

# CENG 111 ALGORİTMALAR VE PROGRAMLAMA

Doç. Dr. Tufan TURACI

tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta 3
- 4 Ekim 2022

# 3. Hafta Konular

- **Algoritmalarla ilgili örnekler**
- **Programlamaya giriş**
- **C programlamaya giriş**

**Soru 1:** Klavyeden girilen pozitif bir tamsayının pozitif tamsayı bölenlerinin sayısını ve asal sayı olup olmadığını bulan bir algoritmanın

- Sözde Kodunu yazınız.
- Akış Diyagramını çiziniz. (Ödev)

## Sözde Kod

Sayı: x

Pozitif bölen sayısı: pbs

Sayaç: i

Geçici sayı: k

1. BAŞLA
2. x değerini gir.
3. Eğer (  $x \leq 0$  ) ise pozitif sayı girilmediğini belirt ve Adım 14'e git.
4. Eğer (  $x = 1$  ) ise pbs=1, Adım 12'ye git.
5. pbs=0, i =0
6.  $i = i + 1$
7.  $k = x$
8.  $k = k - i$
9. Eğer (  $k \geq i$  ) ise Adım 8' e git.
10. Eğer (  $k = 0$  ) ise pbs=pbs+1
11. Eğer (  $i < x$  ) ise Adım 6' ya git.
12. Sayının pozitif bölen sayısının pbs olduğunu belirt.
13. Eğer ( pbs=2 ) ise sayının asal sayı olduğunu belirt, aksi halde sayının asal sayı olmadığını belirt.
14. BİTİR.

**Soru 2:** Klavyeden girilen üç basamaklı pozitif bir tamsayının basamak değerlerini bulan algoritmanın

- Sözde Kodunu yazınız.
- Akış Diyagramını çiziniz. (Ödev)

## Sözde Kod

Sayı:  $x$ , Sayının yüzler basamağı:  $a$ , Sayının onlar basamağı:  $b$ , Sayının birler basamağı:  $c$ , Geçici sayı= $k$

1. BAŞLA
2.  $x$  değerini gir.
3. Eğer (  $x \leq 99$  ) veya (  $x \geq 1000$  ) ise üç basamaklı bir sayı girilmediğini belirt ve Adım 14'e git.
4.  $k=x$ ,  $a=0$ ,  $b=0$
5.  $k=k-100$
6.  $a=a+1$
7. Eğer (  $k \geq 100$  ) ise Adım 5' e git.
8. Adım 11' e git.
9.  $k=k-10$
10.  $b=b+1$
11. Eğer (  $k \geq 10$  ) ise Adım 9' a git.
12.  $c=k$
13.  $x$  sayısının yüzler basamağının  $a$ , onlar basamağının  $b$ , birler basamağının  $c$  olduğunu yazdır.
14. BİTİR

# PROGRAMLAMAYA GİRİŞ

- *Program ve Programlama*
- *IDE (Integrated Development Environment, Tümüleşik Geliştirme Ortamı)*
- *Derleyici (Compiler)*
- *Yorumlayıcı (Interpreter)*
- *Bağlayıcı (Linker)*
- *Çalıştırma (Execution)*
- *Hata Türleri*
- *Debug*

