

Programlama Dilleri ve C:

Bilgisayarda bir ya da bir çok işlemi yaptırmak için, işlemlere karşılık gelen komutlar topluluğuna **bilgisayar programı** denir. Bilgisayar programı yazmaya yarayan bir çok programlama dili vardır. Bunlardan bazıları, Basic, GwBasic, Visual Basic, Quick Basic, Fortran, Cobol, Pascal, Delphi, C, C++, Visual C, ... gibi.

C Dilinde Program Yazma ve Derleme:

Her hangi bir dilde olduğu gibi C dilinde de, önce dilin kuralları çerçevesinde, bir kelime işlemci yardımıyla (örneğin DOS ortamında Edit, Windows'ta Notepad ile) programın kaynak kod dosyası yazılır, diske kaydedilir (dosya kaydedilirken dilin cinsine göre uygun bir uzantı verilir; örneğin Basic'te uzantı bas, Pascal'da pas, C de c dir) ve programlama dilinin derleyicisi ile (örneğin Pascal için tpc.exe, C için tcc.exe) derlenir, hatalar varsa kelime işlemci yardımıyla düzeltilir ve tekrar derlenir. Böylece kendi kendine çalışabilir (uzantısı exe olan) bir dosya elde edilir. Ancak Turbo C program paketinin tc.exe program, hem kaynak kod yazmaya hem de derlemeye ve programın hatalarından arındırmaya yarayan tümleşik bir ortamdır.

Haydi C de Basit Bir Program Yazıp Derleyelim:

Örneğin; ekrana "İzmir Fen Lisesi" yazdıran bir program yazıp derleyelim.

Çözüm:

tc.exe programını çalıştırdığımızda gelen ortamda aşağıdaki kodları yazalım ve Alt-F tuşlarına basarak File menüsü altındaki Save seçeneğinden örneğin ifl adıyla kaydedelim (Turbo C otomatik olarak uzantısını c olarak kaydeder).

```
File Edit Run Compile Project Options Debug Break/w
Edit
Line 2 Col 34 Insert Indent Tab Fill Unindent * D:IFL.C
#include<stdio.h>
main() {printf("İzmir Fen Lisesi");}
```

```
#include<stdio.h>
main() {printf("İzmir Fen Lisesi");}
```

Şimdi de Alt-C tuşlarına basarak programımızı derleyelim. Hatamız yoksa program çalışmaya hazır demektir. Alt-R tuşlarına basarak programımızı çalıştırırız. Ancak program çalışır çalışmaz hemen tümleşik ortama geri döner. Programın çalışması sırasında ne yazdığını görmek istersek, DOS ortamına geçici olarak geçebiliriz. Bunun için Alt-F-OS shell seçeneği ile DOS ortamına geçtiğimizde ekranda programın çıktısı olan **İzmir Fen Lisesi** yazısını görürüz. Tekrar TC nin tümleşik ortamına dönmek için **exit** komutunu yazıp (Enter) tuşuna basmalıyız.

Diskimize dikkatlice baktığımızda adı ifl olan bir kaç dosya görürüz: ifl.c, ifl.bak, ifl.obj, ifl.exe. İşte en sonucusu olan dosya DOS ortamında bağımsız olarak çalışabilen ifl.exe dosyasıdır. DOS ortamında iken ifl.exe (veya sadece ifl) yazıp (Enter) tuşuna basarsak veya Windows ortamında fare ile ifl.exe üstünü çift tıklatarsak ekranda **İzmir Fen Lisesi** yazısını görürüz.

C de Standart Veri Tipleri:

Tip	Uzunluk	Sınırı
unsigned char	1 byte	0 : 255
char	1 byte	-128 : 127
unsigned int	2 byte	0 : 65535
short int	2 byte	-32768 : 32767
int	2 byte	-32768 : 32767
unsigned long	4 byte	0 : 4 294 967 295
long	4 byte	-2 147 483 648 : 2 147 483 647
float	4 byte	-3,4.10 ⁻³⁸ : 3,4.10 ³⁸
double	8 byte	-1,7.10 ⁻³⁰⁸ : 1,7.10 ³⁰⁸
long double	10 byte	-3,4.10 ⁻⁴⁹³² : 3,4.10 ⁴⁰⁹²

Giriş Çıkış Fonksiyonları:

C de bilgisayara giriş ve bilgisayardan çıkış için kullanılan fonksiyonlara denir. En çok kullanılan fonksiyonlardan birincisi **printf()**; çıkış için, ikincisi **scanf()**; giriş için kullanılır.

printf() fonksiyonu:

Fonksiyonun yapısı aşağıdaki gibidir:

printf("çıkış biçimi karakterleri",değerler listesi);

Buradaki "**çıkış biçimi karakterleri**" şunlar olabilir:

a) Açıklama karakterleri:

Olduğu gibi ekranda görünmesini istediğimiz karakterlerdir.

b) Escape düzeni karakterleri:

Kürsörün yerini ve bazı özel karakterleri yazdırmaya yarar. Bu karakterler \ ile başlar. Aşağıdaki listede bu karakterler ve anlamları belirtilmiştir.

Karakter	Anlamı
\n (new line)	Kürsörü alt satıra geçirir
\r (return)	Kürsörü satır başına geçirir
\b (back space)	Kürsörü bir önceki sütuna çeker
\a (alarm, bell)	Bip sesi vermeye yarar
\t (tab)	Kürsörü belirli bir miktar sağa kaydırır
\\	\ koymaya yarar
\'	' koymaya yarar
\"	" koymaya yarar
\?	? koymaya yarar
\0	NULL karakteri

c) Çıkış biçimi belirten karakterler:

Değerler listesi bölümündeki değişkenlerin değerlerinin ekrana yazılış biçimlerini belirten ifadelerdir. Bu ifadeler % ile başlar.

Aşağıdaki listede bu karakterler ve anlamları belirtilmiştir.

Karakter	Anlamı
%d	İşaretsiz tamsayı
%ld	Uzun tamsayı
%f	Ondalık (reel) sayı
%lf	Double (uzun) ondalıklı sayı
%e	Exponentiel (üstel) sayı
%x	Hexadecimal sayı

C Ders Notları

%c	Tek karakter
%s	String (karakter grubu)

Değerler listesi:

Değerler listesine, ekrana yazdırılacak değişkenler veya işlemler yazılır.

scanf() fonksiyonu:

Tanımlanmış değişkenlerin temsil ettiği bellek bölgelerine değer okutmak amacıyla kullanılır.

Fonksiyonun yapısı aşağıdaki gibidir:

scanf("giriş biçimi karakterleri",adresler listesi);

Giriş biçimi karakterleri:

Buradaki giriş biçimi karakterleri, printf() fonksiyonundaki çıkış biçimi karakterlerine karşılık gelir.

Adresler listesi:

Bellekte atama yapılacak değişkenlerin adreslerini belirtmeye yarar. Bunun için değişken adının önüne & konur.

Örnek:

Aşağıdaki alt programlar sonucu ekran çıktısını bulalım.

- printf("İzmir"); printf("Fen"); printf("Lisesi");
- printf("\nİzmir"); printf("\nFen"); printf("\tLisesi");

Çözüm:

- İzmirFenLisesi
- İzmir
Fen Lisesi

Örnek:

Girilen iki tamsayının toplamını yazdıran bir program yazalım.

Çözüm: (cprog01.c)

Öncelikle standart giriş (input) çıkış (output) fonksiyonlarının tanımlı olduğu header dosyasını (**stdio.h**) programa eklemeliyiz. Bunu **#include** deyimi yardımıyla; **#include<stdio.h>** veya **#include "stdio.h"** ile yaparız. Sonra, programda kullanılacak değişken tanımları yapılır. Değişkenlerimiz a ve b gibi iki tamsayı olduğunu varsayalım; bu tanımlama aşağıdaki gibi yapılır.

int a,b;

Artık ana fonksiyonu (ana programı) yazabiliriz.

main()

```
{printf("ilk sayınız");scanf("%d",&a);  
printf("ikinci sayınız");scanf("%d",&b);  
printf("%d",a+b);}
```

Buna göre programın tamamı şöyle olmalıdır:

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

```
int a,b;
```

```
main()
```

```
{printf("ilk sayınız");scanf("%d",&a);  
printf("ikinci sayınız");scanf("%d",&b);  
printf("%d",a+b);}
```

Not:1) Programın sonucunda iki sayının toplamının bulunduğunu belirtmek için; **printf("Toplam=%d",a+b);** komutu yazıldığında sonuç daha anlaşılır hale gelir. Örneğin ilk sayı 67, ikinci sayı 13 girildiyse; programın çıktısı; **Toplam=80** gibi daha anlamlı biçimde görünür.

Not 2) Girilen sayıları ve toplamı yazdırmak istersek bunu; **printf("%d + %d = %d dir",a,b,a+b);** biçiminde yazarız. Örneğin ilk sayı 67 ikinci sayı 13 girildiyse ekranda; **67 + 13 = 80 dir.** görülür.

Not 3) Programın her çalıştırılmasında ekranda, daha önceden kalan görüntünün silinmesini istersek, programda **clrscr();** fonksiyonunu kullanırız.

Not 4) Programın çalışması bittiğinde, hemen tümleşik ortama dönülür.Bu durumda ekrandaki görüntüyü tam göremeyiz.Şayet programın çalışmasını bir tuşa basana kadar bekletmek istersek; bunu **getch();** fonksiyonu yardımıyla yaparız.

Buna göre programın son durumu şöyle olmalıdır:

```
#include<stdio.h>  
int a,b;  
main()  
{clrscr();printf("ilk sayınız");scanf("%d",&a);  
printf("ikinci sayınız");scanf("%d",&b);  
printf("%d + %d = %d dir",a,b,a+b);getch();}
```

C de Atama ve Matematiksel İşlemler:

Diğer programlama dillerinde olduğu gibi C'de de bir değişkene değer ataması ve bir çok işlem (operatör) vardır.

