BM101 Algoritma ve Programlama I

Döngü Komutları

İçerik



- 1. Bileşik Atama Komutu
- 2. ArErma ve Azaltma Operatörleri
- 3. while Komutu
- 4. do--while Komutu
- 5. for Komutu
- 6. break ve continue Komutları
- 7. İç İçe Döngüler

Çözümlü Sorular

Hedefler



- Bileşik atama komutları kullanarak komutlaryazma
- Artırma ve azaltma operatörleri kullanma
- while komutu ile döngü kurma
- do--while komutu ile döngükurma
- f o r komutu ile döngü kurma
- Farklı döngü komutları arasında dönüşüm yapma
- break ve continue komutlarını kullanarak döngüyu sonlandırma
- İç içe döngüler kurabilme

Döngü Komutları

- Döngü komutları, verilen koşula göre komutları tekrar çalışır.
- Bu döngü işlemini while, db-while veya for komutları ile gerçekleştirebiliriz.
- Döngü komutlarını anlatmadan önce bu komutlarla beraber çok kullanılan bileşik atama, arttırma ve azaltma operatörlerini inceleyelim.

1. Bileşik Atama Komutu

 Atama operatörünün (=) hem solunda hem de sağında aynı değişkeni kullanıyorsak bunu bileşik atama komutu kullanarak kısaltabiliriz.

```
değişken = değişken operatör ifade; // Atama komutu

değişken operatör = ifade; // Bileşik atama komutu

sayi = sayi + 3;

sayi +=3;
```

1. Bileşik Atama Komutu...

Tablo 4.1 Bileşik Atama Komutu ile İlgili Örnekler

Atama Komutu	Bileşik Atama Komutu
i = i / k;	i/=k;
j = j - 4;	j-=4;
k = k * 99;	k*=99;
1 = 1 % 2;	1%=2;
no = no / (k+2);	no/=(k+2);
sayi = $3 + (x/2) * sayi;$	sayi*=3+(x/2);

2. Arttırma ve Azaltma Operatörleri

• Bir değişkenin değerini bir artırmak veya bir azaltmak istersek atama veya bileşik atama komutları yerine artırma (++) ve azaltma (--) operatörlerini kullanabiliriz.

```
i++; //i değeri 1 artar
J--; //j değeri 1 azalır
```

2. Arttırma ve Azaltma Operatörleri...

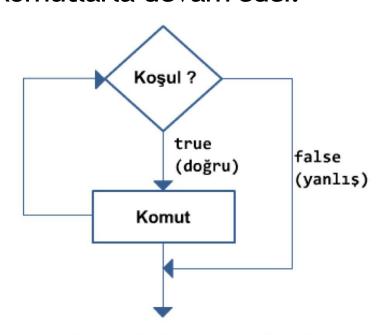
```
i=8,j=6;
i++;
j--;
cout<<"i:"<<i<" j:"<<j;
```

 Arttırma ve azaltma operatörlerini değişkenden önce veya sonra kullanabiliriz. Değişkenden önce kullanılan gösterime ön-ek (prefix) adı verilir.

2. Arttırma ve Azaltma Operatörleri...

 Arttırma ve azaltma operatörlerinin değişkenden sonra kullanımına son – ek (posdix) adı verilir.

 while döngüsünde yer alan komutlar koşul ifadesi true döndürdüğü sürece tekrar edilir. Koşul ifadesi false döndürdüğünde döngü biter ve program döngüden sonraki komutlarla devam eder.



