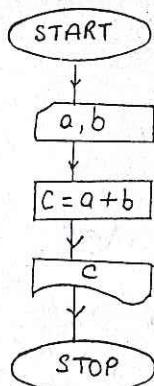


ALGORİTMALAR 27.11.15

İki sayının toplamını bulan algoritma ve programı.

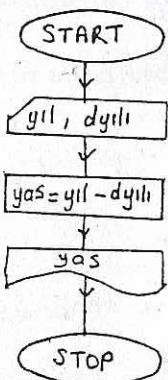


BÖLÜM 1

```

Program toplam (input, output);
uses crt;
var
  a, b, c : integer;
begin
  clrscr;
  writeln ('iki sayı giriniz');
  readln (a,b);
  c := a+b;
  writeln (c);
  readln;
end.
  
```

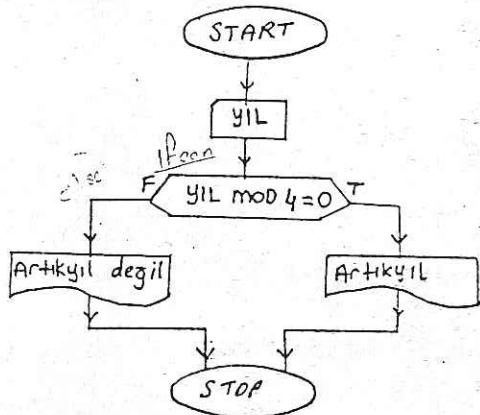
Dogum tarihi verilen kişinin yosunu bulan algoritma ve programı.



```

Program yos (input, output);
uses crt;
var
  yasiniz, yıl, dyili : integer;
begin
  clrscr;
  writeln ('Dogum tarihiniizi giriniz');
  readln (dyili);
  writeln ('yılı giriniz');
  readln (yıl);
  yasiniz := yıl - dyili;
  write (yasiniz);
  read;
end.
  
```

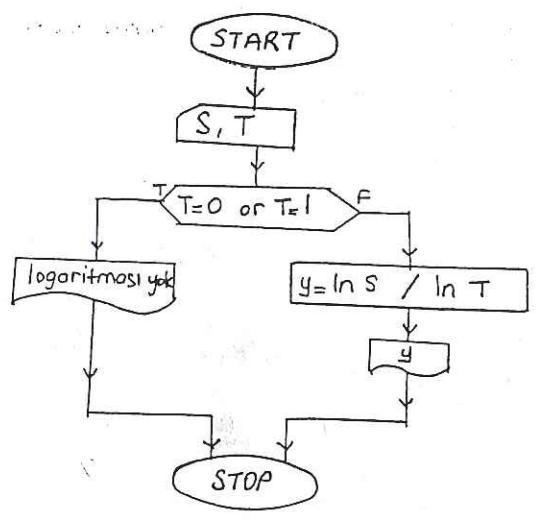
Verilen yılın artıkyıl olup olmadığını bulan algoritma ve programı.



```

Program artıkyıl (input, output);
uses crt;
var
  yıl : integer;
begin
  clrscr;
  writeln ('yılı giriniz');
  readln (yıl);
  if yıl mod 4 = 0 then
    writeln ('Girdığınız yıl artıktır')
  else
    writeln ('Girdığınız yıl artık değildir');
end.
  
```

Verilen bir S sayısının, T tabanına göre logaritmasını bulan algoritma ve programı.



Program logaritma (input, output);
uses crt;

var

$s, t, y : real$;

Begin

clrscr;

writeln ('sayınızı giriniz');
readln (s);

writeln ('Tabanı giriniz');
readln (t);

if ($T=0$) or ($T=1$) then
writeln ('logaritmosi yoktur')

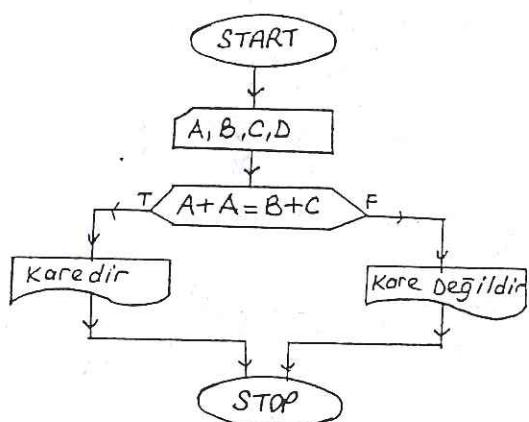
else
 $y := \ln(s) / \ln(t)$;

writeln (y);

End.

$y = \ln(s) / \ln(t)$

Kenarları A, B, C, D olarak verilen bir dörtgenin kare olup olmadığını bulan algoritma ve programı.



Program Koreni (input, output);
uses crt;

Var

$a, b, c, d : integer$;

begin

clrscr;

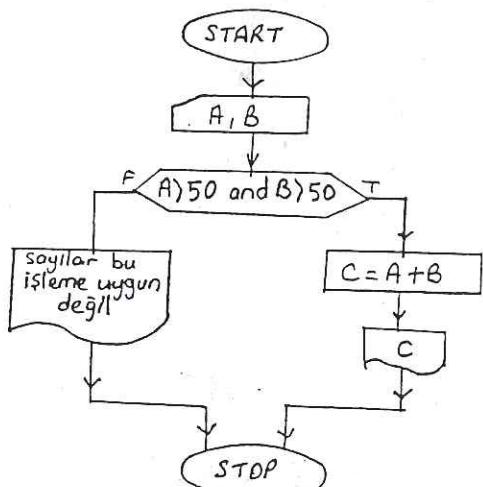
writeln ('Dörtgenin kenar uzunluklarını gir');
readln (a, b, c, d);

if $(a+a) = (b+c)$ then
writeln ('Dörtgen karedir')

else
writeln ('Dörtgen kare değildir');

End.

A ve B sayıları 50 'den büyük ise $C = A+B$ işlemini yapan algoritma ve programı.



Program ellidenbüyük (input, output);
uses crt;

Var

$a, b, c : integer$;

Begin

clrscr;

writeln ('sayıları giriniz');
readln (a, b);

if $(a > 50)$ and $(b > 50)$ then
begin

$c := a+b$;

writeln (c);

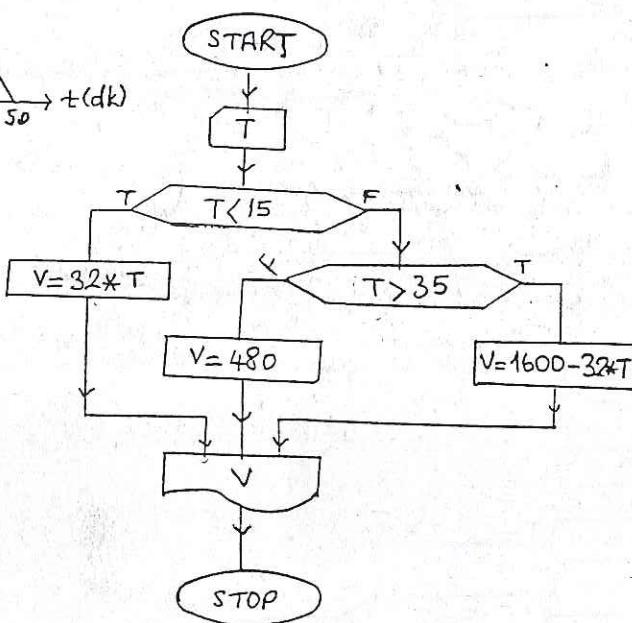
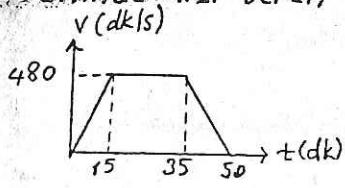
end

else

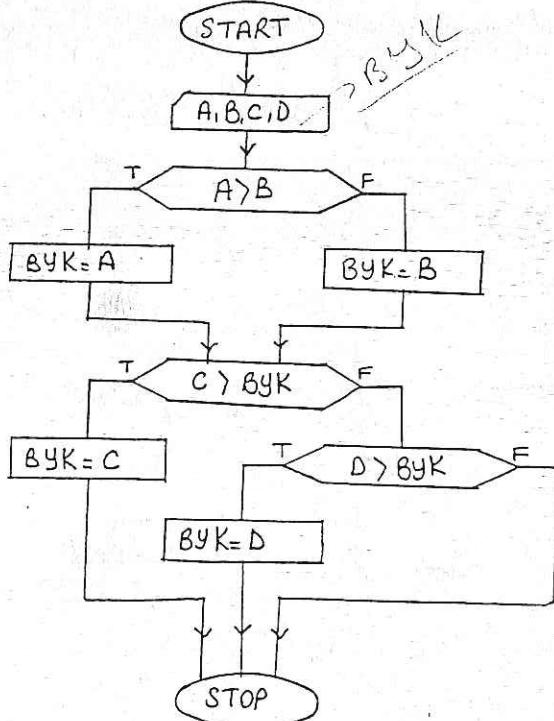
writeln ('sayılar bu işlemeye uygun değil');

End.

Bir uçak 15 dk boyunca düzgün hızlanarak hızı 480 km/s oluyor. Sonra 20 dk sabit hızla gidiyor ve 15 dk boyunca düzgün yavaşlayarak hızı sıfır oluyor. Herhangi bir t anında hızı veren algoritma ve programı.