C de değer atama;

değişken=atanan_değer; biçiminde yapılır.

Şayet aynı değere birden fazla değişkene atanacaksa bunu;

değ1=değ2=değ3=...=değn=atanan_değer; biçiminde yapılır.

Atanan değer sabit olabileceği gibi, başka değişkenlerin ürettiği bir değer de olabilir.

Örnek:

a) a=3;b=7;c=a+b;

b) a=3;b=7;a=a+5;c=a+b;

c) a=3;b=7;a=a-1;b=a+b-2;c=d=a*b+b;d=d-1;

atamalarında a, b, c ve d değişkenlerinin alacağı son değerleri bulalım.

Çözüm:

a) a=3, b=7, c=10

b) a=8, b=7, c=15

c) a=2, b=9, c=27, d=26

C de Arttırma Azaltma ve Sıkıştırılmış Atama:

değişken=değişken+değer; ifadesi ile **değişken** in değeri **değer** kadar arttırılır.

değişken=değişken-değer; ifadesi ile **değişken** in değeri **değer** kadar azaltılır.

C Ders Notları

Bu iki atama sırasıyla **değişken+=değer**; ve **değişken-=değer**; ile de yapılabilir ki bu tür atamalara **sıkıştırılmış atama** denir.

Şayet özel olarak, değişkenin değeri 1 arttırılacaksa bunu; ister **değişken=değişken+1**; ifadesi ile yapabileceğimiz gibi istersek **değişken++**; ifadesi ile de yapabiliriz.

Benzer biçimde değişkenin değeri 1 azaltılacaksa bunu; ister **değişken=değişken-1**; ifadesi ile yapabileceğimiz gibi istersek **değişken--**; ifadesi ile de yapabiliriz.

Matematiksel İşlemler:

Aşağıdaki tabloda, matematiksel işlemler ve C deki karşılıkları ile anlamları verilmiştir.

İşlem	Anlamı
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Modül
++	Bir arttırma
--	Bir eksiltme

Örnek:

a) `a=5;a=a+1;printf("a=%d",a);`
b) `a=5;a++;printf("a=%d",a);`
c) `a=5;b=-13;a++;b--;a=a*b;b=a*b;`
`printf("a=%d b=%d",a,b);`
program parçalarının çalışması sonucu ekrandaki görüntüleri bulalım.

Çözüm:

a) a=6
b) a=6
c) a=20 b=-280

Örnek:

Aşağıdaki matematiksel ifadelerin C de karşılık gelen deyimlerini yazınız.

a) $x=a+b$ b) $x=a^2+b$ c) $y=\frac{2x-1}{x+3}$ d) $x_1 = \frac{-b-\sqrt{b^2-4ac}}{2a}$

Çözüm:

a) `x=a+b;` b) `x=a*a+b;` c) `y=(2*x-1)/(x+3);`
d) `x1=(-b-sqrt(b*b-4*a*c))/(2*a);`

Karşılaştırma Operatörleri:

İki değişkenden birinin diğerinden küçük olup olmadığını, eşit olup olmadığını vs. yarayan operatörlere **karşılaştırma operatörleri** denir. Bu operatörler ve anlamları aşağıda verilmiştir.

Operatör	Anlamı
==	eşittir
!=	eşit değildir
<	küçüktür
>	büyüktür
<=	küçük veya eşittir
>=	büyük veya eşittir

Örnek:

Bir programda; a, b ve c tam sayı değişkenlerinin tanımlandığını ve a ya 7, b ye -5 değerlerinin atandığını varsayalım. Buna göre aşağıdaki değişkenlerin versayalım. Buna göre görüntüleri bulalım.

a) `c=(a>b); printf("a=%d b=%d c=%d",a,b,c);`
b) `c=(a<=b);printf("a=%d b=%d c=%d",a,b,c);`
c) `c=(a!=b);printf("a=%d b=%d c=%d",a,b,c);`
d) `c=(a==(2-b));printf("a=%d b=%d c=%d",a,b,c);`

Çözüm:

a) a nın değeri olan 7, b nin değeri olan -5 ten büyük yani doğru (TRUE) olduğundan, c nin değeri 1 dir. Buna göre ekranda **a=7 b=-5 c=1** görülür.

b) Benzer biçimde 7 -5 ten küçük veya eşit olmadığından yani yanlış (FALSE) olduğundan c nin değeri 0 dir. Buna göre ekranda **a=7 b=-5 c=0** görülür.

c) Benzer biçimde ekranda; **a=7 b=-5 c=1** görülür.

d) Ekranda **a=7 b=-5 c=1** görülür.

C de Kontrol Deyimleri:

1) if şart deyimi:

Yazım şekli; **if(şart ifadesi) {blok;}**

Burada **şart ifadesi** doğru yani değeri TRUE=1 ise **blok**; icra edilir, yanlış yani değeri FALSE=0 ise **blok**; icra edilmez.

Not: Şayet **blok**; birden fazla fonksiyondan oluşuyorsa { } (küme parantezleri arasına) alınmalıdır. Ancak bir fonksiyondan oluşuyorsa küme parantezleri arsına alınmasa da olur.

Örnek: 100 üzerinden girilen bir puan için; 45 ten küçükse **Kaldı** 45 veya daha büyükse **Geçti** yazdıran bir program yazalım.

Çözüm: (cprog02.c)

```
#include<stdio.h>
int puan;
main() {clrscr();
printf("Puanı giriniz ");scanf("%d",&puan);
if(puan<45) printf("Kaldı");
if(puan>=45) printf("Geçti");getch();}
```

2) if-else şart deyimi:

Yazım şekli; **if(şart ifadesi) {blok1;} else {blok2;}**

Örnek:

100 üzerinden girilen bir puan için; 45 ten küçükse **Kaldı** 45 veya daha büyükse **Geçti** yazdıran bir programı if-else deyimi ile yazalım.

Çözüm: (cprog03.c)

```
#include<stdio.h>
int puan;
```

C Ders Notları

```
main() {clrscr();
printf("Puanı giriniz ");scanf("%d",&puan);
if(puan<45) printf("Kaldı"); else printf("Geçti");getch();}
```

Örnek:

100 üzerinden girilen bir puan için;
45 ten küçükse **Başarısız**;
45-54 arası ise **Geçer**;
55-69 arası ise **Orta**;
70-84 arası ise **İyi**;
85-100 arası ise **Pekiyi** yazdıran bir programı if-else deyimi ile yazalım.

Çözüm: (cprog04.c)

```
#include<stdio.h>
int puan;
main() {clrscr();
printf("Puanı giriniz ");scanf("%d",&puan);
if(puan<45) printf("Başarısız");
else if(puan<55) printf("Geçer");
else if(puan<70) printf("Orta");
else if(puan<85) printf("İyi");
else if(puan<=100) printf("Pekiyi");
else printf("Lütfen (1-100 ) arası bir puan giriniz...");getch();}
```

Örnek:

Klavyeden girilen iki reel sayı için;
+ karakterine basıldığında toplamını yazdıran;
- karakterine basıldığında farkını yazdıran;
x karakterine basıldığında çarpımını yazdıran;
/ karakterine basıldığında bölümünü yazdıran bir programı if-else deyimi kullanarak yazalım.

Çözüm: (cprog05.c)

```
#include<stdio.h>
float x,y,sonuc;char kar;
main() {clrscr();
printf("\n1. sayı = ");scanf("%f",&x);
printf("\n2. sayı = ");scanf("%f",&y);
printf("\nişlemi seçiniz [ + - x / ] ");scanf("%s",&kar);
if(kar=='+')
{sonuc=x+y;printf("%f %c %f = %f",x,kar,y,sonuc);}
else if(kar=='-')
{sonuc=x-y;printf("%f %c %f = %f",x,kar,y,sonuc);}
else if(kar=='x')
{sonuc=x*y;printf("%f %c %f = %f",x,kar,y,sonuc);}
else if(kar=='/')
{sonuc=x/y;printf("%f %c %f = %f",x,kar,y,sonuc);}
else printf("Lütfen [ + - x / ] işlemlerinden birini seçiniz...");getch();}
```

Not 1) Sonuçlar yazdırılırken ondalık kısmı 6 haneli olarak görünür.Ondalık kısmının uzunluğunu istersek ayarlayabiliriz. Örneğin ondalık kısımların uzunluğunu 3 hane olmasını istersek bunu;
printf("%.3f %c %.3f = %.3f",x,kar,y,sonuc);
olarak değiştirilmelidir.

Not 2) Programda sonuçları, her else-if bölümünde ayrı ayrı printf fonksiyonu ile yazdıracağımıza, programın sonunda tek bir printf ile de

yazdırabiliriz.Söylediklerimizin uygulaması olan program aşağıdaki gibi olacaktır.

(cprog06.c)

```
#include<stdio.h>
float x,y,sonuc;char kar;
main() {clrscr();
printf("\n1. sayı = ");scanf("%f",&x);
printf("\n2. sayı = ");scanf("%f",&y);
printf("\nişlemi seçiniz [ + - x / ] ");scanf("%s",&kar);
if(kar=='+') sonuc=x+y;
else if(kar=='-') sonuc=x-y;
else if(kar=='x') sonuc=x*y;
else if(kar=='/') sonuc=x/y;
else
{printf("Lütfen [ + - x / ] den birini seçiniz...");exit();}
printf("%.3f %c %.3f = %.3f",x,kar,y,sonuc);getch();}
```

3) switch deyimi:

Yazım şekli;
switch(kontrol ifadesi)
{case değer1:blok1;break;
case değer2:blok2;break;
...
case değern:blokn;break;
default:blok;}

Örnek:

Klavyeden [1-5] arasında bir sayı girildiğinde;
1 için ekrana A, 2 için ekrana B, 3 için ekrana C, 4 için ekrana D, 5 için ekrana E yazan, bu değerler dışında bir değer girildiğinde, **Lütfen [1-5] arasında bir sayı giriniz!** uyarısı yazdıran bir programı switch deyimi kullanarak yazalım.