Şekil 4.1 while Komutunun Akış Şeması

while (koşul) komut;

```
Örnek 4.1
#include <iostream>
using namespace std;
int main()
{
    int sayi;
    cout<<"Pozitif bir sayi giriniz:";</pre>
    cin>>sayi;
                                               // Kullanıcının girdiği sayı okunur
    while (sayi<=0)</pre>
                                              // Sayi negatif veya 0 mı?
         cin>>sayi;
                                               // Sayi negatif olduğu için bir daha okunur
    cout<<"Girdiginiz son sayi:"<<sayi<<endl;</pre>
    return 0;
}
Çıktı
Pozitif bir sayi giriniz:0
-3
5
Girdiginiz son sayi:5
```

```
Örnek 4.2
#include <iostream>
using namespace std;
int main()
{
    int sayi;
    cout<<"Pozitif bir sayi giriniz:";</pre>
    cin>>sayi;
                                                    // Kullanıcının girdiği sayı okunur
                                                    // Sayi negatif veya 0 mi?
    while (sayi<=0){
         cout<<"Yanlis sayi girdiniz."<<endl; // Kullanıcıya uyarı mesajı yazılır
         cout<<"Pozitif bir sayi giriniz:";</pre>
         cin>>sayi;
                                            // Sayi negatif olduğu için bir daha okunur
    cout<<"Girdiginiz son sayi:"<<sayi<<endl;</pre>
    return 0;
}
Cıktı
Pozitif bir sayi giriniz: 0
Yanlis sayi girdiniz.
Pozitif bir sayi giriniz: -4
Yanlis sayi girdiniz.
Pozitif bir sayi giriniz: 6
Girdiginiz son sayi:6
```

```
i=10;
while (i<5){
    j+=i;
    i++;
    }
    cout<<i;</pre>
```



Bu örnekte i'ye atanan 10 değeri 5'ten küçük olmadığı için koşul ifadesi false değerini döndürür ve while döngüsü hiç çalışmaz.

Örnek 4.3

```
#include <iostream>
using namespace std;
int main()
{
    int i=10;
    while (i>0){
        cout<<"i= "<<i<<endl;
        i++;
    }
    return 0;
}</pre>
```

Çıktı

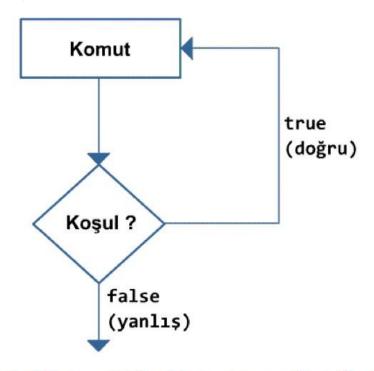
```
i=10
i=11
i=12
...
i=100
i=101
i=102
...
```



Program sonsuz döngüye girer ve sonlanamaz.

4. do-while Komutu

 do--while döngüsünde yer alan komutlar aynı while komutunda olduğu gibi koşul ifadesi true döndürdüğü sürece tekrar edilir.



do komut; while (koşul);

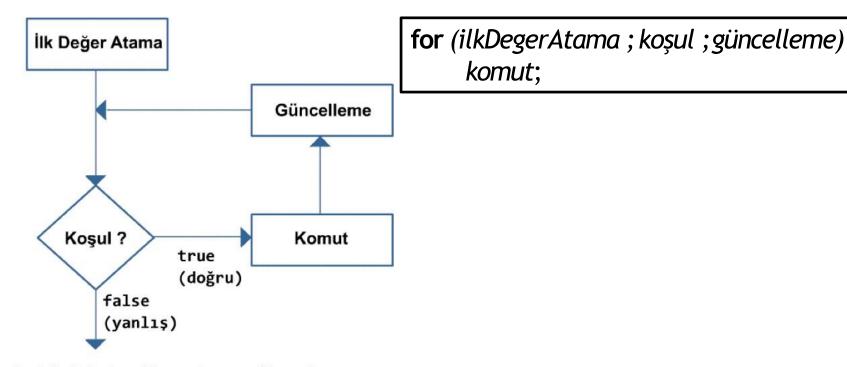
Şekil 4.2 do-while Komutunun Akış Şeması

```
Örnek 4.4
#include <iostream>
using namespace std;
int main()
{
    int sayi;
    cout<<"Pozitif bir sayi giriniz:";</pre>
    do
         cin>>sayi;
                                             // Kullanıcının girdiği sayı okunur
    while (sayi<=0);</pre>
                                             // Sayi negatif veya 0 mi ?
    cout<<"Girdiginiz son sayi:"<<sayi<<endl;</pre>
    return 0;
Çıktı
Pozitif bir sayi giriniz:-3
-6
0
-2
Girdiginiz son sayi:7
```

```
Örnek 4.5
#include <iostream>
using namespace std;
int main()
    int enbuyuk=0,sayi,i=0;
    cout<<"10 pozitif sayi giriniz:";</pre>
    do{
                      // Kullanıcının girdiği sayı okunur
         cin>>sayi;
         if (sayi>enbuyuk) // Şimdiye kadar okunan en büyük sayıyla karşılaştırılır
              enbuyuk = sayi; // Büyükse, yeni en büyük sayı okunan sayı olur
                                         // Okunan sayıların kaç tane olduğu sayılır
         i++;
                                         // 10 sayıdan az mı okundu?
   }while (i<10);
   cout<<"Girdiginiz en buyuk sayi: "<<enbuyuk<<endl;</pre>
   return 0;
Çıktı
10 pozitif sayi giriniz:4 2 6 14 7 9 11 8 5 1
Girdiginiz en buyuk sayi: 14
```