A, B, C, D sayılarından en büyüğünü bulan algoritma ve programı.



```

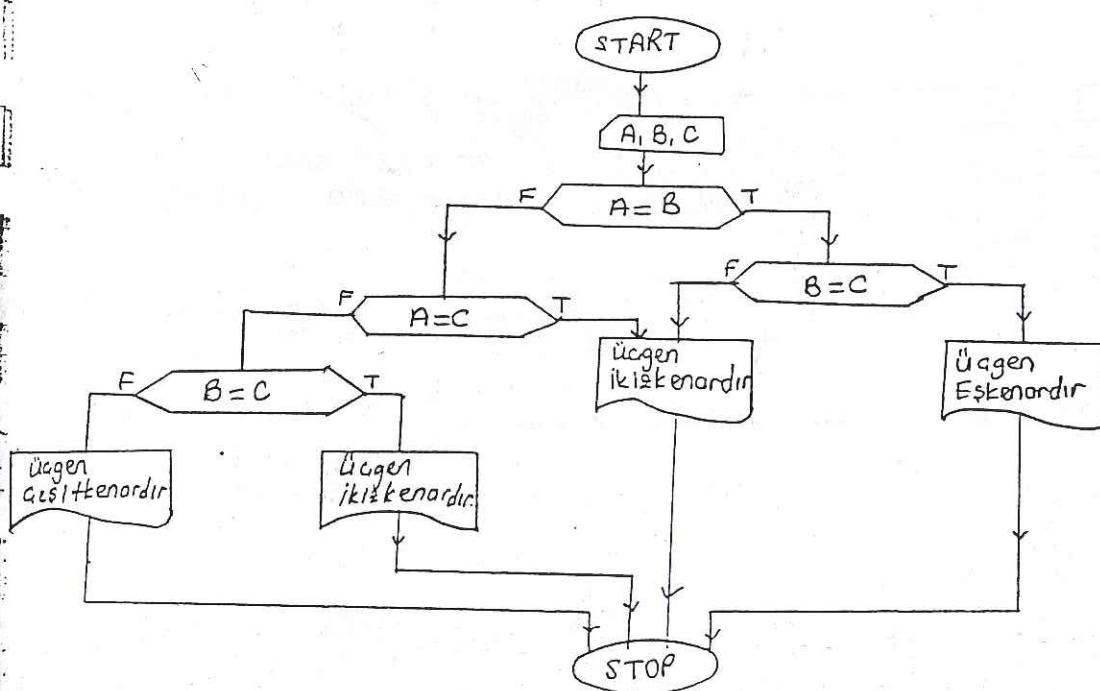
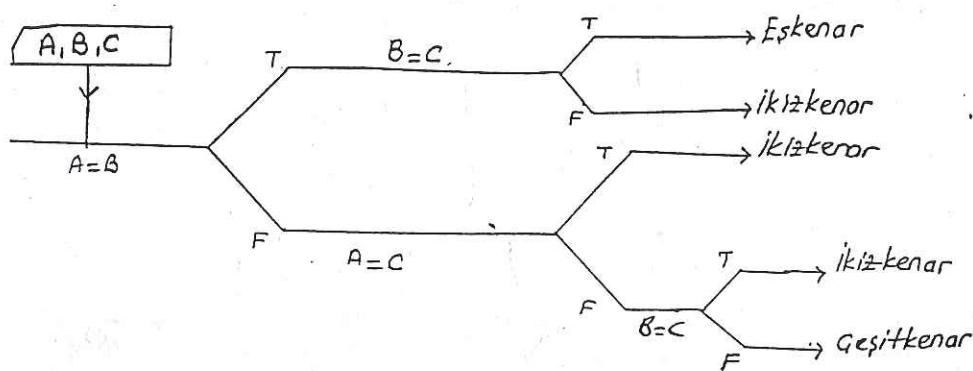
Program ucak (input, output);
uses crt;
var
  t, v : integer;
begin
  clrscr;
  writeln ('suregi dakika cinsinden giriniz');
  Readln (t);
  if t < 15 then
    v := 32 * t
  else
    begin
      if t > 35 then
        v := 1600 - (32 * t)
      else
        v := 480;
    end;
  writeln (v);
End.
  
```

```

Program enbuyuk (input, output);
uses crt;
var
  a, b, c, d, byk : integer;
begin
  clrscr;
  writeln ('sayiları giriniz');
  Readln (a, b, c, d);
  if a > b then
    byk := a
  else
    byk := b;
  if c > byk then
    byk := c
  else
    begin
      if d > byk then
        byk := d;
    end;
  writeln (byk);
End.
  
```

İç açıları verilen üçgenin karar ağacı, algoritması ve programı.

Karar Ağacı:



```

Program ücgen (input, output);
uses crt;
var
  a,b,c : real;
begin
  clrscr;
  writeln ('Üçgenin iç açılarını giriniz?');
  readln (a,b,c);
  if a=b then
    begin
      if b=c then
        writeln ('Üçgen eskenardır')
      else
        writeln ('Üçgen ikizkenardır');
    end;
  else
    begin
      if a=c then
        writeln ('Üçgen ikizkenardır')
      else
        begin
          if b=c then
            ...
  
```

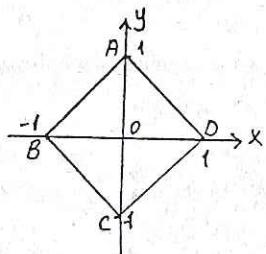
```

else
  writeln ('Üçgen gesittenardır');
end;
end;
end.

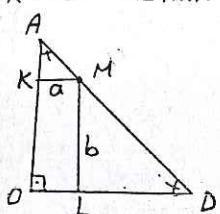
```

Verilen bir noktanın koordinatları (x, y) 'dir. Bu noktanın $|x| + |y| = 1$ grafğının
neresinde bulunduğuunu bulan algoritma, ve programı.

Bilgi: $|x| + |y| = 1$ denklemiñin grafiñi osağridaki gibidir.



$\triangle AOD$ üçgenini ele alırsak; $A = D = 45^\circ$ dir. Üçgen ikizkenardır. Bu üçgen üzerinde bir M
noktasi olsun.

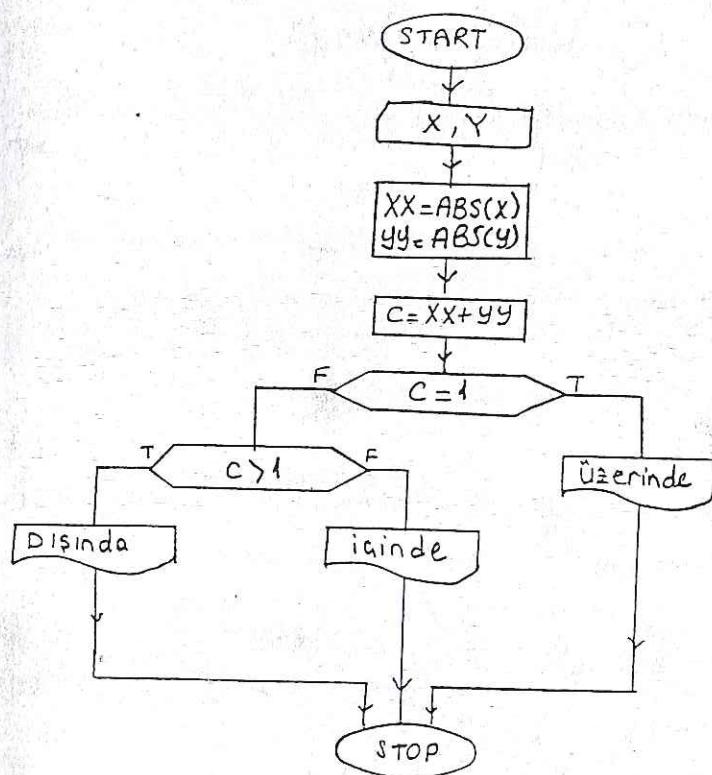


$$|AK| = |OL| = |KM| = a$$

$$|ML| = |KO| = |LD| = b \text{ dir.}$$

$$|AK| + |KO| = a + b = 1 \text{ dir.}$$

Sonuç olarak, bu dörtgen üzerinde hangi noktası alırsak alalım koordinatları
toplamı 1'dir.

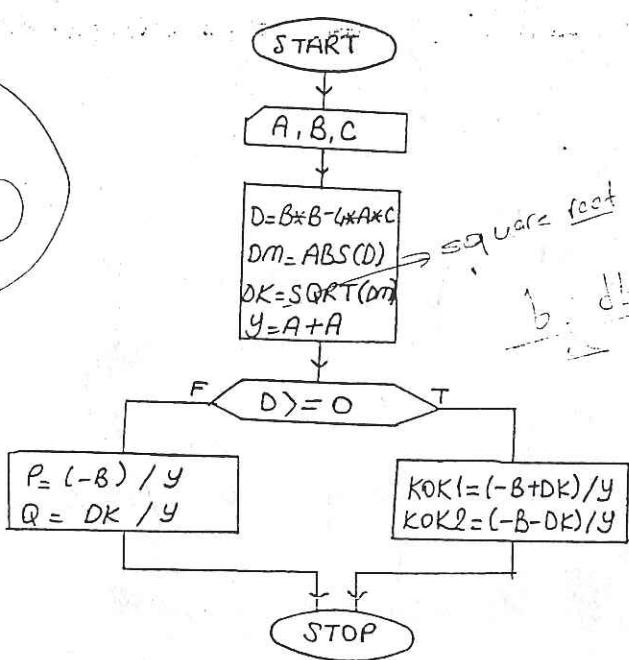
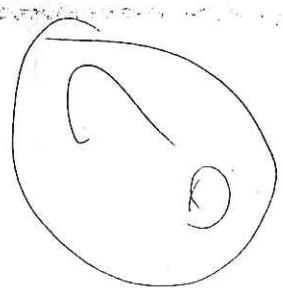


```

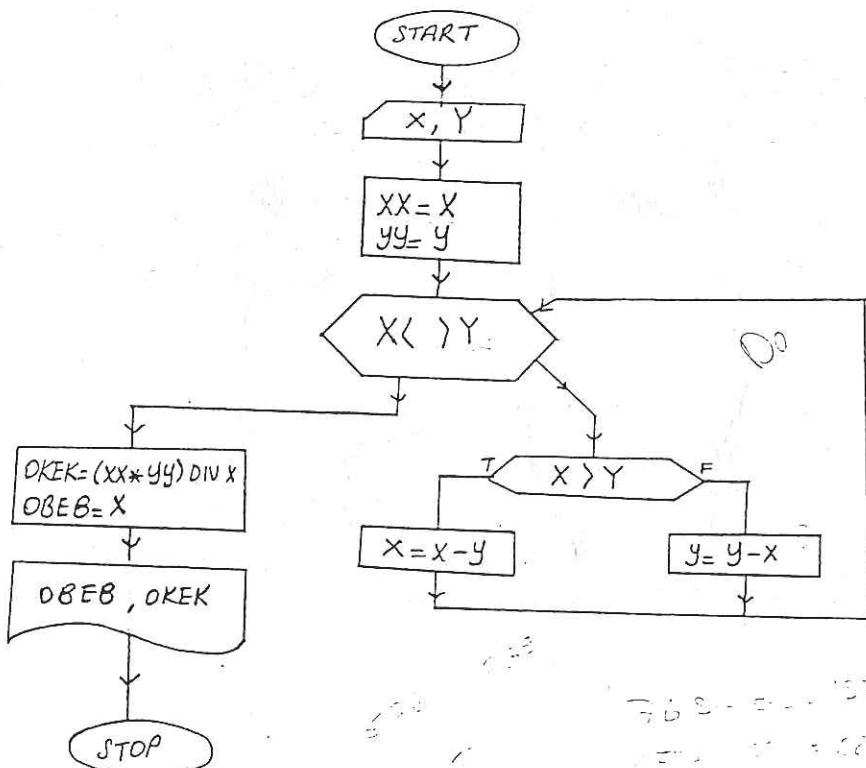
Program Koordinat (input, output);
uses crt;
var
  x, xx, y, yy, c: Integer;
begin
  clrscr;
  writeln ('Koordinatları giriniz');
  readln (x,y);
  xx := abs(x);
  yy := abs(y);
  c := xx+yy;
  if c = 1 then
    writeln ('üzerinde')
  else
    begin
      if c>1 then
        writeln ('dısında')
      else
        writeln ('icinde');
    end;
  End.