Çözüm: (cprog07.c)

```
#include<stdio.h>
int sayi;
main() {clrscr();printf("[1-5] arası bir sayı giriniz");
scanf("%d",&sayi);switch(sayi)
{case 1:printf("\nA");break;
case 2:printf("\nB");break;
case 3:printf("\nC");break;
case 4:printf("\nD");break;
case 5:printf("\nE");break;
default: printf("\nLütfen [1-5] arası bir sayı giriniz");}
getch();}
```

Örnek:

Klavyeden girilen iki reel sayı için;
+ karakterine basıldığında toplamını yazdıran;
- karakterine basıldığında farkını yazdıran;
x karakterine basıldığında çarpımını yazdıran;
/ karakterine basıldığında bölümünü yazdıran;
bir program switch deyimi kullanarak yazalım.

Çözüm: (cprog08.c)

```
#include<stdio.h>
float x,y,sonuc;char kar;
main() {clrscr();
printf("\n1. sayı = ");scanf("%f",&x);
printf("\n2. sayı = ");scanf("%f",&y);
```

C Ders Notları

```
printf("\nİşlemi seçiniz [ + - x / ]");kar=getch();
switch(kar)
{case '+':{sonuc=x+y;break;}
case '-':{sonuc=x-y;break;}
case 'x':{sonuc=x*y;break;}
case '/':{sonuc=x/y;break;}
default:{printf("Lütfen [ + - x / ] den birini seçiniz...");
getch();exit();}}
printf("%f %c %f = %f",x,kar,y,sonuc);getch();}
```

C de Tekrarlı İşlemler:

Her hangi bir programlama dilinde olduğu gibi C de de tekrarlı işlemler için **döngüler** kullanılır.

1) For Döngüsü:

Yazım şekli:

```
for(deyim1;şartlı önerme;deyim2) {blok;}
```

Örnek:

Temiz ekranda alt alta 10 tane **İzmir Fen Lisesi** yazdıran bir program yazalım.

Çözüm: (cprog09.c)

```
#include<stdio.h>
int i;
main()
{clrscr();for(i=1;i<=10;i++) printf("\nİzmir Fen Lisesi");}
```

Örnek:

1 den istenen bir tam sayıya kadar olan sayıların;

- Toplamını,
- Kareleri toplamını,
- Sayıların çarpmaya göre tersleri toplamını bulduran bir programı **for döngüsü** kullanarak yazalım.

Çözüm:

a) (cprog10.c)

```
#include<stdio.h>
int i,son;long t;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%d",&son);t=0;
for(i=1;i<=son;i++) {t=t+i;}
printf("1 den %d ye kadar sayıların toplamı; %ld
dir",son,t);getch();}
```

b) (cprog11.c)

```
#include<stdio.h>
int i,son;long t;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%d",&son);t=0;
for(i=1;i<=son;i++) {t=t+i*i;}
printf("1 den %d ye kadar sayıların kareleri toplamı; %ld
dir",son,t);getch();}
```

c) (cprog12.c)

Bu programda doğal sayıların çarpmaya göre tersleri ondalıklı reel sayılar olacağından, değişkenleri **float** veya **double** tipinde seçmeliyiz.

```
#include<stdio.h>
double i,son,t=0;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%lf",&son);t=0;
for(i=1;i<=son;i++) {t=t+1/i;}
printf("1 den %lf ye kadar sayıların çarpmaya göre
tersleri toplamı; %lf dir",son,t);getch();}
```

Not: Programın sonundaki;

```
printf("1 den %lf ye kadar sayıların çarpmaya göre
tersleri toplamı; %lf dir",son,t);
```

yerine;

```
printf("1 + 1/2 + ... + 1/%.0lf = %lf",son,t);
```

yazılırsa sonuç, daha estetik ve rahat anlaşılır bir biçimde görünecektir.Örneğin program çalıştırıldığına 5 değeri girilirse ekranda;

1 + 1/2 +... +1/5 = 2.28333 görünür.

2) while Döngüsü:

Yazım şekli:

```
while(şartlı önerme) {blok;}
```

Şartlı önerme doğru oldukça blok icra edilir.

Örnek:

1 den istenen bir tam sayıya kadar olan sayıların;

- Toplamını,
- Kareleri toplamını,
- Sayıların çarpmaya göre tersleri toplamını bulduran bir programı **while döngüsü** kullanarak yazalım.

Çözüm:

a) (cprog13.c)

```
#include<stdio.h>
int i,son;long t;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%d",&son);t=0;
while(i<=son){t=t+i;i++;}
printf("1 + 2 + ... + %d = %ld",son,t);getch();}
```

b) (cprog14.c)

```
#include<stdio.h>
int i,son;long t;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%d",&son);t=0;
while(i<=son){t=t+i*i;i++;}
printf("1^2 + 2^2 + ... + %d^2 = %ld",son,t);getch();}
```

c) (cprog15.c)

Bu programda doğal sayıların çarpmaya göre tersleri ondalıklı reel sayılar olacağından, değişkenleri **float** veya **double** tipinde seçmeliyiz.

```
#include<stdio.h>
double i=1,son,t=0;
main()
{clrscr();printf("Kaça kadar toplansın? ");
scanf("%lf",&son);t=0;
while(i<=son) {t=t+1/i;i++;}
printf("1 + 1/2 + ... + 1/%.0lf = %.10lf",son,t);getch();}
```

Örnek:

1.2+2.3+3.4+ ... toplamı için;
a) İstenen sayıda terim toplamını;
b) Toplamı girilen bir sayıya eşit ya da daha büyük olduğu değere kadar hesaplatıp ayrıca kaç terim toplandığını da bulduran bir program yazalım.

Çözüm:

a) (cprog16.c)

```
#include<stdio.h>
long i=1,n,t=0;
main()
{clrscr();printf("Kaç terim toplansın? ");scanf("%ld",&n);
while(i<=n) {t=t+i*(i+1);i++;}
printf("1.2+2.3+ ... +%ld.%ld = %ld",n,n+1,t);getch();}
```

b) (cprog17.c)

```
#include<stdio.h>
long i=1,n,t=0;
main()
{clrscr();printf("Toplam kaç kadar olsun? ");
scanf("%ld",&n);
while(t<=n) {t=t+i*(i+1);i++;}
printf("%ld terim toplamı %ld",i-1,t);
printf("\nYani 1.2+2.3+ ... +%ld.%ld = %ld",i-1,i,t);
getch();}
```

3) do-while Döngüsü:

Yazım şekli:

do {blok;} while(şartlı önerme);

Bu döngüde önce **blok** icra edilir, sonra **şartlı önerme** test edilir, doğru oldukça blok icra edilir.

Örnek:

Ekrana yan yana 5 tane * bastırarak bir programı;
a) for döngüsü ile;
b) while döngüsü ile;
c) while-do döngüsü ile yazalım.

Çözüm:

a) (cprog18.c)

```
#include<stdio.h>
int i;
main()
{clrscr();for(i=1;i<=5;i++) printf("*");getch();}
```

b) (cprog19.c)

```
#include<stdio.h>
int i=1;
main()
{clrscr();while(i<=5) {printf("*");i++;}getch();}
```

c) (cprog20.c)

```
#include<stdio.h>
int i=1;
main()
{clrscr();do {printf("*");i++;} while(i<=5); getch();}
```

Örnek:

Kaç kişi olduğu belli olmayan bir grubun, panları klavyeden girilerek not ortalaması buldurmaya yönelik bir program yazalım.

Not: Her puan girişinde **Devam mı? (E/H)** sorusu gelecek E veya e girildiğinde toplam almaya devam edilecek aksi halde ortalama bulunacak ve ekrana aşağıdaki örneğe benzer biçimde yazdırılacak:

Örnek çıktı:

13 kişinin not ortalaması = 78.345 tir.

Çözüm:

a) (cprog21.c)

```
#include<stdio.h>
int i;float puan,toplam;char cevap;
main()
{i=1;toplam=0;clrscr();
do {printf("\n%d. puan ",i);
scanf("%f",&puan);toplam=toplam+puan;i++;
printf("\nDevam mı? (E/H)"); cevap=getch();}
while(cevap=='E' ||cevap=='e');
printf("%d kişinin not ortalaması = %.3f tir."
,i-1,toplam/(i-1));getch();}
```

4) Şartsız Sapma:

Bazen programın icrası, belli bir noktada, programın akışına aykırı olarak **işaretlenmiş** bir noktaya saptırarak istenebilir.Bu durumda **goto** deyiminden faydalanabiliriz. Profesyonel programcılar bu deyimini pek kullanmak istemezler. Biz de dikkatli davranırsak bu komutu kullanmayabiliriz.Yine de bu deyimini öğrenmekte fayda var...

Yazım şekli:

```
...
etiket:
...
goto etiket;
...
```

Örnek:

Klavyeden girilen sayı kadar, istenen karakteri yan yana yazdıran bir programı ve aynı programın tekrar çalışması için **Devam edelim mi? (E/H)** sorusuna E veya e karakteri girilmesi durumunda programın çalışmasını sağlayan bir programı;
a) goto deyimini kullanarak;
b) do-while döngüsü kullanarak yazalım.

Çözüm:

a) (cprog22.c)

```
#include<stdio.h>
int i,n;char kar;
main()
{basla:i=1;printf("\nHangi karakter? ");scanf("%s",&kar);
printf("\nKaç tane? ");scanf("%d",&n);
for(i=1;i<=n;i++) {printf("%c",kar);}
printf("\nDevam edelim mi? (E/H)");kar=getch();
if(kar=='E' || kar=='e') goto basla;
printf("\n\t\t\tİyi Günler...!");getch();}
```

C Ders Notları

b) (cprog23.c)

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

```
int i,n;char kar;
```

```
main()
```

```
{do
```

```
{i=1;printf("\nHangi karakter? ");scanf("%s",&kar);
```

```
printf("\nKaç tane? ");scanf("%d",&n);
```

```
for(i=1;i<=n;i++) {printf("%c",kar);}
```

```
printf("\nDevam edelim mi? (E/H)");kar=getch();}
```

```
while((kar=='E') || (kar=='e'));
```

```
printf("\n\t\t\tİyi Günler...!");getch();}
```

C de Diziler:

Aynı tipteki veri grupları tek bir dizi ismi ile belirtilebilir.