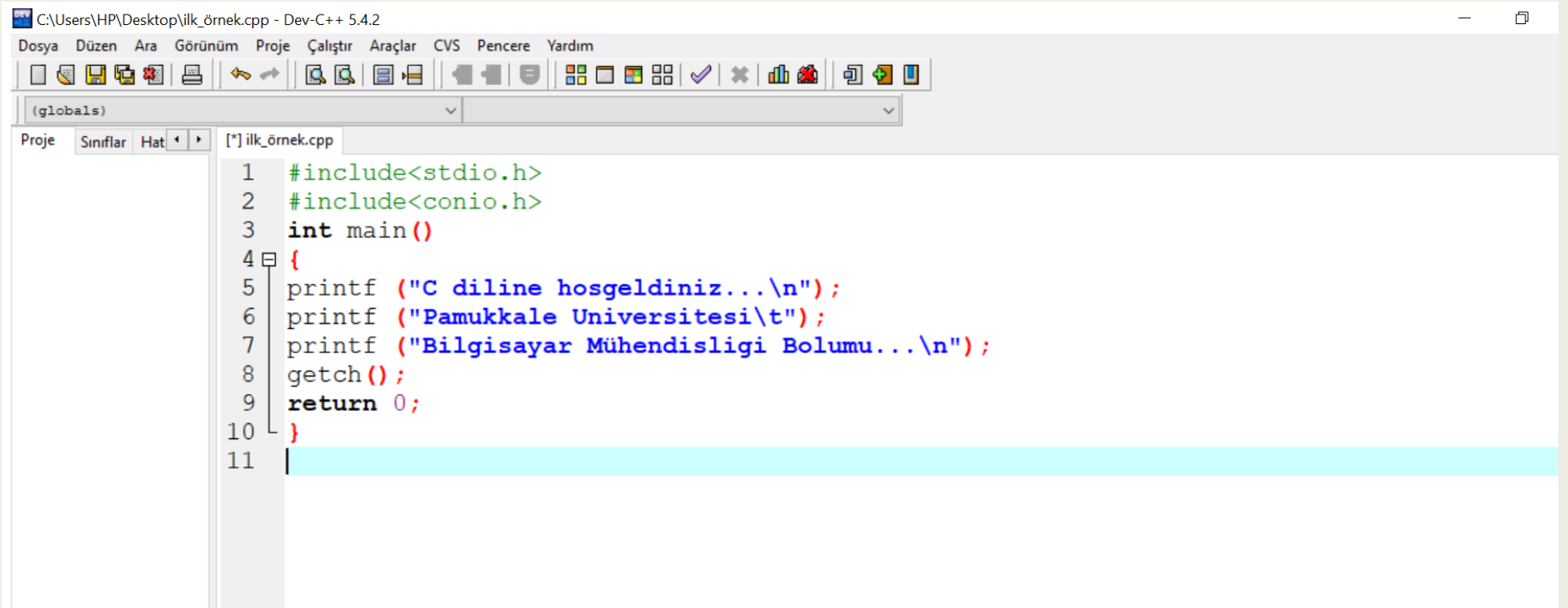
# Program ve Programlama

- Var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlatımların (komutlar, aritmetik işlemler, mantıksal işlemler,...) bütününe program denir.
- Bir programı oluşturabilmek için gerekli komutların belirlenmesi ve uygun biçimde kullanılmasına programlama denir.
- Programlama, bir programlama dili kullanılarak yapılır.
- Yazılan kaynak kodu genellikle bir derleyici ve bağlayıcı yardımıyla belirli bir sistemde çalıştırılabilir hale getirilir. Ayrıca kaynak kodu, bir yorumlayıcı yardımıyla derlemeye gerek duyulmadan satır satır çalıştırılabilir.
- Bir programlama dilini öğrenmekteki tek zorluk programlamanın ne olduğunu öğrenmektir. Bundan sonraki aşamalar daha basittir.



# IDE (Integrated Development Environment – Tümleşik Geliştirme Ortamı)

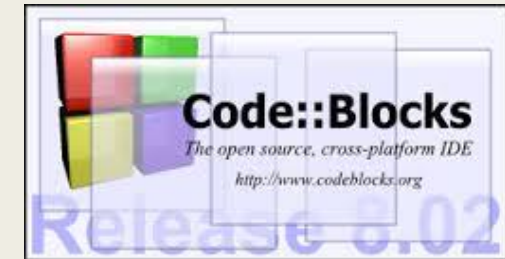
- IDE bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesi için program geliştirme sürecini organize edebilen, geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür.



The screenshot shows a Dev-C++ IDE window titled "C:\Users\HP\Desktop\ilk\_örnek.cpp - Dev-C++ 5.4.2". The menu bar includes "Dosya", "Düzen", "Ara", "Görünüm", "Proje", "Çalıştır", "Araçlar", "CVS", "Pencere", and "Yardım". The toolbar contains various icons for file operations, editing, and execution. The left sidebar shows a "Proje" view with "Sınıflar" and "Hat" tabs. The main editor window displays a C program in "ilk\_örnek.cpp" with the following code:

```
1  #include<stdio.h>
2  #include<conio.h>
3  int main()
4  {
5      printf ("C diline hosgeldiniz...\n");
6      printf ("Pamukkale Universitesi\t");
7      printf ("Bilgisayar Mühendisliği Bolumu...\n");
8      getch();
9      return 0;
10 }
11
```

- En bilinen tümleşik geliştirme ortamları: Microsoft Visual Studio, Dev-C++, Code::Blocks, NetBeans, Eclipse, ...