```

$AX^2 + BX + C = 0$ denkleminin köklerini bulan algoritma ve programı.



OBEBOKEK bulan algoritma ve programı.



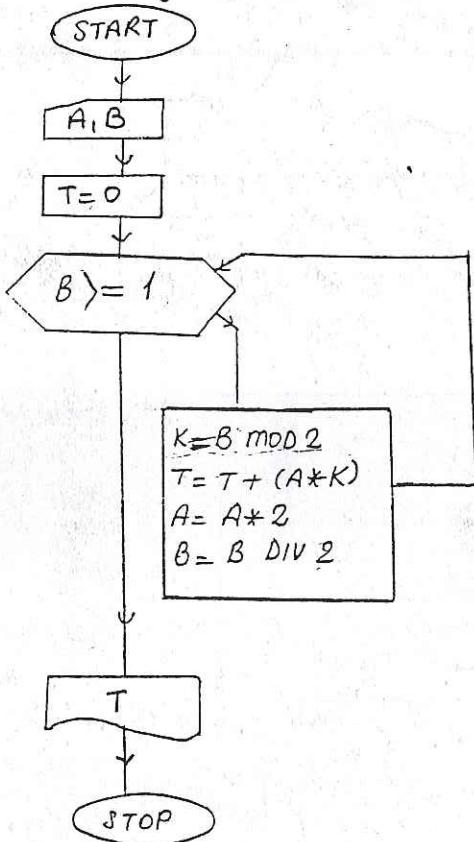
```

Program KATBULMA(input, output);
uses crt;
var
  a,b,c,d,dm,dk,y,p,q,kok1,kok2:real;
begin
  clrscr;
  writeln('a,b,c sayılarını giriniz');
  Readln(a,b,c);
  d := (b*b) - (4*a*c);
  dm := abs(d);
  dk := sqrt(dm);
  y := a + dk;
  if d >= 0 then
    begin
      KOK1 := (-b + dk) / y;
      KOK2 := (-b - dk) / y;
    end
  else
    begin
      p := (-b) / y;
      q := (dk) / y;
      writeln(p:3, q:3);
    end;
  end.
  
```

```

Program OBEBOKEK(input, output);
uses crt;
Var
  x, y, xx, yy, obeb, okek:integer;
Begin
  clrscr;
  writeln('sayıları giriniz');
  Readln(x,y);
  xx := x;
  yy := y;
  While x < y do
  begin
    if x > y then
      x := x - y
    else
      y := y - x;
  end;
  okek := (xx * yy) div x;
  obeb := x;
  Write( obeb:3, okek:3);
  Read;
End.
  
```

iki sayı giriliyor. I. sayı iki ile çarpılırken, II. sayı ikiye bölünüyor. Bu işlem II. sayının bir olmasına kadar devam ediyor. II. sayı tek ise andaki I. sayının değerleri toplanıyor. Toplamanın sonucu bize çarpmayı verir. Bu işlemin algoritma ve programı. (iki sayı arasında doğrudan çarpma işlemi yapmadan çarpanın yolu).



Program Çarpma (input, output);

uses crt;

Var

a,b,t,k:integer;

Begin

clrscr;

Writeln ('iki sayı giriniz');

Readln (a,b);

t:=0;

While b>=1 do

begin

k:=b mod 2;

t:=t+(a*k);

a:=a*2;

b:=b div 2;

end;

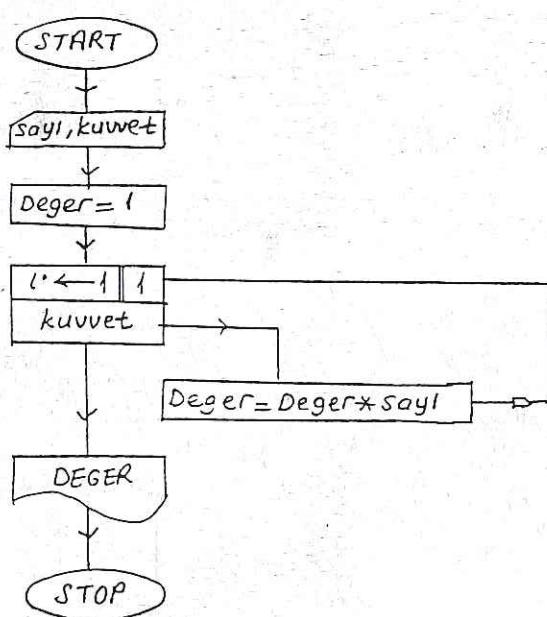
Writeln (t);

Readln;

16 - 18

End.

Herhangi bir sayının herhangi bir dereceden kuvvetini bulan program ve algoritma



Program ustel (input, output);

uses crt;

var

sayi, kuvvet, deger, i:integer;

begin

clrscr;

Writeln ('sayiyi giriniz');

Readln (sayi);

Writeln ('Kuvveti giriniz');

Readln (kuvvet);

deger:=1;

for i:=1 to kuvvet do,

deger:=deger * sayi;

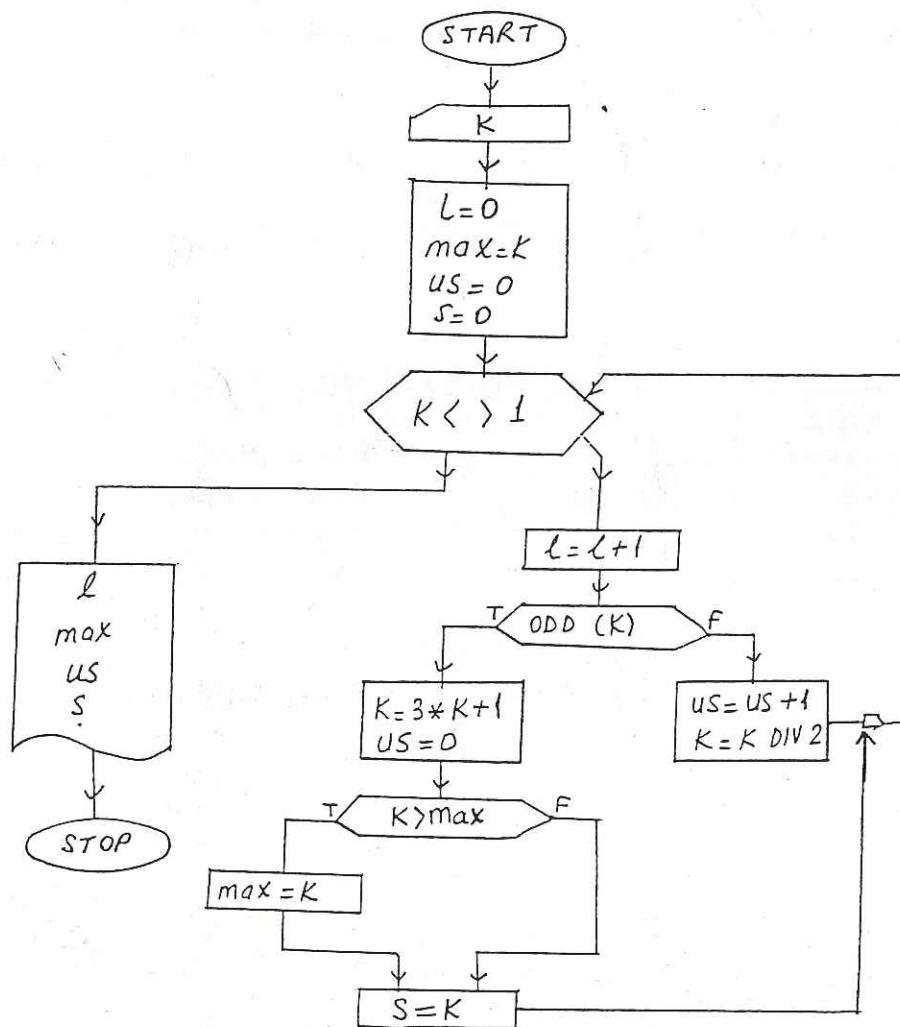
Writeln (deger);

Readln;

End.

(7)

Bir K sayısı tek ise 3 ile çarپılıp 1 ekleniyor; çift ise 2'ye bölünüyor. İşlem K sayısı 1 olana kadar tekrar ediliyor. Bu işlemin kaç adım sürdüğünü, işlem sırasında K sayısının aldığı maximum değeri, K sayısının hangi sayıdan sonra hep çift olarak 1'e ulaştığını ve bu sayının ikinci kezine kuvveti olduğunu bulan algoritma ve programı.



Program KSayisi (input, output);

uses crt;

var

k, l, us, max, s : integer;

Begin

crtscr;

Writeln ('k sayısını giriniz');

Readln (k);

$l := 0$; $max := k$; $us := 0$; $s := 0$;

while $K <> 1$ do

begin

$l := l + 1$;

if odd (k) then

begin

$k := (3 * l) + 1$;

$us := 0$;

if $k > max$ then

$max := k$;

$s := k$;

end

begin

$us := us + 1$;

$k := k \text{ div } 2$;

end;

end;

Write ($l:3$, $max:3$, $s:3$, $us:3$);

End.