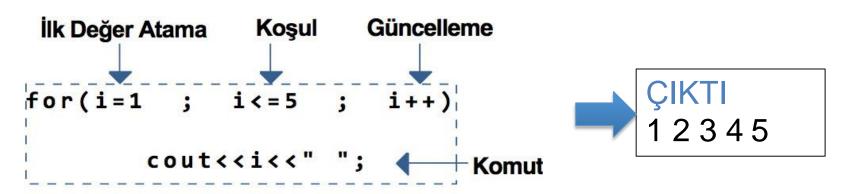
5. for Komutu

 f o r döngüsü özellikle kaç kere döneceği belli olan döngüleri yazarken tercih edilen bir komuttur.



Şekil 4.3 for Komutunun Akış Şeması

5. for Komutu...



Şekil 4.4 for Komutunun İşleyişi

 Değişken tanımları i f ve while komutlarının koşul ifadesi hariç programın herhangi bir yerinde tanımlanabilir.

for (int
$$i=1$$
; $i<10$; $i++$) ...
if (int $z > 4$)... // X Hata

5. for Komutu...

Örnek 4.6 #include <iostream> using namespace std; int main() int sinavNot, enYuksek=0,; for (int sayi=1;sayi<=3;sayi++){</pre> // 3 kere dönen for döngüsü cout<<"Ogrenci notunu giriniz: ";</pre> cin>>sinavNot; // Öğrenci notu okunur if (sinavNot>enYuksek) // Not en yüksek notla karşılaştırılır enYuksek=sinavNot; // Daha büyükse en yüksek nota atanır cout<<"En yuksek not: "<<enYuksek<<endl;</pre> return 0;

Çıktı

```
Ogrenci notunu giriniz: 76
Ogrenci notunu giriniz: 65
Ogrenci notunu giriniz: 88
En yuksek not: 88
```

5. for Komutu...

 f o r döngüsünde aşağıdaki örnekte görüldüğü gibi ilk atama ve güncelleme komutlarında birden fazla komutayer verebiliriz.

```
for (int x=10, y=-5; x>2 && y<0; x-=2, y++) cout<<x<<" "<<y<endl;
```

```
for (int x=10; x<1; x++)
    cout<<x;</pre>
```

Döngü hiç dönmez.

```
a=5;
for(;a<10;){
    cout<<a;
    a++;
}
```

İlk atama komutunu veya güncelleme komutunu boş bırakabiliriz.

6. break ve continue Komutları

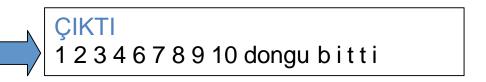
 Bir döngü içinde kullanılan break komutu o döngüden çıkmamızı ve döngüden sonragelen komutlarla programa devam etmemizisağlar.

```
for (int i=1;i<=10;i++){
   if (i==5) break;
   cout<<i;
}
cout<< "dongu bitti";</pre>
CIKTI
1 2 3 4 dongu bitti
```

6. break ve continue Komutlan...

 continue komutu ise döngüde geri kalan komutları yapmadan bir sonraki dönüşe atlar.

```
for(int i=1;i<=10;i++){
   if (i==5) continue;
   cout<<x;
}
cout<< "dongu bitti";</pre>
```



7. İç içe Döngüler

 Herhangi bir döngünün içinde başka döngü veya döngüler yer alabilir. Butip komutlara iç içe(nested) döngü komutu adı verilir.