# Derleyici (Compiler)

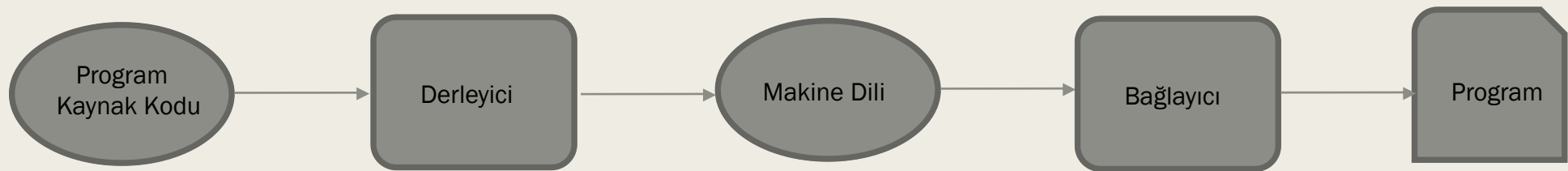
- **Derleyici**, yazılan programın kaynak kodunun mantıksal veya yazımsal hatalar olup olmadığını bulan, bulunan hataları programcıya göstererek programın düzeltilmesine yardım eden, hata yoksa programın çalıştırılması öncesinde kaynak kodu makine diline çeviren bir yazılımdır.

## Yorumlayıcı (Interpreter)

- Yorumlayıcı, kaynak kodu kısımlar kısımlar ele alarak doğrudan çalıştırır.
- Yorumlayıcılar standart bir çalıştırılabilir kod üretmezler.
- Yorumlama işlemi aşama aşama yapılır.
- İlk hatanın bulunduğu yerde programın çalışması kesilir.
- Yorumlayıcılar genelde kaynak koddan, makine diline anlık olarak dönüşüm yaptıkları için, derleyicilere göre daha yavaş çalışırlar.

# Bağlayıcı (Linker) ve Çalıştırma (Execute)

- **Bağlayıcı:** Derleyici tarafından object dosyasına çevrilen bir veya birden çok dosyanın birbirleri ile ilişkilendirmesi ve tek bir çalıştırılabilir dosyaya çevrilmesini sağlayan yazılımdır.
- **Çalıştırma:** Oluşturulan makine dili programının çalıştırılması adımıdır.



# Yazılım Hataları

- Yazılım geliştirme sürecinin herhangi bir aşamasında yapılan programcı hatalarıdır.
- Hiçbir yazılımcı veya programcı isteyerek hata yapmaz.
- Programcı yaptığı yazımsal hataları göremeyebilir.
- Uygulama geliştirme aşamasında hatalar 3 grupta değerlendirilir:
  - *Syntax Error* – Sözdizimi Hataları
  - *Run-time Error* – Çalışma Zamanı Hataları
  - *Logic Error (Bug)* – Mantıksal Hatalar

# Syntax Error

- Yazılan programda programlama dili kurallarından farklı yazımsal durumlardaki hatalardır.
- Bu hataları düzeltmek kolaydır.
- Derleyiciler tarafından hatanın bulunduğu satır rapor edilir.
- IDE'lerde bu sıkıntılar neredeyse yok denecek kadar azdır.

# Run-time Error

- Programın çalıştırılması sırasında karşılaşılan hatalardır. Programcının ele almadığı bir takım aykırı durumlar ortaya çıktığında programın işletim sistemi tarafından kesilmesi ile ortaya çıkar. Bu tip hatalarda hata mesajı çoğunlukla çalışan işletim sisteminin dili ile verilir.
- Eğer bu tip hataları kullanıcı ele almışsa, program programcının vereceği mesajlarla ve uygun şekilde sonlandırılabilir. Bu tip hataların nerelerde ve hangi şartlarda ortaya çıkabileceğini bazen kestirmek zor olabilir.
- Örneğin; olmayan bir dosyayı açmaya çalışmak, var olmayan bir dosyanın üzerine yazmaya çalışmak, olmayan bir bellek kaynağından bellek ayırmaya çalışmak, olmayan bir donanıma ulaşmaya çalışmak gibi hatalardır.
- Sıfıra bölmeye çalışma.