Örneğin; 24 kişilik bir sınıftaki öğrencilerin numaraları bir program içinde kullanılmak istenseydi; her bir elemana değişken adı vermek gerekirdi.

Yani bu grubu;

int ogrno01,ogrno02,ogrno03,...,ogrno24; biçiminde tanımlamak gerekirdi.

Bu ise hem uzun hem de kullanışı zor olur.Bunun yerine adı **ogrno** olan 24

elemanlı bir dizi tanımlayabiliriz.Bunu da;

int ogrno[24]; tanımlamasıyla yapabiliriz.

Burada ogrno[0] dizinin 1. terimini, ogrno[1] dizinin 2. terimini,...,ogrno[23] ile dizinin 24. terimini belirtir.Buradaki 0,1,2,...,23 değerlerine dizinin **indisi** denir.

Dizi Çeşitleri:

Yukarıda tanımlanan dizinin herhangi bir elemanına ulaşmak için tek bir indis yeterlidir.Bu biçimdeki dizilere **bir boyutlu** diziler denir.Bunun gibi iki boyutlu, üç boyutlu, ..., n boyutlu (**çok boyutlu**) dizilerden de bahsedilebilir.

Bir Dizinin Tanımı:

tip dizi_adi[eleman sayısı]; (bir boyutlu dizi)

tip dizi_adi[eleman sayısı1] [eleman sayısı2]; (iki boyutlu dizi)

tip dizi_adi[eleman sayısı1] [eleman sayısı2] [eleman sayısı3]; (üç boyutlu dizi)

...

Dizinin Elemanlarına Değer Atama:

a) Dizinin elemanlarına verilecek olan değerler programın çalışması sırasında klavyeden **scanf(i)** fonksiyonu aracılığı ile tanımlanabilir.

b) Aşağıdaki biçimde de diziye ilk değer atama yapılabilir;

```
<tür> <dizi adı> [ [uzunluk] ]={d1,d2,d3,...};
```

Not 1) Diziler tanımlanırken **uzunluk** belirtilmek zorunda değildir.

2) Dizi uzunluğu belirtildiğinde, dizi elemanları **uzunluk** tan fazla sayıda olamaz.

Örnek:

```
a) int puan[5]={55,80,95,45,77};
```

```
b) float agirlik[7]={55.6,77,98.345,66.7,88,95.5,77};
```

```
c) char il[10]='l','z','m','i','r';
```

```
d) char okul[20]="İzmir Fen Lisesi";
```

Örnek:

İlk değeri **İzmir Fen** olan bir karakter dizisini ekrana, karakter karakter yazdıran bir program yazalım

a) (cprog24.c)

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

```
char okul[10]='İ','z','m','i','r',' ','F','e','n';
```

```
int i;
```

```
main()
```

```
{clrscr();for(i=0;i<=9;i++)printf("%c",okul[i]);getch();}
```

b) (cprog25.c)

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

```
char okul[10]="İzmir Fen";
```

```
int i;
```

```
main()
```

```
{clrscr();for(i=0;i<=9;i++)printf("%c",okul[i]);getch();}
```

Örnek: Adı **sayi** olan tamsayılardan oluşan 10 elemanlı bir sayı dizisini a) scanf() fonksiyonu ile b) ilk değer ataması ile oluşturalım.

Çözüm:

a) (cprog26.c)

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

```
int sayi[10],i;
```

```
void main()
```

```
{ for (i=0;i<=9;i++)
```

```
{ printf("\n %d sayı ->",i+1);scanf("%d",&sayi[i]);
```

```
getch();}
```

b) (cprog27.c)

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

```
int sayi[10]={7,-13,4,55,2006,-2005,0,88,99,23};
```

```
main() {clrscr();getch();}
```

Not: Burada main() ana fonksiyonu önünde **void** deyimini görüyoruz.Bu deyim bir değer döndürmeyen fonksiyonların önüne konulur.Burada konulmasa da olur.

Örnek: Yukarıda oluşturulan sayı dizisinin;

a) 5. elemanını yazdıran;

b) İstenen sıradaki elemanı yazdıran;

c) Dizinin toplamını ve aritmetik ortalamasını buldurup sonucu ekrana yazdıran kodları ilave ediniz.

Çözüm:

```
a) printf("%d",sayi[4]);
```

```
b) printf("\n dizinin kaçınıcı elemanını istersin?(1-10)");scanf("%d",&i);  
printf("%d",sayi[i-1]);
```

c) (cprog28.c)

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

```
int sayi[10],i;long top;float ort;
```

```
void main ()
```

```
{ for (i=0;i<=9;i++)
```

```
{ printf("\n %d sayı ->",i+1);scanf("%d",&sayi[i]);
```

```
top=0;
```

```
for(i=0;i<=9;i++)
```

C Ders Notları

```
{top=top+sayi[i];ort=top/10;printf("\n dizinin toplamı = %ld",top);
printf("\n dizinin ortalaması = %.3f",ort);
getch();}
```

Örnek: 5 kişilik bir öğrenci grubunun numaralarını **ogrno** dizisine atayarak, sırayla alt alta listeleyen bir program yazalım.

Çözüm: (cprog29.c)

```
#include<stdio.h>
int ogrno[4],i;
void main ()
{ for (i=0;i<=4;i++)
{ printf("\nogrno[%d] no ->",i);scanf("%d",&ogrno[i]);
printf("\n Girilen elemanlar...");
for (i=0;i<=4;i++)
{printf("\n %d. öğrenci, ogrno[%d] -> %d" ,i+1,i,ogrno[i]); getch();}
```

Örnek:

Satır ve sütunlardan oluşan, dikdörtgen biçimindeki reel sayı tablolarına **matris** denir.

Örneğin;

$$\begin{bmatrix} 1 & -3 & 4 \\ -2 & 0 & 1 \end{bmatrix}$$

matrisi 2 satır 3 sütundan oluşan bir matris yani 2x3 lük bir matristir.

Matrisler iki boyutlu dizilere bir örnektir.Örneğin yukarıdaki matris a matrisi olsun; burada a[1][1] in değeri 1; a[1][2] in değeri -3; a[1][3] in değeri 4; a[2][1] in değeri -2; a[2][2] in değeri 0; a[2][3] in değeri 0 dir.

2x3 lük bir matrisin elemanlarını girip, matrisi ekranda görüntüleyen bir program yazalım.

(cprog30.c)

```
#include<stdio.h>
float a[2][3];int i,j;
void main ()
/*Matrisin elemanları giriliyor*/
{ for (i=0;i<=1;i++) {for(j=0;j<=2;j++)
{printf("%d. %d. ",i+1,j+1);scanf("%f",&a[i][j]);}}
/*Matris görüntüleniyor*/
{ for (i=0;i<=1;i++)
{for(j=0;j<=2;j++){printf("\t%.2f",a[i][j]);printf("\n");}
getch();}}
```

Not:

1) Matris görüntülenirken iç-içe iki tane for döngüsü kullanılmıştır.i değişkeni satır sayısını, j değişkeni ise sütun sayısını tutmaktadır.

```
{ for (i=0;i<=1;i++)
{for(j=0;j<=2;j++){printf("\t%.2f",a[i][j]);printf("\n");}}
```

2) Matrisin elemanları ekrana yazdırılırken yazdırma formatında **\t** ifadesi (bir tab atlatma) kullanılmıştır.

3) i döngüsü tamamlandığında, bir sonraki dizi elemanını satır başından itibaren yazdırmak için **\n** (new line) kullanılmıştır.

4) Matrisin elemanlarını satır ve sütunlar biçiminde yazdırmak için **gotoxy(m,n)**; fonksiyonundan faydalanılabilir.Bu fonksiyondan sonra kullanılan printf

fonksiyonu ile yazdırılan ifadenin başlangıç noktası m. sütun, n. satır olacaktır.Buna göre program aşağıdaki gibi olmalıdır:

(cprog31.c)

```
#include<stdio.h>
float a[2][3];int i,j;
void main ()
/*Matrisin elemanları giriliyor*/
{ for (i=0;i<=1;i++) {for(j=0;j<=2;j++)
{printf("%d. %d. ",i+1,j+1);scanf("%f",&a[i][j]);}}
/*Matris görüntüleniyor*/
clrscr();{ for (i=0;i<=1;i++) {for(j=0;j<=2;j++)
{gotoxy(7*(j+1),i+1);printf("%.2f",a[i][j]);}getch();}}
```

Örnek: Satır ve sütun sayısı eşit olan ve esas köşegen üzerindeki elemanları 1 geri kalan elemanları 0 olan kare matrise **birim matris** denir.Buna göre klavyeden girilen n. (n≤20) mertebeden bir kare matrisi yazdıran bir program yazalım.

Çözüm: (cprog32.c)

```
#include<stdio.h>
int a[21][21],i,j,n;
void main ()
/*birim matris oluşturuluyor*/
{printf("Birim matrisin mertebesini gir (n<=20) -> ");
scanf("%d",&n);clrscr();
{ for (i=1;i<=n;i++) {for(j=1;j<=n;j++)
if (i==j) a[i][j]=1; else a[i][j]=0;}}
```

```
/*Matris yazdırılıyor*/
for(i=1;i<=n;i++){for(j=1;j<=n;j++)
{gotoxy(3*j,i);printf("%d",a[i][j]);}getch();}
```

Not: Programdaki /* ile */ arasındaki bölüm derleyici tarafından göz önüne alınmaz.Bu bölüm açıklama amaçlı kullanılır.