7. İç içe Döngüler...

```
Örnek 4.7
#include <iostream>
using namespace std;
int main()
{
     int sayi, fakt;
     for (int sayac=1;sayac<=2;sayac++){</pre>
                                                        // Döngü 2 kere döner
          cout<<"Bir sayi giriniz:";</pre>
          cin>>sayi;
                                                        // Bir sayı okunur
          fakt=1;
                                                        // Faktoriyel hesabı 1 ile başlar
                                                        // İç döngü sayı kadar döner
          for (int i=1;i<=sayi;i++)</pre>
               fakt *= i:
                                                        // Her dönüşte i'nin değeri fakt ile çarpılır
          cout<<sayi<<"!="<<fakt<<endl;</pre>
                                                        // Sayının faktoriyeli yazdırılır
     return 0;
}
Çıktı
Bir sayi giriniz:3
3! = 6
Bir sayi giriniz:4
4!=24
```

7. İç içe Döngüler...

```
Örnek 4.8
#include <iostream>
using namespace std;
int main()
{
    char harf:
    int sayi;
    cout<<"Noktayla biten 2 cumleyi satir satir giriniz."<<endl;</pre>
    for (int i=1;i<=2;i++){
                                                  // Döngü satırlar için 2 kere döner
         sayi=0;
         cin>>harf;
                                                  // Harf okunur
         while (harf!='.'){
                                                // İç döngü . okunana kadar döner
             if ((harf=='A')||(harf=='a'))  // Harf A veya a ise
                                                  // A savisi arttırılır
                  savi++;
                                                  // Harf okunur
             cin>>harf;
         cout<<i<<".cumlede "<<sayi<<" tane A harfi var."<<endl;</pre>
    return 0;
}
Cıktı
Noktayla biten 2 cumleyi satir satir giriniz.
Ben sinemaya gitmeyi dusunuyorum.
1.cumlede 2 tane A harfi var
Beraber gidebiliriz aslinda.
2.cumlede 3 tane A harfi var
```

Soru

4. Aşağıda verilen döngüleri do-while döngü komutu ile yeniden yazınız. a. #include <iostream> using namespace std; int main() int s1, s2, s3; double ortalama; char secenek = 'e'; while((secenek != 'h') && (secenek != 'H')){ cout<<"Sayi 1:"; cin>>s1; cout<<"Sayi 2:"; cin>>s2; cout<<"Sayi 3:"; cin>>s3; ortalama = (s1 + s2 + s3) / 3;cout<<"("<<s1<<"+"<<s2<<"+"<<s3<<")/3 = "<<ortalama<<endl;</pre> cout<<"Devam etmek ister misiniz?(e/h)";</pre> cin>>secenek; return 0;

Cevap

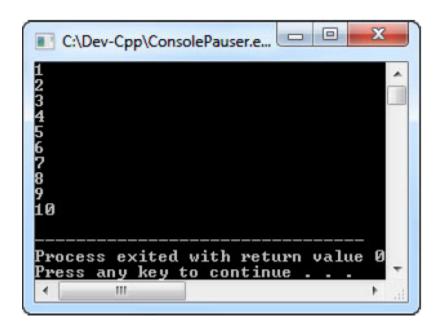
```
4.
  a. int s1, s2, s3;
       double ortalama;
       char secenek;
       do{
           cout<<"Sayi 1:";
           cin>>s1;
           cout<<"Sayi 2:";</pre>
           cin>>s2;
           cout<<"Sayi 3:";</pre>
           cin>>s3:
           ortalama = (s1 + s2 + s3) / 3;
           cout<<"("<<s1<<"+"<<s2<<"+"<<s3<<")/3 = "<<ortalama<<endl;
           cout<<"Devam etmek ister misiniz?(e/h)";</pre>
           cin>>secenek;
       }while((secenek != 'h') && (secenek != 'H'));
```

Soru

```
// Iden 10a kadar tamsayilar
#include <iostream>
using namespace std;
int i;
int main()
{
   for (i=1;i<=10;i++)
   cout << i << "\n";
   return 0;
}</pre>
```



Cevap



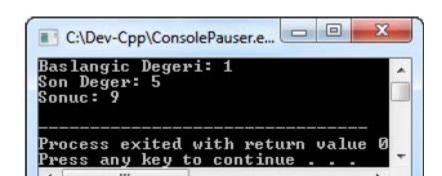


Soru

```
// girilen iki tamsayı arasında kalan
// tamsayıların toplamı
#include <iostream>
using namespace std;
int main()
    int i, baslangic, son;
    int toplam=0;
    cout << "Baslangic Degeri: ";
    cin >> baslangic;
    cout << "Son Deger: ";</pre>
    cin >> son;
// İki deger arasındaki toplam hesaplaniyor...
    for (i=baslangic+1;i<son;i++)
    toplam+=i;
    cout << "Sonuc: " << toplam
         << "\n";
return 0;
```