Analiz: $K = 13$ sayısı (input)

$l = 1$ $K \rightarrow$ tek $\rightarrow K = 3 * 13 + 1 = 40$ $max = 40$ $s = 40$

$l = 2$ $K \rightarrow$ çift $\rightarrow K = 40 / 2 = 20$ $us = 1$

$l = 3$ $K \rightarrow$ çift $\rightarrow K = 20 / 2 = 10$ $us = 2$

$l = 4$ $K \rightarrow$ çift $\rightarrow K = 10 / 2 = 5$ $us = 3$

$l = 5$ $K \rightarrow$ tek $\rightarrow K = 16$ $us = 0$ $s = 16$

$l = 6$ $K \rightarrow$ çift $\rightarrow K = 16 / 2 = 8$ $us = 1$

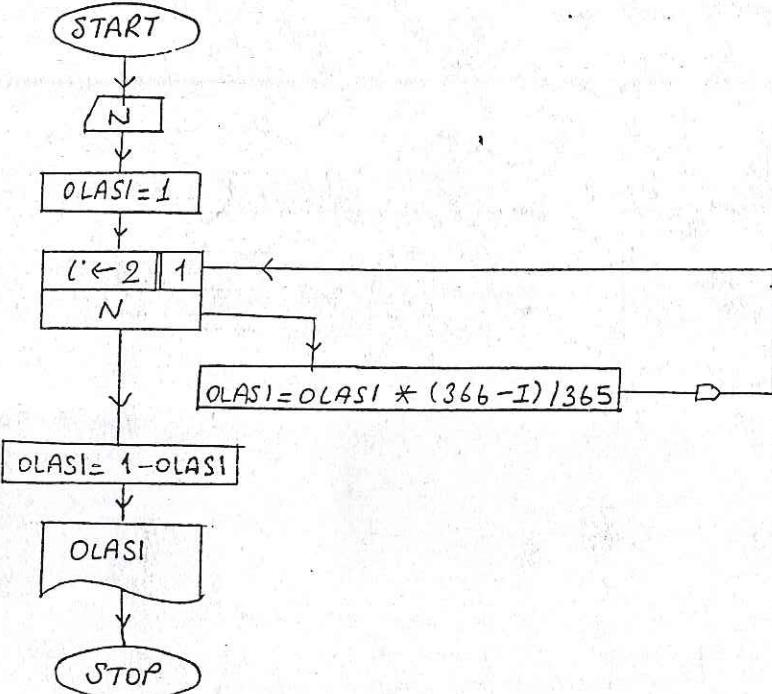
$l = 7$ $K \rightarrow$ çift $\rightarrow K = 8 / 2 = 4$ $us = 2$

$l = 8$ $K \rightarrow$ çift $\rightarrow K = 4 / 2 = 2$ $us = 3$

$l = 9$ $K \rightarrow$ çift $\rightarrow K = 2 / 2 = 1$ $us = 4$

Output: $l = 9$, $max = 40$, $s = 2^9 = 16$, $us = 4$

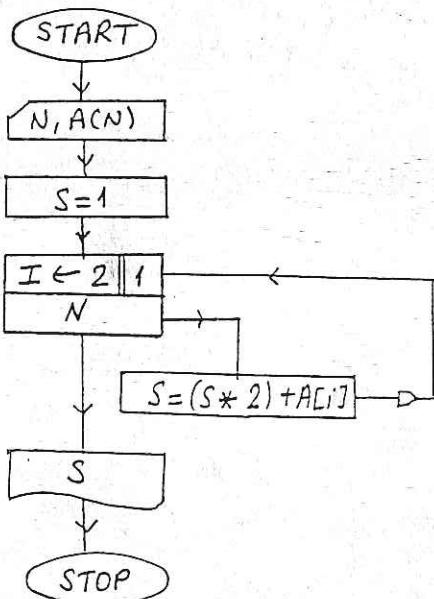
Bir gruptaki kişilerin enaz iki tanesinin yaşı günlerinin aynı güne gelme olasılığı
 $1 - \left[\frac{366-2}{365} * \frac{366-3}{365} * \frac{366-4}{365} * \dots * \frac{366-N}{365} \right]$ seklinde dir. Buna göre 60 kişilik bir grupta iki kişinin yaşı günlerinin aynı olma olasılığını hesaplayan algoritma ve programı.



```

Program olasilik(input,output);
uses crt;
Var
  n,i:integer;
  olasi:real;
Begin
  clrscr;
  writeln('Gruptaki kişi sayısı');
  Readln(n);
  olasi := 1;
  for i:= 2 to n do
    olasi := olasi * (366-i)/365;
  olasi := 1 - olasi;
  writeln(olasici);
  Readln;
End.
  
```

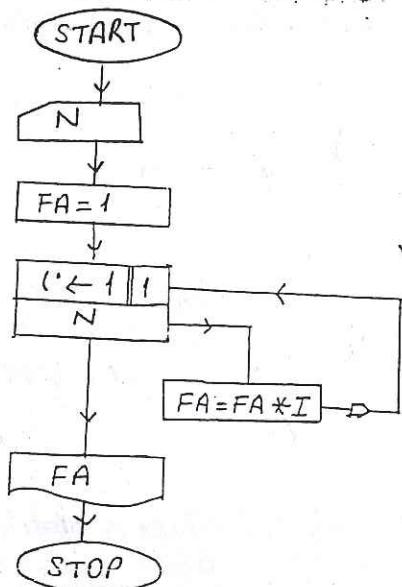
Binary bir sayıyı decimal bir sayıya çeviren algoritma ve programı.
(2 tabanındaki bir sayıyı 10 tabanına çeviren algoritma ve programı.)



```

Program cevirme(input,output);
uses crt;
var
  s,n,i:integer;
  a:array [1..10] of integer;
Begin
  clrscr;
  writeln('Dizinin boyutunu gir');
  Readln(n);
  write('Dizi elemanlarını gir');
  FOR i:= 1 to n do
    Read (a[i]);
  s:= 1 ;
  for i:= 2 to n do
    s:= (s*2)+a[i];
  writeln(s);
End.
  
```

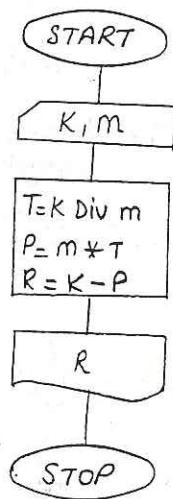
Faktöriyel Hesaplayan Algoritma ve Programı.



```

Program faktoriyel (input, output);
uses crt;
Var
  n, fa, i : integer;
begin
  clrscr;
  writeln ('faktöriyeli alınacak sayıyı gir');
  Readln (n);
  fa:=1;
  for i:=1 to n do
    fa:=fa*i;
  writeln (fa);
End.
  
```

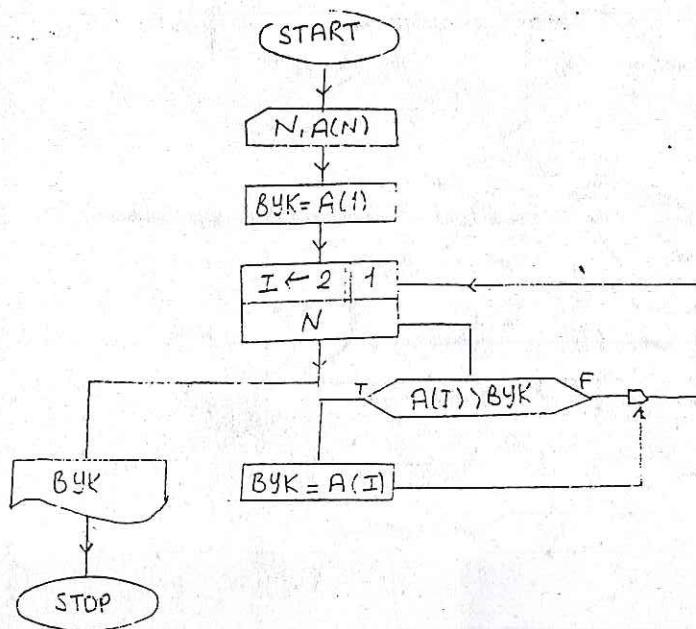
Mod alan algoritma, ve programı.



```

Program ModAlma (input, output);
uses crt;
var
  k, m, t, p, r : integer;
begin
  clrscr;
  writeln ('Sayıyı giriniz');
  Readln (k);
  writeln ('modu giriniz');
  Readln (m);
  t := k div m;
  p := m * t;
  r := k - p;
  writeln (r);
end.
  
```

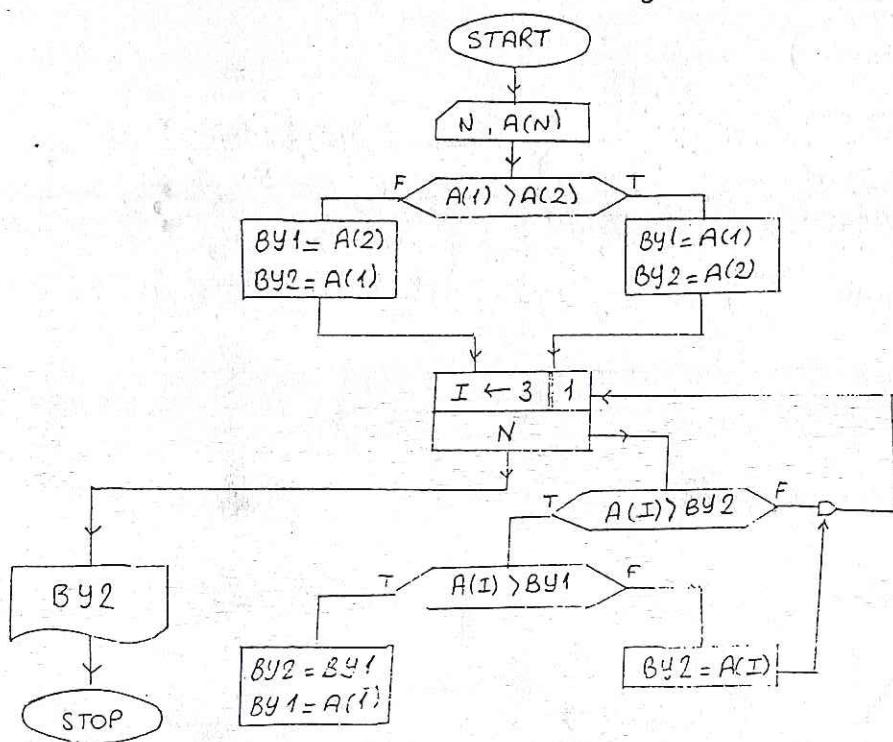
Bir dizinin en büyük elemanını bulan algoritma ve programı.



```

Program enbuyuk(input, output);
uses crt;
var
  n, i: integers;
  A: Array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizinin boyutunu gir');
  Readln (n);
  write ('Dizi elemanlarını gir');
  for i:=1 to n do
    read (A[i]);
  BYK := A[1];
  for i:= 2 to n do
    begin
      if A(i) > BYK then
        BYK := A[i];
    end;
  writeln (BYK); End.
  