## Logic Error (Bug)

- Bir Programdaki en tehlikeli hatadır.
- Programlamanın mantığında hata vardır. Hata programın kullanımı sırasında görülür.
- Bir problemin yanlış çözülmesi veya ticari bir programda hesap kayıtlarının yanlış tutulması gibi hatalardır.
- Mantıksal hatalar her Programda bulunabilir.
- Günümüzde bir çok yazılım firması mantıksal hataların olduğunu kabul eder ve bununla ilgili olarak programı yenilerler yada yama yazılımlar (Uptade, Patch) yaparlar.

# Debug (Bugdan kurtulma)

- Mantıksal hataları giderebilmek ve yazılımdaki hataları (bug) bulabilmek için yapılan işlemin adıdır.
- Yazılan programın adım adım ve denetim altında çalıştırılmasıdır.
- Programın her adımında ilgili değişkenlerin hangi değere sahip olduğunu görmeyi sağlayarak anormal bir durumu daha kolay izleyip bulmanızı sağlar.

# C PROGRAMLAMA DİLİNE GİRİŞ

# C Tarihçesi

- C, temelde iki eski dile dayanarak, BCPL ve B, 1972 yılında Dennis Ritchie tarafından Bell Laboratuvarlarında geliştirilmiştir.
- C, UNIX'in geliştirilmesinde kullanılmıştır.
- Günümüzdeki tüm modern işletim sistemlerinin yazılmasında kullanılmaktadır.
- Donanımdan bağımsızdır.
- 1970'lerin sonunda, C şu anda geleneksel C olarak bilinen haline geldi.
- Zamanla C'nin yayılması , birbirine benzer ama genellikle uyumsuz, bir çok çeşidinin ortaya çıkmasına sebep oldu.
- 1983 yılında, *American National Standards Committee*'nin bilgisayar ve bilgi işlem komitesi tarafından C'nin sistem bağımsız bir tanımı yapıldı.
- 1989 yılında bu standart onaylandı ve 1999 yılında tekrar gözden geçirildi.

# C Programlama Dili Tercih Nedeni

- C, en popüler dillerden birisidir.
- C, güçlü ve esnek bir dildir. C ile işletim sistemi veya derleyici yazabilir, kelime işlemciler oluşturabilir veya grafikler çizebilirsiniz.
- C, yazılım geliştirme ortamları oldukça fazladır.
- C, özel komut ve veri tipi tanımlamasına izin verir.
- C, taşınabilir bir dildir.
- C, gelişimini tamamlamış ve standardı oluşmuş bir dildir.
- C, yapısal bir dildir. C kodları fonksiyon olarak adlandırılan alt programlardan oluşmuştur.
- C++, Java, JavaScript, JavaApplet, PHP, C# gibi diller C dilinden esinlenmiştir.

# Standart C Kütüphanesi

C programları *fonksiyon* adı verilen parçalardan ya da modüllerden oluşur.

- Bir programcı kendi fonksiyonlarını yazması mümkündür.
  - *Avantaj: Programcı tam olarak nasıl çalıştığını bilir.*
  - *Dezavantaj: Yazmak ve geliştirmek zaman alır.*
- Programcılar genellikle C standart kütüphanesindeki hazır fonksiyonları kullanır.
- Kütüphanedeki fonksiyonları etkin bir şekilde kullanmalıyız.

# Performans

Kendi yazdığınız fonksiyonlar yerine, ANSI standart kütüphanesi fonksiyonlarını kullanmak programın performansını artırır. Çünkü bu fonksiyonlar verimli çalışmaları için özenle yazılmıştır.

# C Programı Geliştirme Ortamının Temelleri

- Tipik olarak bir C programı çalışmadan önce altı safhadan geçer.

- *Yazım(Edit)*
- *Önişleme(Preprocess)*
- *Derleme(Compile)*
- *Bağlama(Link)*
- *Yükleme(Load)*
- *Çalıştırma(Execute)*



# Genel Programlama Hataları

Programlar, her zaman ilk denemede çalıştırılmayabilir. Örneğin; 0' a bölme gibi hatalar programın çalışma anında ortaya çıkar. Bu yüzden, bu hatalara çalışma zamanı hataları (RUNTIME/EXECUTION TIME ERROR) denir. 0'a bölmek ölümcül bir hatadır. Ölümcül hatalar, programın başarılı bir biçimde tamamlanamadan sonlanması anlamına gelir. Ölümcül olmayan hatalar ise programın yanlış sonuçlar verecek biçimde çalışmasına sebep olur.(**Not:** Bazı sistemlerde 0'a bölmek ölümcül hata değildir.)