Örnek:

Klavyeden girilen n elemanlı (n≤10) bir sayı dizisinin; a) İstenen bir sıradaki elemanını ekrana yazdıran; b) Dizinin en büyük ve en küçük elemanını bulup yazdıran; c) İstenen iki sayı arasındaki elemanlar toplamını bulduran; d) Diziyi küçükten büyüğe doğru sıralayıp ekrana yazdıran bir program yazalım.

Çözüm:

a) (cprog33.c)

```
#include<stdio.h>
int i,j,n;float a[10],x,eb,ek;
void main()
{clrscr();
printf("Dizi kaç elemanlı olsun? (n<=10) ");
scanf("%d",&n);
for(i=1;i<=n;i++){printf("%d . eleman\n",i);
scanf("%f",&a[i]);}
printf("\ndizininin kaçınıcı elemanını istersin? ");
scanf("%d",&n);
printf("%d . sıradaki eleman %.3f dir",n,a[n]);getch();}
```


C Ders Notları

b) (cprog34.c)

```
#include<stdio.h>
int i,j,n;float a[10],x,eb,ek;
void main()
{ek=9999;eb=-9999;clrscr();printf("Dizi kaç elemanlı
olsun? (n<=10) ");scanf("%d",&n);
for(i=1;i<=n;i++){printf("\n%d . eleman ",i);
scanf("%f",&a[i]);}

for(i=1;i<=n;i++){if (ek>=a[i]) ek=a[i];if (eb<=a[i])
eb=a[i];}
printf("dizinin en küçük elemanı %.3f en büyük elemanı
%.3f dir",ek,eb);getch();}
```

c) (cprog35.c)

```
#include<stdio.h>
int i,j,n;float a[10],x,eb,ek;
void main()
{x=0;clrscr();printf("Dizi kaç elemanlı olsun? (n<=10) ");
scanf("%d",&n);
for(i=1;i<=n;i++)
{printf("%d . eleman\n",i);
scanf("%f",&a[i]);}

printf("\naralığın alt sınırını gir =");scanf("%f",&ek);
printf("\naralığın üst sınırını gir =");scanf("%f",&eb);

for(i=1;i<=n;i++){if (ek<=a[i] & a[i]<=eb) then x=x+a[i];}
printf("toplam = %.2f dir",x);getch();}
```

d) (cprog36.c)

```
#include<stdio.h>
int i,j,n;float a[10],x,eb,ek;
void main()
{clrscr();printf("Dizi kaç elemanlı olsun? (n<=10)
");scanf("%d",&n);
for(i=1;i<=n;i++){printf("\n%d . eleman ",i);
scanf("%f",&a[i]);}
for(i=1;i<=n;i++) for(j=i;j<=n;j++){if (a[i]>a[j])
{x=a[j];a[j]=a[i];a[i]=x;}}
printf("..... sıralanmış elemalar.....");
for(i=1;i<=n;i++){printf("\n%d . eleman %.3f",i,a[i]);}
getch();}
```

Örnek: -1 den dan 1 e kadar 0.1 er artışla (21 elemanlı) a adında bir reel sayı dizisine aktarıp küplerini de kup adlı diziye aktarıp listeleyen bir program yazalım.

Çözüm: (cprog37.c)

```
#include<stdio.h>
int i,n;double x,a[22],kup[22];
main() {clrscr();x=-
1;for(i=1;i<=21;i++){a[i]=x;kup[i]=a[i]*a[i]*a[i];x=x+0.1;}
for(i=1;i<=21;i++)printf("\n%d %.2f %.3f ",i,a[i],kup[i]);
getch();}
```

String Dizileri:

C'de, diğer programlama dillerinde olduğu anlamda **string** (karakter dizisi) yoktur.Bunu programcı karakterlerden oluşan bir dizi olarak tanımlar.Bunu **char** deyimi yardımıyla yaparız.

Örneğin 10 karakterlik ad adlı bir stringi; **char ad[10];** deyimiyle yapabiliriz.

Şayet karakter dizisinden birden fazla tanımlanacaksa bunu aşağıdaki gibi; 2 boyutlu karakter dizisi ile yapabiliriz.

char dizininadı[eleman_sayısı][string_uzunluğu];

Örnek:

Haftanın günlerini, 10 karakterlik **gun** adlı bir string dizisine aktarıp sırayla listeleyen bir program yazalım.

Çözüm: (cprog38.c)

```
#include<stdio.h>
int i;
char gun[7][10];
main()
{clrscr();for(i=1;i<=7;i++){printf("\n%d.gün ",i);
scanf("%s",gun[i]);}
for(i=1;i<=7;i){printf("\n %d. gün %s",i,gun[i]);}
getch();}
```

C de Fonksiyonlar:

C de alt programların tümü bir fonksiyon olarak tanımlanır.Genelde bir fonksiyon, tanımlanan işlemleri yaptıktan sonra bir değer geri döndürür.Ancak fonksiyonun bir değer döndürmesi gerekmediğinde fonksiyon tipi **void** olarak belirtilmelidir.

Fonksiyonun Yapısı:

```
fonksiyon_tipi fonksiyon_adi(parametreler)
{tanımlama bloğu;
ana blok;
...
return(geri döndürülecek değer);}
```

Örnek: Adı **topla** olan ve girilen iki tam sayının toplamını döndüren bir fonksiyon tanımlayalım.Daha sonra 7 ile -13 ün toplamını bu fonksiyonu kullanarak ekrana yazdıran bir program yazalım.

Çözüm: (cprog39.c)

```
#include<stdio.h>
int topla(int a,int b)
{return(a+b);}
main()
{clrscr();printf("toplam %d",topla(7,-13));
getch();}
```

Örnek: Adları **topla, cikar,carp** ve **bol** olan dört fonksiyon tanımlayalım öyle ki parametre olarak girilen iki sayıyı toplasın, çıkarsın, çarpsın veya bölsün ve sonucu ekranda aşağıdaki örneğe benzer yazsın: **13.000 / 7.000 = 1.857** gibi.

Çözüm: (cprog40.c)

```
#include<stdio.h>
#include<conio.h>
float topla(float a,float b)
{return(a+b);}
```

C Ders Notları

```
float cıkar(float a,float b)
{return(a-b);}
float carp(float a,float b)
{return(a*b);}
float bol(float a,float b)
{return(a/b);}
float x,y;char kar;
main()
{clrscr();printf("\nişlem yapılacak ilk sayı ->");
scanf("%f",&x);
printf("\nişlem yapılacak ikinci sayı ->");scanf("%f",&y);
printf("\nişlemi seçiniz +, -, x, / ->");kar=getch();
switch(kar)
{case '+':printf("%.3f + %.3f = %.3f",x,y,topla(x,y));break;
case '-':printf("%.3f - %.3f = %.3f",x,y,cıkar(x,y));break;
case 'x':printf("%.3f x %.3f = %.3f",x,y,carp(x,y));break;
case '/':printf("%.3f / %.3f = %.3f",x,y,bol(x,y));break;}
getch();}
```

Örnek: Adı **cizgi** olan ve birinci parametre sayısı kadar , ikinci parametre karakterini yan yana ekranda görüntüleyen bir fonksiyon yazıp kullanan bir program yazalım.

Çözüm: (cprog41.c)

```
#include<stdio.h>
void ciz(int n,char kar)
{int i;for(i=1;i<=n;i++) printf("%c",kar);}
main()
{char kar;int n;clrscr();
printf("\nhangi karakteri kullanacaksın ->");
kar=getch();printf("%c",kar);
printf("\nkaç defa yazdıracaksın ->"); scanf("%d",&n);
ciz(n,kar);
getch();}
```

Örnek:

- n bir doğal sayı olmak üzere 1 den n e kadar olan sayılar toplamını **topla(n)** fonksiyonu ile bulduran,
- Sayıların kareleri toplamını **topkare(n)** fonksiyonu ile bulduran,
- Sayıların kareköklerinin toplamını **topkarekok(n)** fonksiyonu ile bulduran,
- m ve n doğal sayılar olmak üzere m den n ye kadar olan sayıların toplamını **toplamm(m,n)** fonksiyonu ile bulduran,
- n doğal sayı olmak üzere n! değerini **faktoryel(n)** fonksiyonu ile bulduran,

f) e sayısının seriye açılımı; $e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$ olduğuna göre; x reel sayı olmak üzere serinin ilk n+1 terimini toplayarak e sayısının yaklaşık değerini bulan bir **e(n)** fonksiyonunu **faktoryel(n)** fonksiyonunu kullanarak yazınız.

g) $e^x = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ serisinden faydalanarak x reel sayı olmak üzere serinin ilk n+1 terimini kullanan **e(x,n)** fonksiyonu ile bulduran bir program yazalım.

Çözüm:

a) (cprog42.c)

```
#include<stdio.h>
long topla(long n)
{long i,t=0;for(i=1;i<=n;i++) t=t+i; return(t);}
main()
{long x;clrscr();
printf("\n1 den kaçta kadar toplam istersin? ");
scanf("%ld",&x);
printf("1+2+3+...+%ld = %ld dir.",x,topla(x));
getch();}
```

Not: Bu programı çalıştırdığımızda x yerine 70 000 gibi bir değer yazıldığında negatif bir değer çıkar ki bu yanlış bir sonuçtur.Bunun sebebi bu toplamın long sınırını taşıdığındandır.Bunu düzeltmek için değişkenleri long (yani uzun tam sayı) yerine double alabiliriz.Bunu aşağıdaki programda görebiliriz.