```

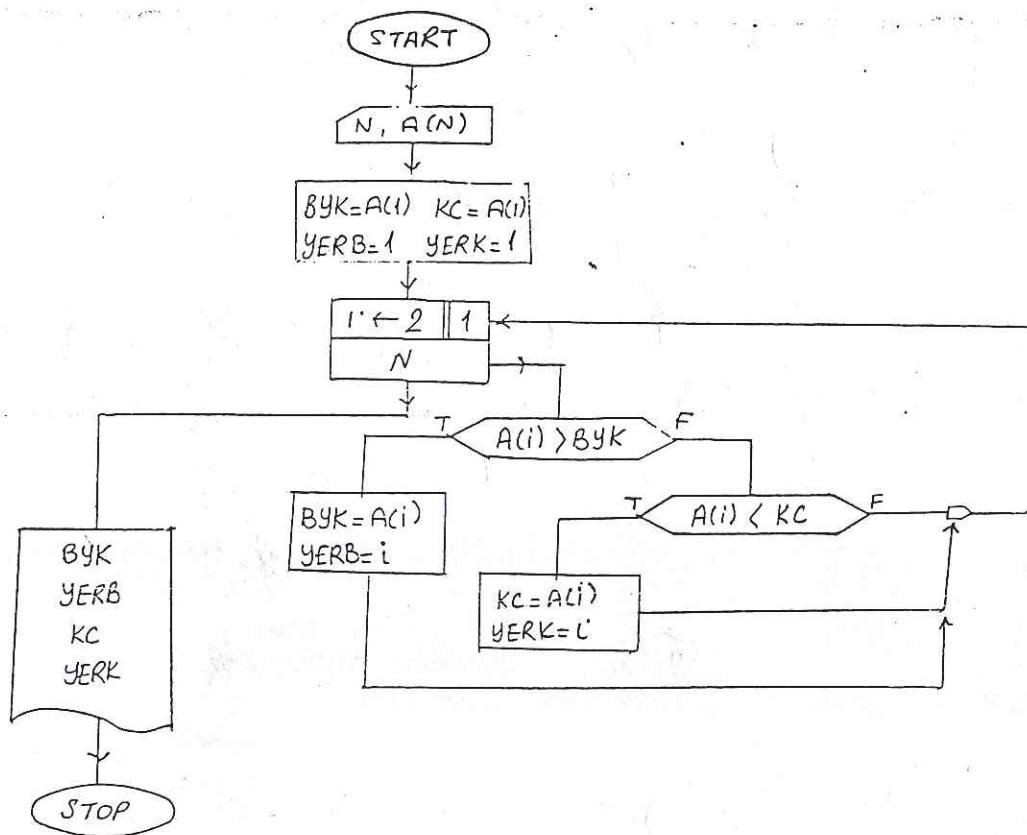
Bir dizinin ikinci büyük elemanını bulan algoritma ve programı.



```

Program ikincibuyuk(input, output);
uses crt;
var
  BY1, BY2, n, i: integer;
  A: Array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu gir');
  Readln (n);
  writeln ('Dizi elemanlarını gir');
  for i:= 1 to n do
    read (A[i]);
  if A(1) > A(2) then
    begin
      BY1 := A(1);
      BY2 := A(2);
    end
  else
    begin
      BY1 := A(2);
      BY2 := A(1);
    end;
  for i:= 3 to n do
    begin
      if A(i) > BY1 then
        begin
          BY2 := BY1;
          BY1 := A(i);
        end
      else
        BY2 := A(i);
    end;
  writeln (BY2); End.
  
```

Bir dizinin en büyük ve en küçük elemanlarını ve yerini bulan algoritma ve program.



Program büyükKucuk (input, output);
uses crt;
Var

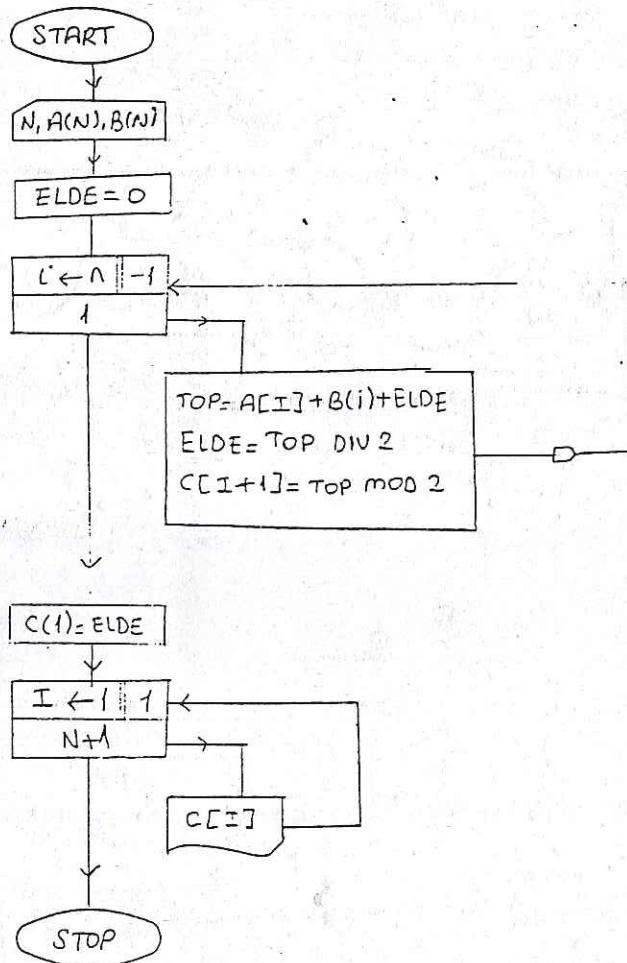
```

Yerb, Yerk, byk, kc, i, n: integer;
A: Array [1..10] of integer;
Begin
  clrscr;
  writeln ('Dizi kaç elemandır?');
  Readln (n);
  writeln ('Elemanları giriniz');
  For i:= 1 to n do
    Readln (A[i]);
  BYK := A(1); KC := A(1);
  YERB := 1; Yerk := 1;
  For i := 2 to n do
    Begin
      if A[i] > BYK then
        begin
          BYK := A(i);
          YERB := i;
        end
      else
        begin
          if A[i] < KC then
            begin
              KC := A(i);
              YERK := i;
            end;
        end;
    end;
  end;

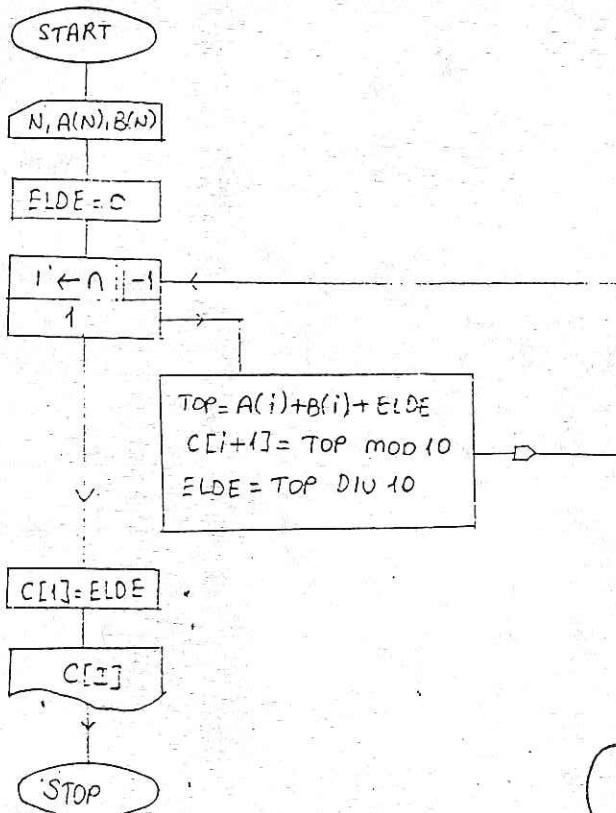
```

writeln (Byk:4, Yerb:4, KC:4, Yerk:4);
End.

İkilik düzende aynı basamaklı iki sayı verildiğinde bunların toplamını yapan algoritma ve programı.



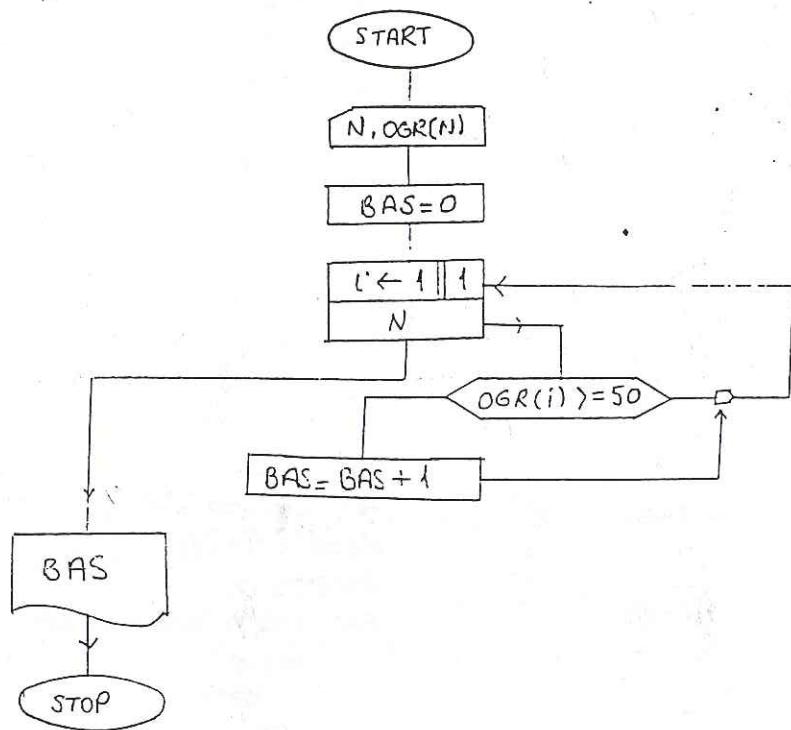
Her bir digit (hanesi) bir dizinin bir elemanı olarak verilen N digitlik iki sayı arasında toplama işlemi yaparak elde edeceğiniz değerini yeni bir dizeye yerleştiren algoritma ve programı.



Program ikilikdizeleri (input, output)
uses crt;
var
n, i, top, elde : integer;
a, b : array [1..10] of integer;
c : array [1..11] of integer;
Begin
clrscr;
writeln ('Dizi boyutunu gir');
readln (n);
write ('A dizisini gir');
for i := 1 to n do
read (A[i]);
write ('B dizisini gir');
for i := 1 to n do
read (B[i]);
elde := 0;
for i := n downto 1 do
begin
top := A[i] + B[i] + elde;
elde := top div 2;
c[i+1] := top mod 2;
end;
c[1] := elde;
for i := 1 to n+1 do
write (c[i]);
End.