# C Hakkında Genel Notlar

- Programlarda Açıklık
  - *Bu durum programların okunması, anlaşılması ve geliştirilmesi için gereklidir.*
- C taşınabilir bir dildir.
  - *Programlar pek çok farklı bilgisayarda çalışabilir.*
  - *Taşınırılık en önemli hedeflerden biridir.*


# İyi Programlama

- C programlarınızı basitçe ve doğrudan yazın.
- Programlarınızı gereksiz kullanımları deneyerek uzatmayın.

# Basit C programı-Bir Metni Yazdırmak

```
/* C ile ilk program*/  
#include<stdio.h>  
int main()  
{  
    printf("C diline hosgeldiniz!\n");  
    return 0;  
}
```

Ekran görüntüsü:



C diline hosgeldiniz!

# Basit C programı-Bir Metni Yazdırmak

Satır satır programı incelersek;

1. `/*` ve `*/` arasına yorum yazılır– derleyici görmezden gelir
2. `#include` C önışlemcisine bir emir göndermektedir. Bu satır, önışlemciye standart giriş/çıkış öncü dosyası(`stdio.h`) içeriğinin programa eklenmesini söyler.
3. `int main()` her C programının bir parçasıdır. C programları bir veya birden fazla fonksiyon içerebilir ancak bunlardan biri mutlaka `main` olmalıdır. C’de her program `main` fonksiyonunu çalıştırarak başlar.
4. Küme parantezi, `{` , her fonksiyonun gövdesinin başına yazılır. `}`, küme parantezi ise sonuna yazılmalıdır. Bu iki parantez arasında kalan program parçacığına blok denir.
5. `printf` konsola yazdırma işlemini gerçekleştirir.
6. `Return` ise `main` fonksiyonunu döndüreceği değeri belirtir.

# Basit C programı-Bir Metni Yazdırmak

Ters bölü, çıkış karakteri olarak adlandırılır ve printf'in farklı bir iş yapması gerektiğini belirtir. printf, ters bölü işaretiyle karşılaştığında, bu işaretten sonraki karaktere bakarak bazı özel işlemler yapar.

Çıkış Sırası	Tanım
\n	Yeni satır. İmleci yeni satırın başına geçirir.
\t	Yatay sekme. İmleci bir sonraki sekme başlangıcına taşır.
\a	Alarm.
\\	Ters bölü. printf içindeki ters bölü karakterini yazdırır.
\"	Tırnak.printf içinde tırnak karakterini yazdırır.

## Örnek:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
int main()
{
    setlocale(LC_ALL, "Turkish");
    printf("C dilini Öğreniyorum...\n");
    printf("Pamukkale Üniversitesi\t");
    printf("Bilgisayar Mühendisliği\n");
    printf("\n\"Bilgisayar Mühendisliği\"\n");
    getch ();
    return 0;
}
```

## Ekran Çıktısı:

```
C dilini Öğreniyorum...
Pamukkale Üniversitesi  Bilgisayar Mühendisliği
"Bilgisayar Mühendisliği"
-----
Process exited with return value 0
Press any key to continue . . .
```

# Basit C Programı-İki Sayıyı Toplatmak

```
#include<stdio.h>
#include<conio.h>                                //getch()'i kullana bilmek için
int main()
{
    int tamsayi1,tamsayi2,toplam;                //bildirim
    printf("İlk tamsayiyi giriniz\n");           //mesaj yazdırma
    scanf("%d",&tamsayi1);                       //ilk sayıyı okuma
    printf("İkinci tamsayiyi giriniz\n");        //mesaj yazdırma
    scanf("%d",&tamsayi2);                       //ikinci sayıyı okuma
    toplam=tamsayi1+tamsayi2;                   //toplamın atanması
    printf("Toplam %d dir\n",toplam);           //toplamı yazdırma