Cevap



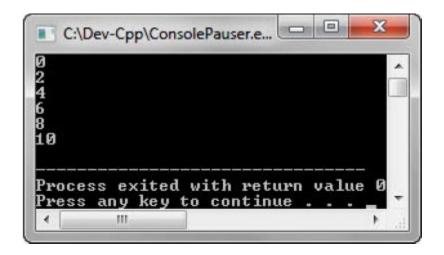


Soru

```
// Odan 10a kadar çift tamsayilar
#include <iostream>
using namespace std;
int main()
{
    int i;
    for(i=0;i<=10;i+=2) {
        cout << i << endl;
    }
}</pre>
```



Cevap



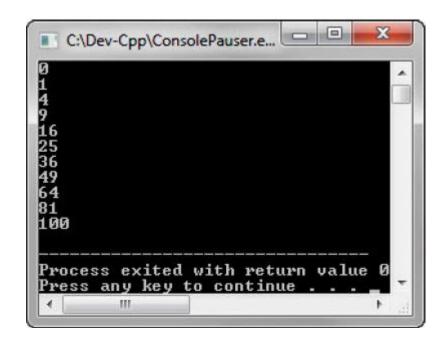


Soru

```
// Iden 10a kadar tamsayilarin kareleri
#include <iostream>
using namespace std;
int main()
{
    int i=0;
    for(;i<=10;){
        cout << i*i << endl;
        i++;
    }
    return 0;
}</pre>
```



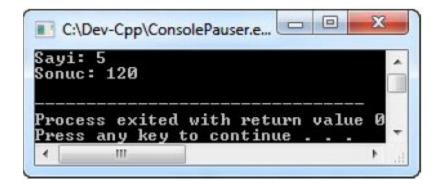
Cevap





```
#include <iostream>
using namespace std;
// verilen bir sayisal
//degerin faktoriyelinin hesaplanmasi
int main ()
    int i, sayi;
    int faktoriyel=1;
    cout << "Sayi: ";
    cin >> sayi;
// Verilen sayinin faktoriyeli hesaplaniyor..
    for (i=1;i<=sayi;i++)
    faktoriyel*=i;
    cout << "Sonuc: " << faktoriyel
    << "\n";
    return 0:
```

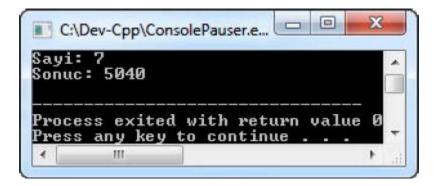






```
/* verilen bir sayisal degerin
faktoriyelinin hesaplanmasi */
#include <iostream>
using namespace std;
int main() {
       int i=1, sayi, faktoriyel=1;
      cout << "Sayi: ";
      cin >> sayi;
      while (i <= sayi)
             faktoriyel*=i;
              j++;
      cout << "Sonuc: "
      << faktoriyel</pre>
      << "\n";
       return 0;
```





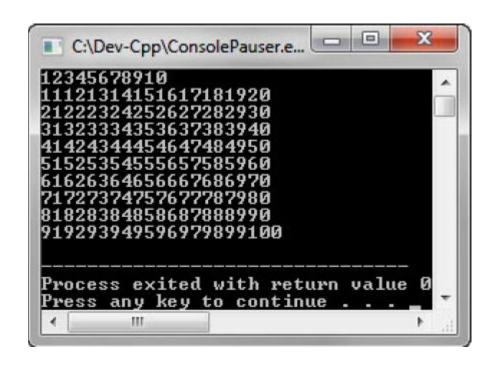


```
// Odan 100e kadar tamsayilar

#include <iostream>
using namespace std;

int main()
{
        int i=0;
        do{
            cout << ++i;
            if (i%10==0)
            cout << endl;
            } while (i<100);
        return 0;
}</pre>
```

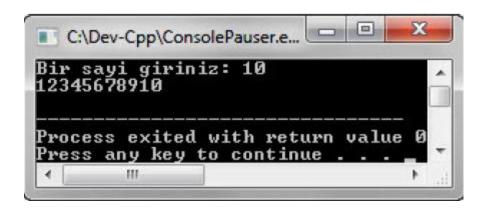






```
// girilen sayiya kadar tamsayilar
#include <iostream>
using namespace std;
int main()
       int sayi, i=0;
       cout << "Bir sayi giriniz: ";</pre>
       cin >> sayi;
       do {
              j++;
              cout << i:
              }while(i<sayi);</pre>
              cout << endl;
```