(cprog43.c)

```
#include<stdio.h>
double topla(long n)
{long i;double t=0;for(i=1;i<=n;i++) t=t+i; return(t);}
main()
{long x;clrscr();
printf("\n1 den kaçta kadar toplam istersin? ");
scanf("%ld",&x);
printf("1+2+3+...+%ld = %.0lf",x,topla(x));
getch();}
```

b) (cprog44.c)

```
#include<stdio.h>
double topkare(long n)
{long i;double t=0;for(i=1;i<=n;i++) t=t+i*i; return(t);}
main()
{long x;clrscr();
printf("\n1 den kaçta kadar toplam istersin? ");
scanf("%ld",&x);
printf("1^2+2^2+3^2+...+%ld^2 = %.0lf",x,topla(x));
getch();}
```

c) (cprog45.c)

```
#include<stdio.h>
#include<math.h>
double topkarekok(float n)
{float i;double t=0;for(i=1;i<=n;i++) t=t+sqrt(i); return(t);}
main()
{float x;clrscr();
printf("\n1 den kaçta kadar toplam istersin? ");
scanf("%f",&x);
printf("1 den %.0f ye kadar karekokler toplamı = %.0lf",x,topkarekok(x));
getch();}
```

d) (cprog46.c)

```
#include<stdio.h>
double toplamn(long m,long n)
{long i;double t=0;for(i=m;i<=n;i++) t=t+i; return(t);}
main()
{long x,y;clrscr();
printf("\nilk sayı kaç olsun? ");scanf("%ld",&x);
printf("\nikinci sayı kaç olsun? ");scanf("%ld",&y);
printf("%ld+%ld+...+%ld = %.0lf",x,x+1,y,toplamn(x,y));}
```

C Ders Notları

```
getch();}
```

e) (cprog47.c)

```
#include<stdio.h>
double faktoryel(int n)
{int i;double c=1;for(i=1;i<=n;i++) c=c*i; return(c);}
main()
{int x;clrscr();
printf("\nkaç faktöryel istersin? ");scanf("%d",&x);
printf("%d! = %.0lf",x,faktoryel(x));
getch();}
```

f) (cprog48.c)

```
#include<stdio.h>
double faktoryel(int n)
{int i;double c=1;for(i=1;i<=n;i++) c=c*i; return(c);}
```

```
double e(int n)
{int i;double
t=0;for(i=1;i<=n;i++)t=t+1/faktoryel(i);return(1+t);}
```

```
main()
{int m;clrscr();
printf("\ne sayısı için kaç terim toplansın? ");
scanf("%d",&m);
printf(" %.20lf",e(m));
getch();}
```

g) (cprog49.c)

```
#include<stdio.h>
#include<math.h>
double faktoryel(int n)
{int i;double c=1;for(i=1;i<=n;i++) c=c*i; return(c);}
```

```
double e(float x,int n)
{int i;double t=0;
for(i=1;i<=n;i++)t=t+pow(x,i)/faktoryel(i);return(1+t);}
```

```
main()
{int m;float y;clrscr();
printf("\n e nin üssü kaç olsun? ");scanf("%f",&y);
printf("\ne üssü %.0f sayısı için kaç terim
toplansın(n<30)? ",y); scanf("%d",&m);
printf(" %.20lf",e(y,m));
getch();}
```

Not: Programdaki **pow(x,y)** fonksiyonu x^y olup **math.h** header dosyasında tanımlıdır.

Örnek:

a) Girilen bir doğal sayının asal olup olmadığını bulduran bir program yazalım.

Not: Sadece 1 e ve kendine bölünebilen sayılara **asal sayı** denir. Bir n sayısının asal sayı olup olmadığını denetlemek için, 2 den n-1 e kadar sayıların n nin böleni olup olmadığını araştırmak gerekir. Hatta 2 den n/2 ye kadar olan sayıların n nin böleni olup olmadığını araştırmak yeterlidir. Hatta 2 den \sqrt{n} ye kadar olan sayıların n nin böleni olup olmadığını araştırmak yeterlidir. Bunun için C de % (mod işlemi) kullanılabilir.

Örneğin 15%4 işleminin sonucu, 15 in 4 e bölümündeki kalan demektir ki bu da 3 tür.

b) 2 den itibaren girilen bir sayıya kadar olan asal sayıları bulup sırayla ekrana yazdıran bir programı **asalmi(n)** fonksiyonu kullanarak yazalım.

Çözüm:

a) (cprog50.c)

```
#include<stdio.h>
main()
{int i,n,s=0;clrscr();
printf("\nhangi sayı ? ");scanf("%d",&n);
for(i=1;i<=n;i++) {if(i%n!=0) s++;}
if(s!=0) printf("%d asal!",n); else printf("%d asal
değil!",n);
getch();}
```

b) (cprog51.c)

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

void asalmi(long n)
{long i,s=0;for(i=2;i<=sqrt(n);i++) {if(n%i==0) s=s+1;}
if(s==0) printf("%d ",n);}
```

```
main() {long j,kac;clrscr();
printf("\nkaçı kadar olan asal sayıları istersin? ");
scanf("%ld",&kac);
for(j=2;j<=kac;j++)asalmi(j);getch();}
```

C de Bazı Standart Fonksiyonlar:

conio.h Fonksiyonları:

clrscr(); metin ekranını temizler ve kursorü ekranının sol üst köşesine konumlandırır.

Kullanımı: clrscr();

textbackground(); metin zemin rengini belirler.

Kullanımı:

textbackground(reng kodu);

Bu renkler 8 tane olup, adları ve kodları şunlardır:

Renk Adı	Renk kodu	Türkçe karş.
Black	0	Siyah
Blue	1	Mavi
Green	2	Yeşil
Cyan	3	Turkuaz
Red	4	Kırmızı
Magenta	5	Mor
Brown	6	Kahverengi
LightGray	7	Açık gri

textcolor(); text ekranında, yazı (karakter) rengini belirler.

Kullanımı:

textcolor(reng kodu);

C Ders Notları

Programlarımızda kullanabileceğimiz yazı renkleri ve kodları aşağıdaki listede verilmiştir.

Renk Adı	Renk kodu	Türkçe karş.
Black	0	Siyah
Blue	1	Mavi
Green	2	Yeşil
Cyan	3	Turkuaz
Red	4	Kırmızı
Magenta	5	Mor
Brown	6	Kahverengi
LightGray	7	Açık gri
DarkGray	8	Koyu gri
LightBlue	9	Açık mavi
LightGreen	10	Açık yeşil
LightCyan	11	Açık turkuaz
LightRed	12	Açık kırmızı
LightMagenta	13	Açık mor
Yellow	14	Sarı
White	15	Beyaz

Örnek:

Kırmızı zeminde beyaz renkle **Türkiye** yazısını yazdıran bir program yazalım.

Çözüm: (cprog52.c)

```
#include<stdio.h>
main()
{textbackground(4);clrscr();textcolor(15);
printf("Türkiye");getch();}
```

Örnek:

8 farklı renkli zeminde, 16 farklı yazı renginde, her (Enter) tuşuna basıldığında, ekrana "İFL" yazdıran bir program yazalım.

Çözüm: (cprog53.c)

```
#include<stdio.h>
int i,j;
main()
{clrscr();for(i=0;i<=7;i=i+1) for(j=0;j<=15;j++)
{{textbackground(i);textcolor(j);printf("FL");}
getch();clrscr();}}
```

textattr(); text ekranında yazının yanıp-sönmesini, renk ve zemin rengi niteliklerini ayarlamaya yarar. Bu nitelikler 8 bit ile ayarlanır.

İlk biti yazının yanıp sönmesini ayarlar, değeri 0-1 aralığındadır. 0 ise normal, 1 ise yanıp-söner,

Sonraki üç biti zemin rengini ayarlar, değeri 0-7 aralığındadır,

Son dört biti yazı rengini ayarlar değeri 0-15 aralığındadır.

Örnek: Mavi zeminde kırmızı renkte yanıp sönen İzmir yazısını yazdırmak için ilgili nitelikleri hesaplayıp ilgili programı yazalım.

Çözüm: (cprog54.c)

Yanıp sönmenin (1) binary kodu 1;

mavi rengin (1) binary kod karşılığı 001; kırmızı rengin (4) binary kod karşılığı 0100; değerlerini yan yana yazalım; 10010100, bu değer onlu sayma düzeninde değerini hesaplayalım; $1.128+1.16+1.4=148$ yapar. O halde programı aşağıdaki gibi yazabiliriz:

```
#include<stdio.h>
int i,j;
main()
{textattr(148);clrscr();printf("İzmir");getch();}
```

gotoxy(m,n); kursorü text ekranının n. satır m. sütuna konumlanmasını sağlar.

Örnek: İzmir Fen Lisesi yazısını, mavi zeminde, beyaz renkli olarak önce 1. sütun 1. satıra, 2. sütun 2. satıra, ..., 20. sütun, 20. satıra yazdıran bir program yazalım.

Çözüm: (cprog55.c)

```
#include<stdio.h>
int i;
main()
{textbackground(1);textcolor(15);clrscr();
for(i=1;i<=20;i++)
{gotoxy(i,i);printf("İzmir Fen Lisesi");}getch();}
```

window(su1,sa1,su2,sa2); text ekranında sol üst köşesi sa1 satır, su1 sütun ile sağ üst köşesi sa2 satır, su2 sütun olan bir pencere açmaya yarar.

Örnek: Sol üst köşesi (5,10) sağ alt köşesi (25,15) koordinatlı bir pencerenin zemin rengini kırmızı yapalım ve bu pencerenin tam ortasına gri renkli İFL yazdıran bir program yazalım.

Çözüm: (cprog56.c)

```
#include<stdio.h>
main()
{window(5,10,25,15);textbackground(4);textcolor(7);
clrscr();gotoxy(10,3);printf("İFL");getch();}
```

kbhit(); klavyedeki herhangi bir tuşa basılıp basılmadığını kontrol etmeye yarar. Bir tuşa basılmadığı sürece 0 değerini, basıldığında 1 değerini döndürür.

Örnek: Ekrana **www.ifl.k12.tr** ifadesini;

- Sürekli yazdıran;
- Her (enter) tuşuna bastıkça yazdıran;
- Herhangi bir tuşa basana kadar yazdıran birer program yazalım.

Çözüm:

a) (cprog57.c)