    getch();                                    //ekranı bekletme
    return 0;
}
```



# Basit C Programı-İki Sayıyı Toplatmak

- Bildirim tamsayi1, tamsayi2 ve toplam değişkenlerinin int tipinde olduklarını yani tamsayı değerleri tutacağını belirtir.
- scanf kullanıcının gireceği değeri almak için kullanılır.

# Format Belirteçleri

Belirteç	Biçim
%d,%i	Tamsayı(Decimal, Integer)
%u	İşaretsiz Tamsayı(Unsigned)
%f	Kayan Noktalı Sayı(Float)
%c	Karakter(Char)
%O	8 Tabanında Sayı(Octal)
%x,%X	16 Tabanında Sayı(hexadecimal)
%e	Üssel Gösterim(Exponential)
%s	Karakter Dizisi(String)
l,h	Long ve short ön eki

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
int main()
```

```
{int x;
```

```
printf("Bir tamsayi girin.\n");
```

```
scanf("%d", &x);
```

```
printf("Onluk tabanda = \t%d\n", x);
```

```
printf("Sekizlik tabanda = \t%o\n", x);
```

```
printf("Onaltilik tabanda = \t%X\n", x);
```

```
getch(); return 0;
```

```
}
```

```
Bir tamsayi girin.  
29  
Onluk tabanda =          29  
Sekizlik tabanda =       35  
Onaltilik tabanda =      1D
```

```
#include<stdio.h>
#include<conio.h>
int main()
{int x=1,X;
float y=.1;
double Y=2;
printf("x= %d , X= %d\n",x,X);
printf("y= %f , Y= %e\n",y,Y);
getch();
return 0;
}
```

```
x= 1 , X= 2686792
y= 0.100000 , Y= 2.000000e+000
```

# Aritmetik Operatörler

Operatör	Açıklama	Ornek	Anlami
+	Toplama	$x+y$	x ve y nin toplamı
-	Çıkarma	$x-y$	x ve y nin farkı
*	Çarpma	$x*y$	x ve y nin çarpımı
/	Bölme	$x/y$	x ve y nin oranı
%	Mod	$x\%y$	$x/y$ den kalan sayı

Operatör	İşlem	Öncelik
()	Parantez	İlk
*,/,%	Çarpım, Bölüm ve Mod alma	İkinci
+, -	Toplama, Çıkarma	Son

# Aritmetik Operatörler

```
int x, y = -2, z = 12;
```

```
x = y = z;           // hepsi 12 değerini alır
```

```
y = x % 5;           // y 2 değerini alır
```

```
x = z % y;           // x 0 değerini alır
```

```
x = 12/5;             // x 2 değerini alır (tamsayı bölme)
```

```
float f = 12/5;       // f 2.0 değerini alır
```

# Program:

```
#include<stdio.h>
#include<conio.h>
int main()
{ int x, y = -2, z = 12;
printf("x degeri: %d\n",x);
printf("y degeri: %d\n",y);
printf("z degeri: %d\n",z);
x = y = z; // hepsi 12 değerini alır
printf("\n");
printf("x degeri: %d\n",x);
printf("y degeri: %d\n",y);
printf("z degeri: %d\n",z);
y = x % 5; // y 2 değerini alır
```

```
printf("\n");
printf("y degeri: %d\n",y);
x = z % y; // x 0 değerini alır
printf("\n");
printf("x degeri: %d\n",x);
x = 12/5; // x 2 değerini alır (tamsayılı bölme)
printf("\n");
printf("x degeri: %d\n",x);
float f = 12/5;
printf("\n");
printf("f degeri: %f\n",f);
f = 12.0/5;
printf("\n");
printf("f degeri: %f\n",f);
getch();
return 0;
}
```

## Ekran Çıktısı:

```
x degeri: 0
y degeri: -2
z degeri: 12

x degeri: 12
y degeri: 12
z degeri: 12

y degeri: 2

x degeri: 0

x degeri: 2

f degeri: 2.000000

f degeri: 2.400000

-----
Process exited with return value 0
Press any key to continue . . .
```



# Kaynaklar

- C: How to Program Third Edition Harvey M. Deitel ; Paul J. Deitel.
- C Programlama Dili Dr. Rıfat Çölkesen Papatya Yayıncılık.
- Problem Solving and Program Design in C, 7/E Jeri R. Hanly; Elliot B. Koffman.
- C Programlama dili; İbrahim Güney; Nobel Yayıncılık.
- Algoritma Geliştirme ve Programlamaya Giriş, Fahri Vatansever, Seçkin yayıncılık
- C Programlama Ders Notları, A. Kadir YALDIR, Pamukkale Üniversitesi ders notları.