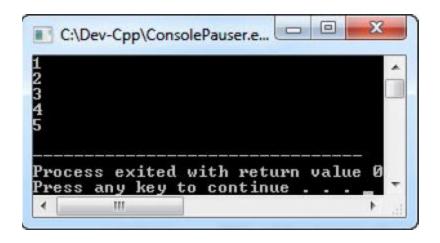






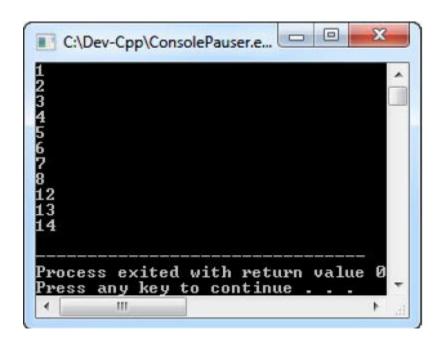
```
// break deyiminin kullanilmasi
#include <iostream>
using namespace std;
int i;
int main()
  for (i=1;i<100;i++)
      cout << i << "\n";
      if (i==5)
      break;
```









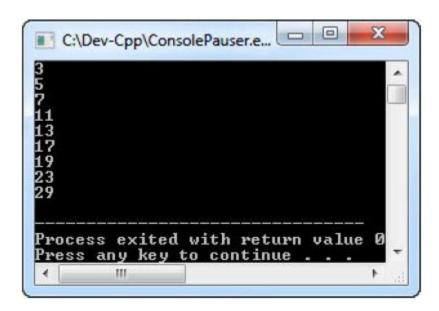




```
Soru
           /* 1 ile 30 arasindaki
           asal sayilar */
           #include <iostream>
           using namespace std;
           main()
             int i, n=2;
             while (++n \le 30)
                  i=1;
                 while (++i \le n)
                  if (n\%i==0)
                 break;
                  if (i==n)
                  cout << n
                  << "\n";
```

return 0;

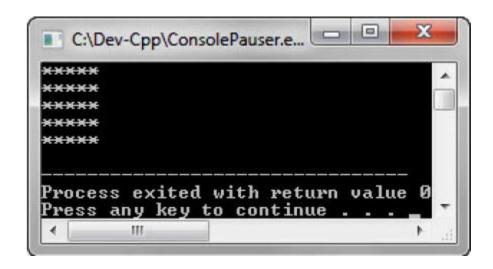






```
// icice dongulerin kullanilmasi
#include <iostream>
using namespace std;
int main()
       int i=5, j;
       while (i \ge 1) {
              j=1;
              while(j \le 5) {
                     cout << "*";
                     j++;
              cout << endl;
              i--;
```







Soru

10.Bir üçgenin kenarlarını okuyup dik açılı olup olmadığını aşağıda verilen Pisagor teorimi ile bulabiliriz.

$$h^2 = s^2 + t^2$$

kullanıcı 'H' karakteri girince program sona erer.

Belirsiz sayıda üçgenin s, t ve h kenarlarını okuyup dik açılı olup olmadıklarını bulan bir program yazınız. Örnek çıktıda görüldüğü üzere, her üçgenden sonra kullanıcıya devam edelim mi sorusu sorulur ve program

Örnek Çıktı

Ucgenin iki kenarini giriniz: 3 4 Ucgenin hipotenusunu giriniz: 5

Dik acili ucgen

Devam etmek ister misiniz (E/H): H



```
10. #include <iostream>
   #include <math.h>
   using namespace std;
   int main()
       int s, t, h; char secim;
       do{
            cout<<"Ucgenin iki kenarini giriniz: ";</pre>
            cin>>s>>t;
            cout<<"Ucgenin hipotenusunu giriniz: ";</pre>
            cin>>h:
            if ((sqrt(s*s+t*t))==h) cout<<"Dik acili ucgen"<<endl;</pre>
            cout<<"Devam etmek ister misiniz (E/H):";</pre>
            cin>>secim;
         }while((secim !='H') || (secim !='h'));
         return 0;
```