Program onikitoplama (input, output)
uses crt;
var
n, i, elde, top : integer;
a, b : array [1..10] of integer;
c : array [1..11] of integer;
Begin
clrscr;
writeln ('Dizi boyutunu giriniz');
readln ('n');
write ('A dizisini gir');
for i := 1 to n do
read (A[i]);
write ('B dizisini gir');
for i := 1 to n do
read (B[i]);
elde := 0;
for i := n downto 1 do
begin
top := A[i] + B[i] + elde;
c[i+1] := top mod 10;
elde := top div 10;
end.

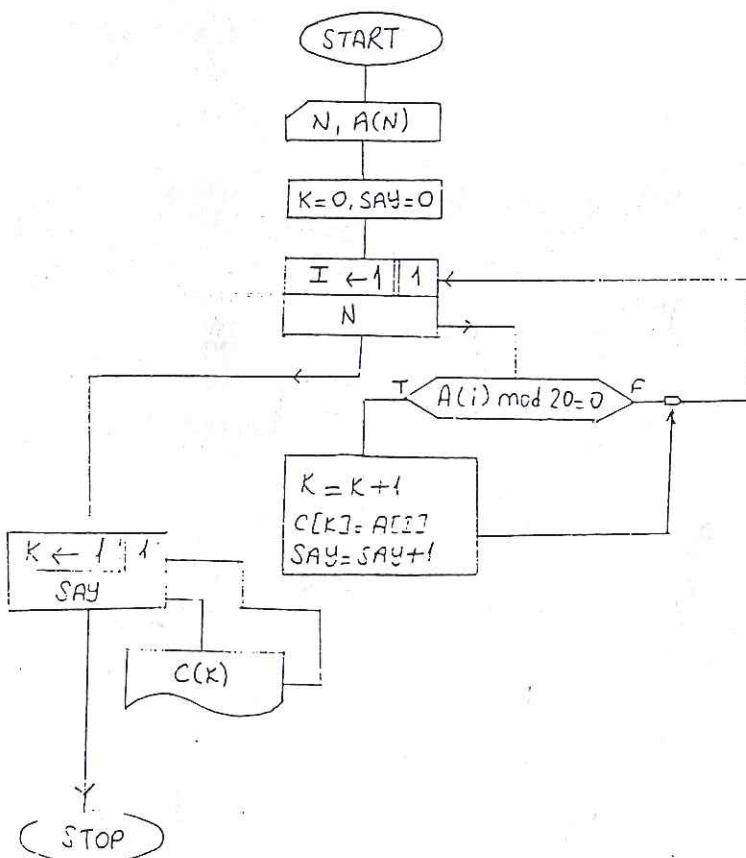
Bir sınıfındaki 50 öğrencinin bir dersden aldığı yıl sonu notları veriliyor. Başarı notu 50 olduğuna göre kaç öğrencinin başarılı olduğunu bulan algoritma ve programı.



```

Program basariliogrenci (input, output)
uses crt;
var
  n, bas, i : integer;
  ogr : array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi kaç elemanlı?');
  readln (n);
  write ('Elemanları gir');
  for i := 1 to n do
    read (ogr[i]);
  BAS := 0;
  for i := 1 to n do
    begin
      if OGR[i] ≥ 50 then
        BAS := BAS + 1;
    end;
  writeln (BAS);
end.
  
```

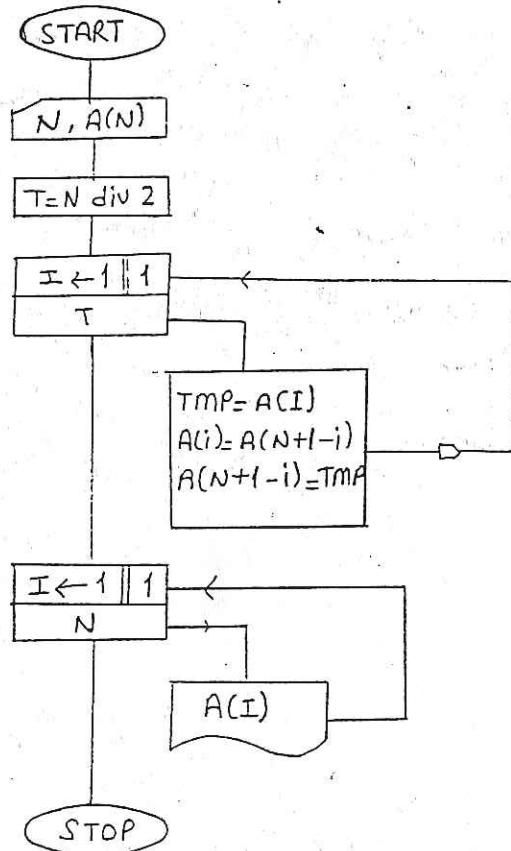
Bir dizide hem 4'e hem 5'e bölünebilen sayıları bulan algoritma ve programı.



```

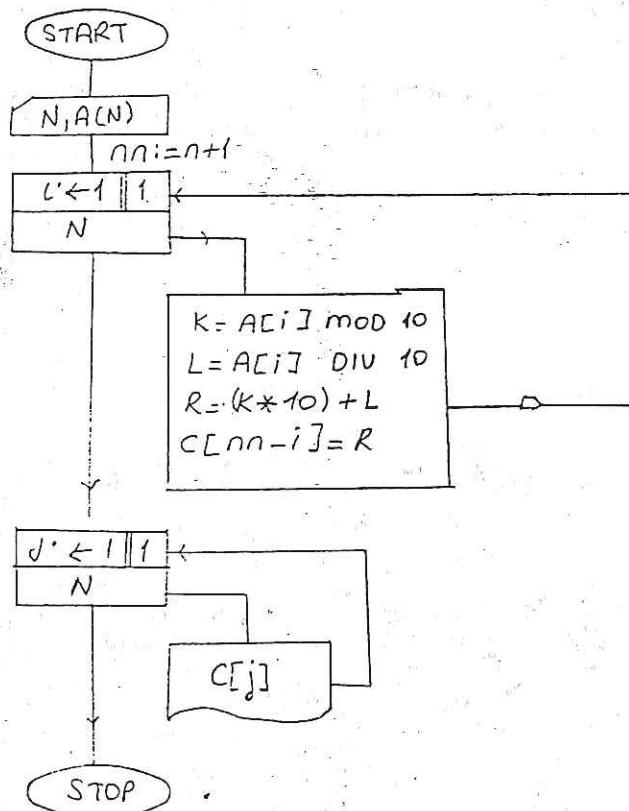
Program yirmiyebolunme (input, output)
uses crt;
var
  n, k, say, i : integer;
  a, c : array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizinizin boyutu nedir?');
  readln (n);
  write ('Diziyi gir');
  for i := 1 to n do
    read (a[i]);
  k := 0; say := 0;
  for i := 1 to n do
    begin
      if a[i] mod 20 = 0 then
        begin
          k := k + 1;
          c[k] := a[i];
          say := say + 1;
        end;
    end;
  for k := 1 to say do
    write (c[k]:5);
end.
  
```

Bir dizide dizî elemanlarının sondan başa gelerek şekilde düzenlenmesini sağlayan algoritma ve programı.



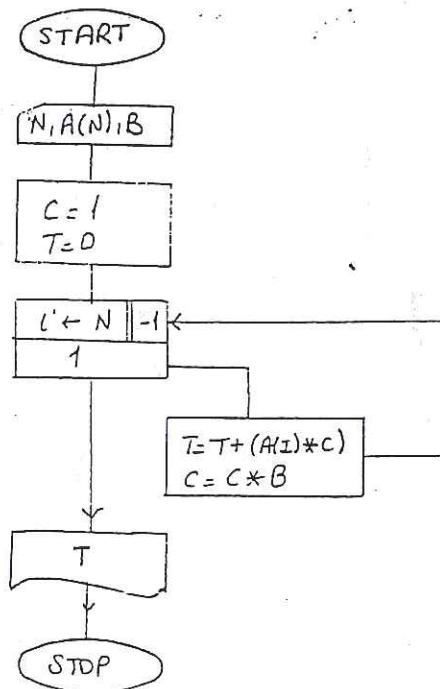
Program sondan basa (input, output) uses crt;
var
i, n, t, tmp: integer;
a: array [1..10] of integer;
Begin
clrscr;
writeln ('Dizi boyutunu giriniz');
Readln (n);
Write ('Dizi elemanlarını giriniz');
For i:=1 to n do
read (a[i]);
t := n div 2;
for i := 1 to t do
begin
tmp := a[i];
a[i] := a[n+1-i];
a[n+1-i] := tmp;
end;
for i := 1 to n do
write (a[i]:4);
End.

47-29-34-11 dizisinden 11-43-92-74 dizisini elde eden algoritma ve programı.



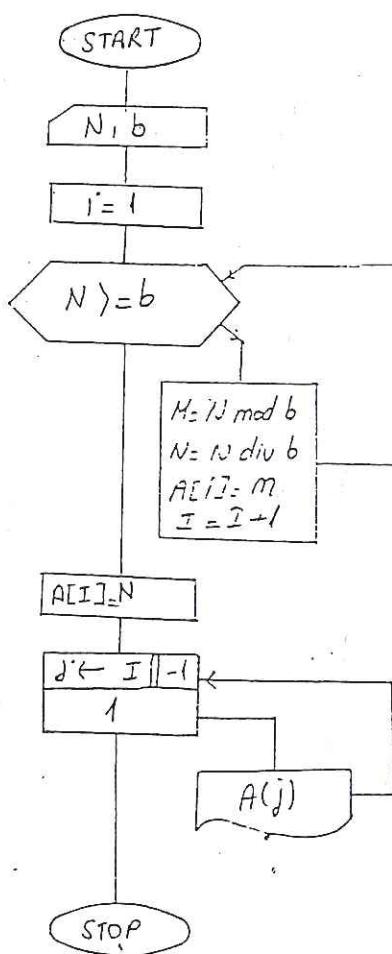
Program yeridizi (input, output) uses crt;
var
j, n, i, k, l, r, nn: integer;
a, c: array [1..10] of integer;
Begin
clrscr;
writeln ('Dizi boyutunu giri');
Readln (n);
Write ('Dizi elemanlarını giri');
FOR i:=1 to n do
Read (a[i]);
nn:=n+1;
for i:=1 to N do
begin
k := a[i] mod 10;
l := a[i] div 10;
r := (k*10)+l;
c[nn-i]:=r;
end;
for j:=1 to n do
write (c[j]:3);
End.

Herhangi bir düzendeki bir sayının onluk düzene çevrilmesini sağlayan algoritma ve programı.



