yaz.

Adım 6: Bitir.

Algoritma Geliştirmek

C dilini ve komutlarını öğrenmek, programlamaya başlamak için şarttır ama algoritma oluşturamadığımız sürece bir program oluşturmazsınız.

Algoritma, mantıktır. Yani neyi, nasıl yapacağınızı belirtir. Algoritma türetmek için geliştirilmiş bir metot yok. Her program için o metodu sizin bulmanız gerekiyor.

Algoritma Geliştirmek

Algoritma, programlamanın bel kemiğidir. Algoritmayı oluşturduğunuz takdirde tüm programlama dillerinde kullanabilirsiniz.

Fakat programlama mantığını bir kere oturtursanız, C komutlarının yerine pekâlâ başka dillere ait komutları da öğrenebilir ve büyük bir zorluk çekmeden diğer dillerde de program yazabilirsiniz.