```
#include<stdio.h>
main() {do{printf("www.ifl.k12.tr ");}while(1==1);}
```

b) (cprog58.c)

```
#include<stdio.h>
main() {do{printf("www.ifl.k12.tr ");}while(getch());}
```

C Ders Notları

c) (cprog59.c)

```
#include<stdio.h>
main() {clrscr();do{printf("www.ifl.k12.tr" );}
while(!kbhit());}
```

lowvideo(); text ekranında yazının görüntüsünü matlaştırmaya yarar.

highvideo(); text ekranında yazının görüntüsünü parlaklaştırmaya yarar.

normvideo(); text ekranında yazının görüntüsünü normal hale getirmeye yarar.

wherex(); text ekranındaki kursorün (imlecin) o andaki bulunduğu sütun sayısını (x pozisyonunu) elde etmeye yarar.

wherey(); text ekranındaki kursorün (imlecin) o andaki bulunduğu satır sayısını (y pozisyonunu) elde etmeye yarar.

Örnek: Kursorün en son bulunduğu satır ve sütunu bildiren bir program yazalım.

Çözüm: (cprog60.c)

```
#include<stdio.h>
int sat,sut;
main() {sat=wherex();sut=wherey();
printf("kursor en son (%d , %d) koordinatlarında
idi",sut,sat);getch();}
```

string.h Fonksiyonları:

strcpy(str1,str2); C de herhangi bir stringi direkt olarak bir değişkene atama yapılamaz.Bunun yerine strcpy(); fonksiyonu ile kaynak karakter dizisini (str2 stringini), hedef karakter dizisine (str1 stringine) kopyalayabiliriz.

strncpy(str1,str2,n); kaynak karakter dizisinin (str2 stringinin) ilk n karakterini, hedef karakter dizisine (str1 stringine) kopyalamaya yarar.

strlen(str); str stringinin karakter sayısını bulmaya yarar.

strcat(str1,str2); str1 hedef stringinin sonuna, str2 kaynak stringini eklemeye yarar.

strncat(str1,str2,n); str1 hedef stringinin sonuna, str2 kaynak stringinin ilk n karakterini eklemeye yarar.

strcmp(str1,str2); str1 stringi ile str2 stringini karşılaştırmaya yarar.Şayet str1 küçükse geriye döndürülen sayı negatif, str2 daha büyükse pozitif eşitse sıfırdır.

strncmp(str1,str2,n); iki stringin ilk n karakterini karşılaştırmaya yarar.

strlwr(str); str stringinin karakterlerini küçük harfe çevirmeye yarar.

strupr(str); str stringinin karakterlerini büyük harfe çevirmeye yarar.

strstr(str1,str2); str1 de str2 stringini arar ve ilk bulunduğu yerin adresini döndürür.

strchr(str,ch); str de ch karakterini arar ve ilk bulunduğu yerin adresini döndürür.

strset(str,ch); str stringindeki karakterlerin tümünü ch karakteri ile değiştirir.

strnset(str,ch,n); str stringindeki ilk n karakteri ch karakteri ile değiştirir.

strrev(str); str stringin tersine çevirmeye yarar.

math.h Fonksiyonları:

abs(x); x sayısının mutlak değerini bulmaya yarar.

cos(x); radyan olarak belirtilen sayısının kosinüsünü bulmaya yarar.

Not: Dereceyi radyana çevirmek için pi sayısı (3.1415926) ile çarpıp 180 e bölmek gerekir.

sin(x); radyan olarak belirtilen sayısının sinüsünü bulmaya yarar.

tan(x); radyan olarak belirtilen sayısının tanjantını bulmaya yarar.

Örnek: 135° nin tanjantıyla 60° nin kosinüsünü ekrana yazdıran bir program yazalım.

Çözüm: (cprog61.c)

```
#include<stdio.h>
#include<math.h>
float pi=3.1415926;
main()
{clrscr();
printf("%f %f",tan(135*pi/180),cos(60*pi/180));
getch();}
```

asin(x); -1 ile 1 arasındaki bir sayının arksinüsünü bulmaya yarar.

acos(x); -1 ile 1 arasındaki bir sayının arkkosinüsünü bulmaya yarar.

atan(x); x sayısının artanjantını bulmaya yarar.

exp(x); e^x değerini bulmaya yarar.

pow(x,y); x^y değerini bulmaya yarar.

log(x); x in doğal logaritmasını (ln x) bulmaya yarar.

C Ders Notları

log10(x); x in 10 tabanındaki logaritmasını (log x) bulmaya yarar.

sqrt(x); x pozitif sayısının karekökünü (\sqrt{x}) bulmaya yarar.

ceil(x); x sayısını yukarıya yuvarlamaya yarar.

floor(x); x sayısını aşağıya yuvarlamaya (tam değerini) yarar.

dos.h Fonksiyonları:

delay(x); programın çalışmasını x milisaniye durdurmaya yarar.

sleep(x); programın çalışmasını x saniye durdurmaya yarar.

sound(x); bilgisayarın hoparloründen x hertzlik frekansta ses çıkarmaya yarar.

nosound(); bilgisayarın hoparloründen çıkan sesi durdurmaya yarar.

Örnek: Bilgisayarın hoparloründen 10 saniye 1000 hertzlik ses çıkartıp sesini kesen bir program yazalım.

Çözüm: (cprog62.c)

```
#include<stdio.h>
#include<dos.h>
main() {sound(1000);sleep(10);nosound();}
stdlib.h Fonksiyonları:
```

randomize(); rastgele sayı üreticisini başlatmaya yarar.

random(x); 0 ile x tamsayısı arasında bir sayı üretmeye yarar.

Örnek:

- 0 ile 100 arasında 10 tane rastgele sayı üreten bir program yazalım.
- 30 ile 50 arasında rastgele bir sayı üreten bir program yazalım.
- Girilen m tam sayısı ile n tam sayısı arasında rastgele bir sayı üreten bir program yazalım.

Çözüm:

a) **(cprog63.c)**

```
#include<stdio.h>
#include<stdlib.h>
long i;
main(){clrscr();randomize();for(i=1;i<=10;i++)
{printf("\n%d",random(100));getch();}
```

b) **(cprog64.c)**

```
#include<stdio.h>
#include<stdlib.h>
lmain(){clrscr();randomize();
printf("\n%d",20+random(30));getch();}
```

c) **(cprog65.c)**

```
#include<stdio.h>
#include<stdlib.h>
int m,n;
main(){clrscr();printf("\nİlk sayı = ");scanf("%d",&m);
printf("\nİkinci say = ");scanf("%d",&n);
randomize();
printf("\n%d",m+random(n-m));getch();}
```

Örnek: Bilgisayarın rastgele tuttuğu, 1 ile 100 arasındaki sayıyı bulmaya çalışacağız. Bilgisayar tahminimizi soracak, girdiğimiz tahmin bilgisayarın tuttuğu sayıdan **küçükse "Büyültünüz", büyükse "Küçültünüz"** yazacak. Sonuçta kaç defada bildiğimizi ekrana yazdıran bir program yazalım .

Çözüm: (cprog66.c)

```
#include<stdio.h>
#include<stdlib.h>
int x,tahmin,say;
main(){clrscr();
printf("\nSAYI BULMACA OYUNU");
printf("\n=====");
printf("\n Bu oyun 1 ile 100 ");
printf("\n arasında tutulan ");
printf("\n sayıyı bulma esa- ");
printf("\n sına dayanır. ");
printf("\nHaydi kolay gelsin.");
randomize();x=1+random(100);
do {printf("\nTahminini gir ...");scanf("%d",&tahmin);
if(tahmin<x){printf("\nBiraz büyült...");} else
if(tahmin>x){printf("\nBiraz küçült...");}say++;}
while(x!=tahmin);
printf("\nTutulan sayı %d idi bunu %d defada
bildin...",x,say);
getch();}
```

C'de Grafik:

C'de grafik çizdirmeye yarayan bir çok fonksiyon vardır. Bu fonksiyonlar **graphics.h** header dosyasında bulunur. Şimdi bunlardan bazılarını görelim:

graphics.h Fonksiyonları:

initgraph(); Donanımı grafik arabirimine hazırlamak için *grafik sürücüsü* ve *modunu* grafik kartından tanıy ve arabirimi yükler.

Kullanımı:

initgraph(&gsurucusu,&gmodu,surucuyolu);

graphresult(); En son kullanılan hatalı (çalışmamış) grafik kodunu verir. **GrOk**, graphics.h ta tanımlanmış bir sabit olup 0 (sıfır) değerine karşılık gelir. Eğer grafik komutu doğru çalışmış ise bu kod üretilir.

Kullanımı: **tamsayıdeğişken=graphresult();**

Örneğin; **hata kodu=graphresult();**

grapherrormsg(); Üretilen grafik hata koduna karşılık gelen mesajı, İngilizce olarak bir stringte verir.

Kullanımı: **grapherrormsg(hata kodu);**

Örnek:

```
...
printf("Grafik hatası:%s\n", grapherrormsg(hata kodu));
...
```

closegraph(); Grafik ekranı kapatarak, text (yazı) ekranına geçmeye yarar.

line(); Grafik ekranda, belirtilen noktalar arasında bir doğru parçası (çizgi) çizer.

Kullanımı: **line(x1,y1,x2,y2);**

Şimdi buraya kadar olan bilgilerle bir program yazalım.

Örnek: Önce text ekranında olduğunu belirten, bir tuşa basıldığında, grafik ortama geçip, grafik ekranda, (0,0) koordinatlı noktayı, (100,200) koordinatlı noktaya birleştiren, bir tuşa basıldığında grafik ortamını kapatıp tekrar text ortamına dönen bir program yazalım.

Çözüm: (cprog67.c)

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

main()
{int gsurucusu=DETECT,gmodu,hata kodu;clrscr();
printf("Şimdi Text ortamındayız ...");getch();
initgraph(&gsurucusu,&gmodu,"");
line(0,0,100,200);getch();closegraph();
printf("Tekrar text ortamına döndük...");getch();}
```

cleardevice(); Grafik ekranı siler.

setcolor(); Grafik çizgi rengini tanımlamaya yarar.

setbkcolor(); Grafik zemin rengini tanımlamaya yarar

Örnek: (cprog68.c)

```
#include<stdio.h>
#include<graphics.h>
main()
{int gsurucusu=DETECT,gmodu,hata kodu,i;
clrscr();
initgraph(&gsurucusu,&gmodu,"");
setbkcolor(1);for(i=1;i<=15;i++)
{setcolor(i);line(0,0,100+25*i,200+25*i);getch();}
cleardevice();getch();closegraph();getch();}
```

Yukarıdaki programın çalıştırılması durumundaki olayları belirtiniz.