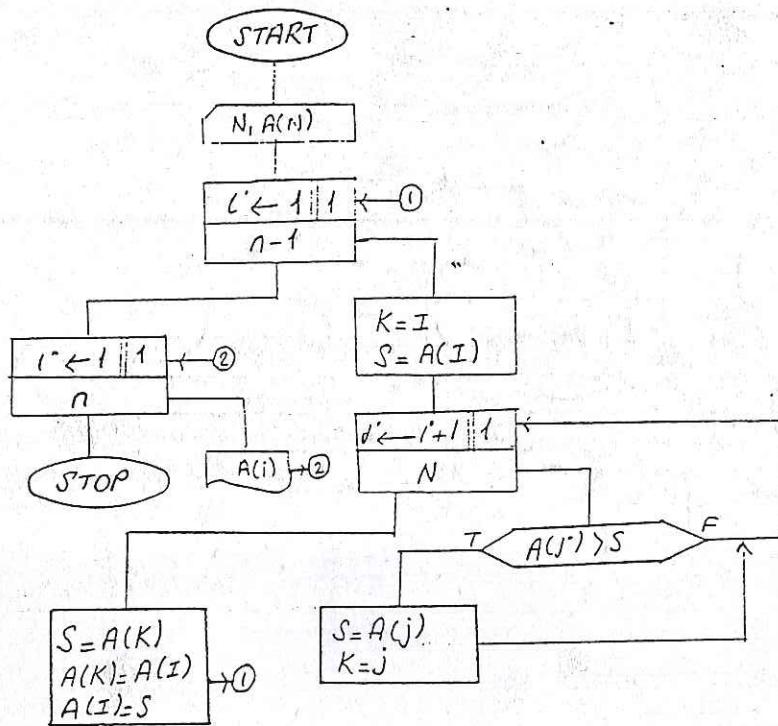
Program yenitaban (input, output);
 uses crt;
 var
 n, b, c, t, i : integer;
 a:array [1..10] of integer;
 begin
 clrscr;
 writeln ('Dizi boyunu gir');
 readln (n);
 write ('Dizi elementlerini gir');
 for i:= 1 to n do
 read (A[i]);
 writeln ('Tabani gir');
 readln (B);
 c:= 1; t:= 0;
 for i:= N downto 1 do
 begin
 c:= c * B;
 t:= t + (A[i]*c);
 end;
 writeln (t);
 end.

Onluk düzendeki bir sayıyı herhangi bir düzene çevirem algoritma ve programı.



Program yenidüzen (input, output);
 uses crt;
 var
 n, b, i, j, m: integer;
 a:array [1..10] of integer;
 begin
 clrscr;
 writeln ('Sayiyi gir');
 readln (n);
 writeln ('Tabani gir');
 readln (b);
 i:= 1;
 while N > b do
 begin
 m := N Mod b;
 N := N div b;
 A[i]:= m;
 i := i + 1;
 end;
 A[i]:= n;
 for j:= i downto 1 do
 write (A[j]:3);
 end.

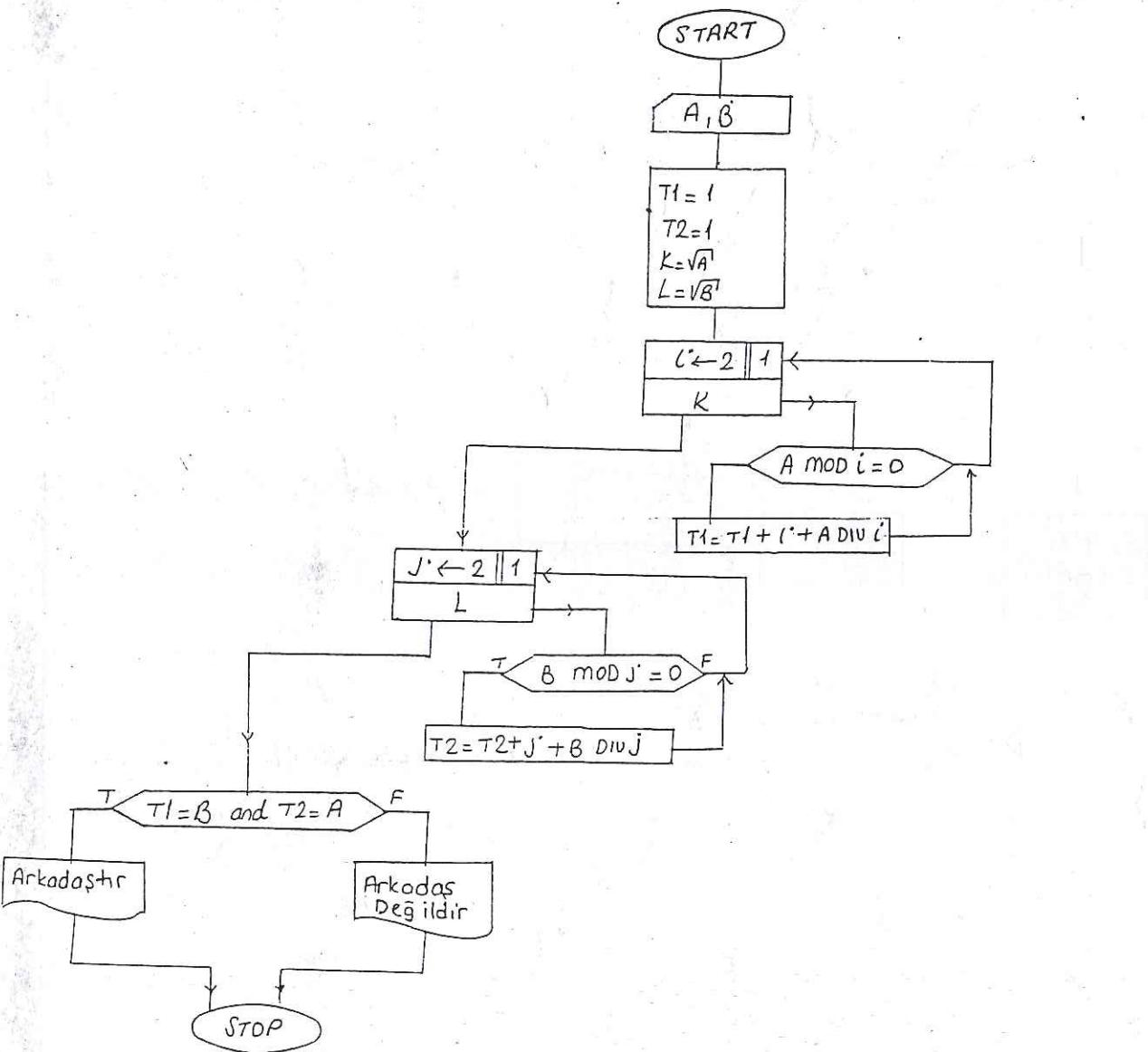
Dizi elementlerinin büyükten küçüğe sıralanmasını sağlayan algoritma ve programı.
(SELECTION SORT)



```

Program Selection (input, output);
uses crt;
Var
  i, n, k, s, j : integer;
  A: array[1..10] of integer;
  Writeln ('Dizi boyutunu giriniz');
  Readln (n);
  Writeln ('Dizi elementlerini giriniz');
  For i:= 1 to n do
    Read (A[i]);
  For i:= 1 to n-1 do
    Begin
      k := i;
      s := A[i];
      for j := i+1 to n do
        Begin
          if A[j] > s then
            Begin
              s := A[j];
              k := j;
            End;
        End;
      A[k] := A[i];
      A[i] := s;
    End;
    For i := 1 to n do
      Write (A[i]:3);
    End.
  
```

Güçlerinin toplamı karşılıklı kendilerine eşit olan iki sayıya "arkadaş sayı" denir.
Buna göre girilen iki sayının arkadaş olup olmadığını bulan algoritma ve programı.



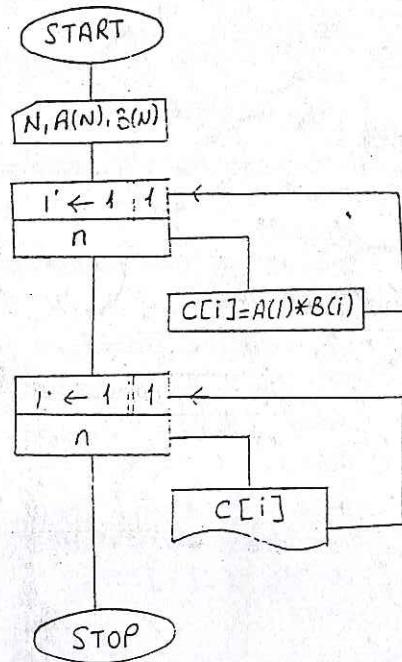
```

Program arkadas (input, output);
uses crt;
var
  a, b, t1, t2, i, j, k, l: integer;
begin
  clrscr;
  writeln ('iki sayı giriniz');
  readln (a, b);
  t1 := 1; t2 := 1;
  k := trunc(sqrt(a));
  l := trunc(sqrt(b));
  for i := 2 to k do
    begin
      if a mod i = 0 then
        t1 := t1 + i + (a div i);
    end;
  for j := 2 to l do
    begin
      if b mod j = 0 then
        t2 := t2 + i + (b div i);
    end;
end.
    
```

```

if ( $t_1 = b$ ) and ( $t_2 = a$ ) then
begin
  writeln ('Sayilar arkadastir');
  readln;
end
else
  writeln ('Sayilar arkadas degildir');
  readln;
end.
    
```