Çözüm:

- 1) Ekran silinir.
- 2) Grafik ekrana geçer.
- 3) Mavirenkli zeminde;
- 4) 1 den 15 e kadar çizgi renklerini kullanarak;

5) Başlangıcı (0,0) (sol üst köşe), bitimi (100,200) noktasına her seferinde 25 ile i sayaç değeri çarpımı kadar eklenerek bulunan noktaya, i sayaç değeri renkkodunda doğru parçaları çizer.

outtext(); Grafik ekranda bir string yazdırmaya yarar.

Kullanımı: **outtext(string);**

Örnek:

```
...
outtext("Grafik Ekranı");
...
```

outtextxy(); Grafik ekranda, belirlenen koordinattan itibaren, bir string yazdırmaya yarar.

Kullanımı: **outtextxy(x,y,string);**

Örnek: Grafik ekranda (10,50) koordinatlı nokta ile (100,200) koordinatlı noktalar arasını birleştiren ve uç noktalara sırasıyla A, B yazan [AB] doğru parçasını çizdiren bir program yazalım.

Çözüm: (cprog69.c)

```
#include<stdio.h>
#include<graphics.h>
main()
{int gsurucusu=DETECT,gmodu;
clrscr();
initgraph(&gsurucusu,&gmodu,"");
outtextxy(10,50,"A");line(10,50,100,200);
outtextxy(100,200,"B");getch();cleardevice();
closegraph();getch();}
```

moveto(x,y); Piksel imlecini grafik ekranının (x,y) koordinatına taşır.

moverel(dx,dy); Piksel imlecininin x koordinatın dx kadar, y koordinatını da dy kadar artırır.

lineto(x,y); Aktif piksel imlecinin bulunduğu yerden, (x,y) koordinatlı noktayı doğru parçası ile birleştirir.

getmaxx(); Grafik ekranın maksimum x (yatay) koordinatını verir.

getmaxy(); Grafik ekranın maksimum y (düşey) koordinatını verir.

getx(); Aktif piksel imlecinin x koordinatını verir.

gety(); Aktif piksel imlecinin y koordinatını verir.

Örnek: (cprog70.c)

```
#include<stdio.h>
#include<graphics.h>
main()
{int gsurucusu=DETECT,gmodu;
clrscr();
initgraph(&gsurucusu,&gmodu,"");}
```

C Ders Notları

```
moveto(50,50);lineto(200,50);lineto(200,300);
lineto(50,50);getch();
closegraph();}
```

Yukarıdaki programın çıktısını tahmin edin.

Aynı çıktıyı veren programı sadece line(); fonksiyonu ile nasıl yazabiliriz?

Örnek: Grafik ekranının, rastgele bir noktasından başlayarak, her defasında bir tuşa bastığımızda, ekranın rastgele bir noktasına doğru bir çizgi çizen, c karakterine bastığımızda programı sonlandıran bir program yazalım.

Çözüm: (cprog71.c)

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
main()
{int gsurucusu=DETECT,gmodu,x,y;
char kar="";
clrscr();
initgraph(&gsurucusu,&gmodu,"");randomize();
x=random(getmaxx());y=random(getmaxy());
moveto(x,y);
while(kar!='c'){kar=getch();
x=random(getmaxx());y=random(getmaxy());
lineto(x,y);}closegraph();}
```

rectangle(x1,y1,x2,y2); (x1,y1) koordinatları ve (x2,y2) koordinatlarını köşegen kabul eden dikdörtgeni çizmeye yarar.

Örnek:

Aşağıdaki programın ekran çıktısını tahmin ediniz.

```
#include<stdio.h>
#include<graphics.h>
main()
{int gsurucusu=DETECT,gmodu,i;
initgraph(&gsurucusu,&gmodu,"");
for(i=1;i<=30;i++){setcolor(i);
rectangle(20*i,50,50+20*i,200);i=i+2;}getch();
closegraph();}
```

putpixel(x,y,renk); Ekranın (x,y) koordinatıyla belirli yere renk kodlu renkte bir nokta (piksel) koymaya yarar.

Örnek: Her hangi bir tuşa her basışta, ekranın rastgele noktalarına rastgele renkte noktalar koyan bir program yazalım.

Çözüm: (cprog73.c)

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
main()
{int gsurucusu=DETECT,gmodu,i,x,y;
```

```
clrscr();
initgraph(&gsurucusu,&gmodu,"");
randomize();
do{i=random(15);x=random(getmaxx());y=random(getmaxy());putpixel(x,y,i);getch();}
while(1==1);closegraph();}
```

getpixel(x,y); Ekranın (x,y) koordinatındaki noktanın rengini verir.

bar(x1,y1,x2,y2); (x1,y1) koordinatları ve (x2,y2) koordinatlarını köşegen kabul eden dikdörtgensel bölge ve içini (kutu) çizmeye yarar.

Bar3d(x1,y1,x2,y2,derinlik,ust); (x1,y1) koordinatları ve (x2,y2) koordinatlarını köşegen kabul eden dikdörtgensel bölgeye derinlik değeri ile belirli, 3 boyulu kutu çizdirmeye yarar. Buradaki ust değeri 0 ise kutunun üst kısmı görünmez, 1 ise görünür.

arc(x,y,basaci,bitaci,yaricap); Merkezi (x,y), basaci başlangıç açısı değerinden, bitaci bitiş açısı değerine kadar, yaricap yarıçap uzunluğunda bir çember yayı çizmeye yarar.

circle(x,y,yaricap); Merkezi (x,y) ve yaricap yarıçap uzunluğunda bir çember çizmeye yarar.

ellipse(x,y,basaci,bitaci,xyaricap,yyaricap); Merkezi (x,y), basaci başlangıç açısı değerinden, bitaci bitiş açısı değerine kadar, xyaricap asal eksen uzunluğu yarıçapı,yyaricap yedek eksen uzunluğu yarıçapı kadar olan bir elips yayı çizmeye yarar.

fillellipse(x,y,xyaricap,yyaricap); Merkezi (x,y), xyaricap asal eksen uzunluğu yarıçapı,yyaricap yedek eksen uzunluğu yarıçapı kadar olan içi dolu bir elips çizmeye yarar.

pieslice(x,y,basaci,bitaci,yaricap); Merkezi (x,y), basaci başlangıç açısı değerinden, bitaci bitiş açısı değerine kadar ve yaricap yarıçap uzunluğunda bir daire dilimi çizmeye yarar.

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
main()
{int gsurucusu=DETECT,gmodu,i,x,y;
```

```
clrscr();
initgraph(&gsurucusu,&gmodu,"");
arc(getmaxx()/2,getmaxy()/2,0,90,getmaxy()/2);
ellipse(100,100,0,260,100,50);
fillellipse(200,300,100,50);pieslice(300,100,30,120,100);
getch();
closegraph();}
```


C Ders Notları

Örnek: (cprog74.c)

Aşağıdaki programın ekran çıktısını tahmin edin.

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
main()
{int gsurucusu=DETECT,gmodu,i,x,y;

clrscr();
initgraph(&gsurucusu,&gmodu,"");
arc(getmaxx()/2,getmaxy()/2,0,90,getmaxy()/2);
ellipse(100,100,0,260,100,50);
fillellipse(200,300,100,50);pieslice(300,100,30,120,100);
getch();
closegraph();}
```

Örnek: (cprog75.c)

Aşağıdaki oyun programını inceleyerek, neler olduğunu açıklayınız.

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
#include<math.h>
main()
{int gsurucusu=DETECT,gmodu,i,j,r,x,y,tahx,tahy;
float u,r1,puan=0;
char cevap;
clrscr();
initgraph(&gsurucusu,&gmodu,"");
do{setcolor(4);
randomize();x=random(getmaxx());
y=random(getmaxy());r=random(getmaxy()/2);
circle(x,y,r);gotoxy(1,1);
printf("Ekran (x):%d (y):%d",getmaxx(),getmaxy());
gotoxy(1,2);printf("Tahmin (x): ");
gotoxy(1,3);printf("Tahmin (y): ");
gotoxy(12,2);scanf("%d",&tahx);
gotoxy(12,3);scanf("%d",&tahy);
for(j=1;j<=5;j++)
{setcolor(0);for(i=3;i<=9;i=i+3){circle(tahx,tahy,i);}
sleep(1);
setcolor(4);for(i=3;i<=9;i=i+3){circle(tahx,tahy,i);}
sleep(1);}

u=sqrt(pow((x-tahx),2)+pow((y-tahy),2));
r1=r;if(u<r1)
{printf("Tam isabet...!");puan=puan+10*r1/(r1-u);}
if(u==r){printf("Müthiş...Tam merkezinden
vurdunnn...!");puan=puan+100;}
if(u>r){printf("Karavana...!");puan=puan-10*r1/(u-r1);}

printf("\nPuan : %0.0f ",puan);
gotoxy(65,24);printf("Devam mı?(D/d)");
cevap=getch();clrscr();cleardevice();}
while(cevap=='D' || cevap=='d');
}
```