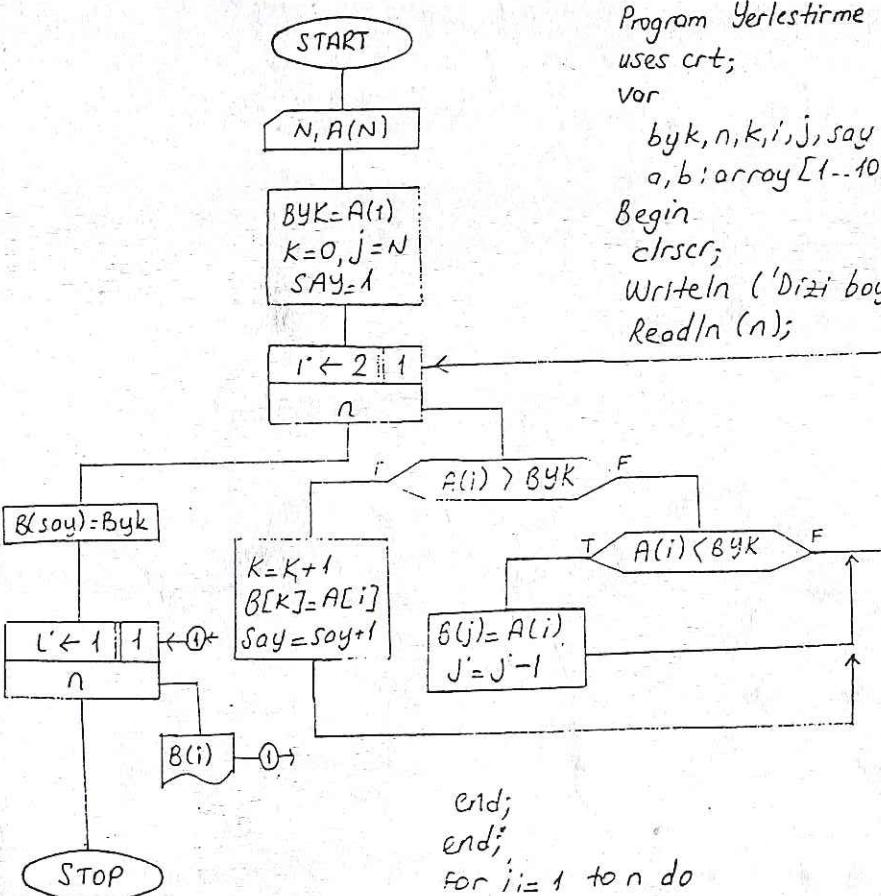
0 ve 1'lerden oluşan iki diziden bir c dizisinin oluşması, ikisi de 1 ise 1, biri sıfır diğer 1 ise 0 olacak şekilde. Bu işlemi yapan algoritma ve programı.



```

Program. dizin (input, output);
uses crt;
Var
  i,n:integer;
  a,b,c:array [1..10] of integer;
Begin
  clrscr;
  writeln ('Dizi Boyutu');
  Readln (n);
  writeln ('A dizisini gir');
  For i:= 1 to n do
    Read (A[i]);
  writeln ('B dizisini gir');
  For i:= 1 to n do
    Read (B[i]);
  For i:= 1 to n do
    C[i] := A[i] * B[i];
  for i:= 1 to n do
    write (C[i]:3);
  End.
  
```

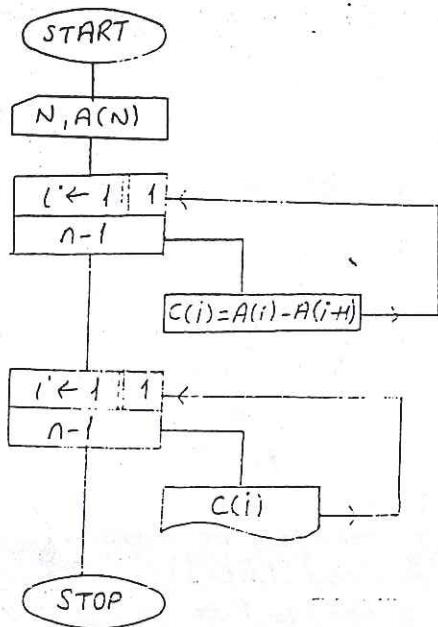
Bir dizinin ilk gözündeki elemanı büyük kabul ederek, ondan büyük olanları yeni bir dizinin başından, küçük olanları sonundan itibaren yerlestiren algoritma ve programı.



```

Program Yerlestirme (input, output);
uses crt;
Var
  byk,n,k,i,j,say:integer;
  a,b:array [1..10] of integer;
Begin
  clrscr;
  writeln ('Dizi boyutunu girin');
  Readln (n);
  writeln ('Elemanları gir');
  For i:= 1 to n do
    Read (A[i]);
  BYK := A[1];
  K := 0; j := N; say := 1;
  For i:= 2 to n do
    Begin
      If A[i] > BYK then
        begin
          k := k + 1;
          B[k] := A[i];
          say := say + 1;
        end
      else
        begin
          If A[i] < BYK then
            begin
              B[j] := A[i];
              j := j - 1;
            end
        end
    End;
  End.
  
```

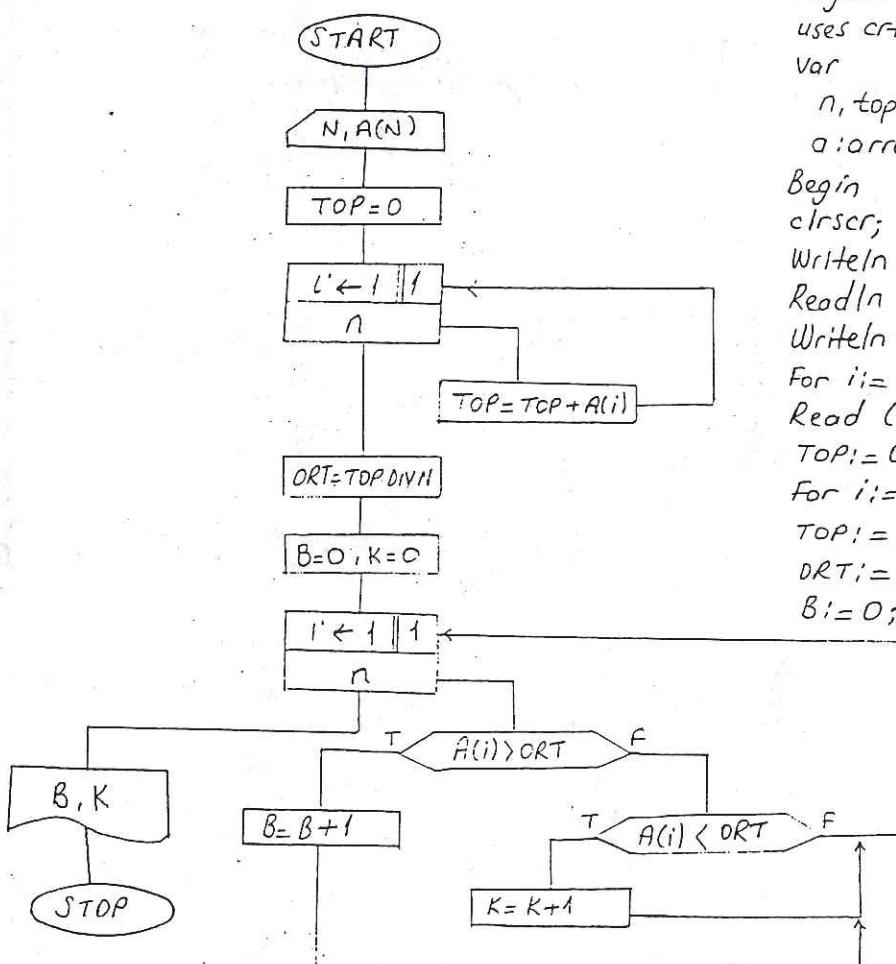
Bir dizinin elementleri arasındaki farkı baska bir dizİYE yazdırın algoritma ve programı.



```

Program fark (input, output);
uses crt;
var
  n, i: integer;
  a, c: array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu giriniz');
  readln (n);
  writeln ('Elemanları giriniz');
  for i := 1 to n do
    read (a[i]);
  for i := 1 to n-1 do
    c[i] := a[i] - a[i+1];
  for i := 1 to n-1 do
    write (c[i]:3);
end.
  
```

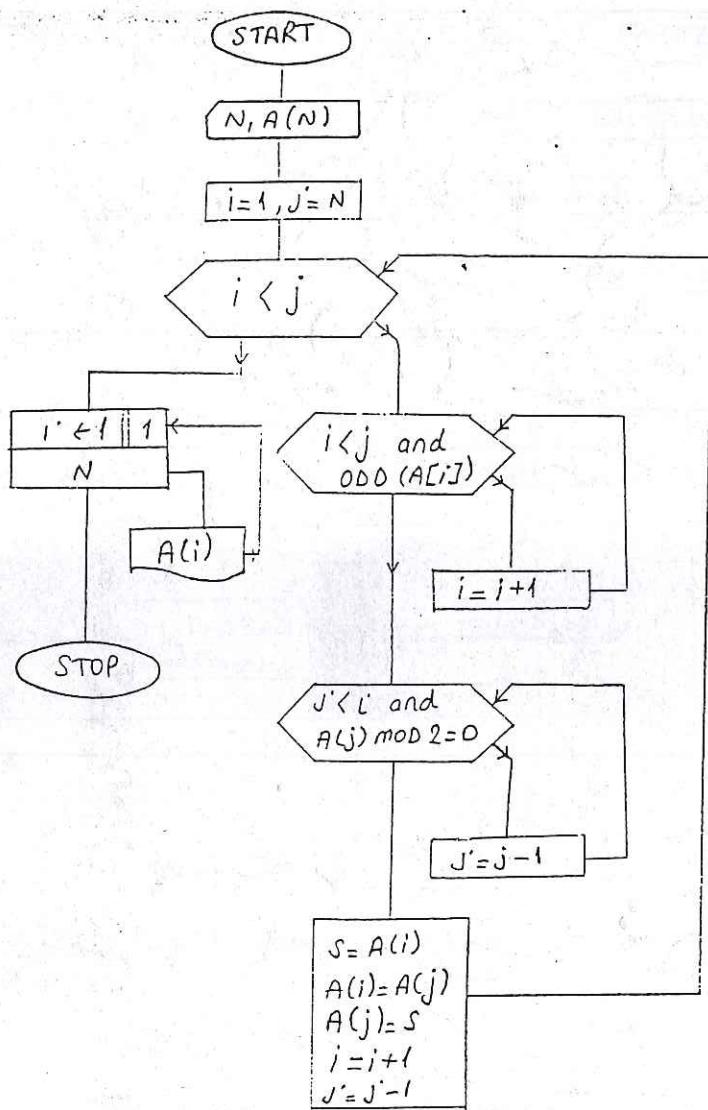
Bir dizinin elemanlarının ortalaması değerini; ortalamadan büyük ve küçük elemen sayısını bulan algoritma ve programı.



```

Program Ortalama (input, output);
uses crt;
var
  n, top, i, ort, b, k: integer;
  a: array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutu');
  readln (n);
  writeln ('Dizi elemanları');
  for i := 1 to n do
    read (a[i]);
  top := 0;
  for i := 1 to n do
    top := top + a[i];
  ort := top div n;
  b := 0;
  k := 0;
  for i := 1 to n do
    begin
      if a[i] > ort then
        b := b + 1
      else
        begin
          if a[i] < ort then
            k := k + 1;
        end;
    end;
  write (b:3, k:3);
end.
  
```

Aynı dizide tekler başa, çiftler sona gelecek şekilde dizili elemanlarını yerlestiren algoritma ve programı.



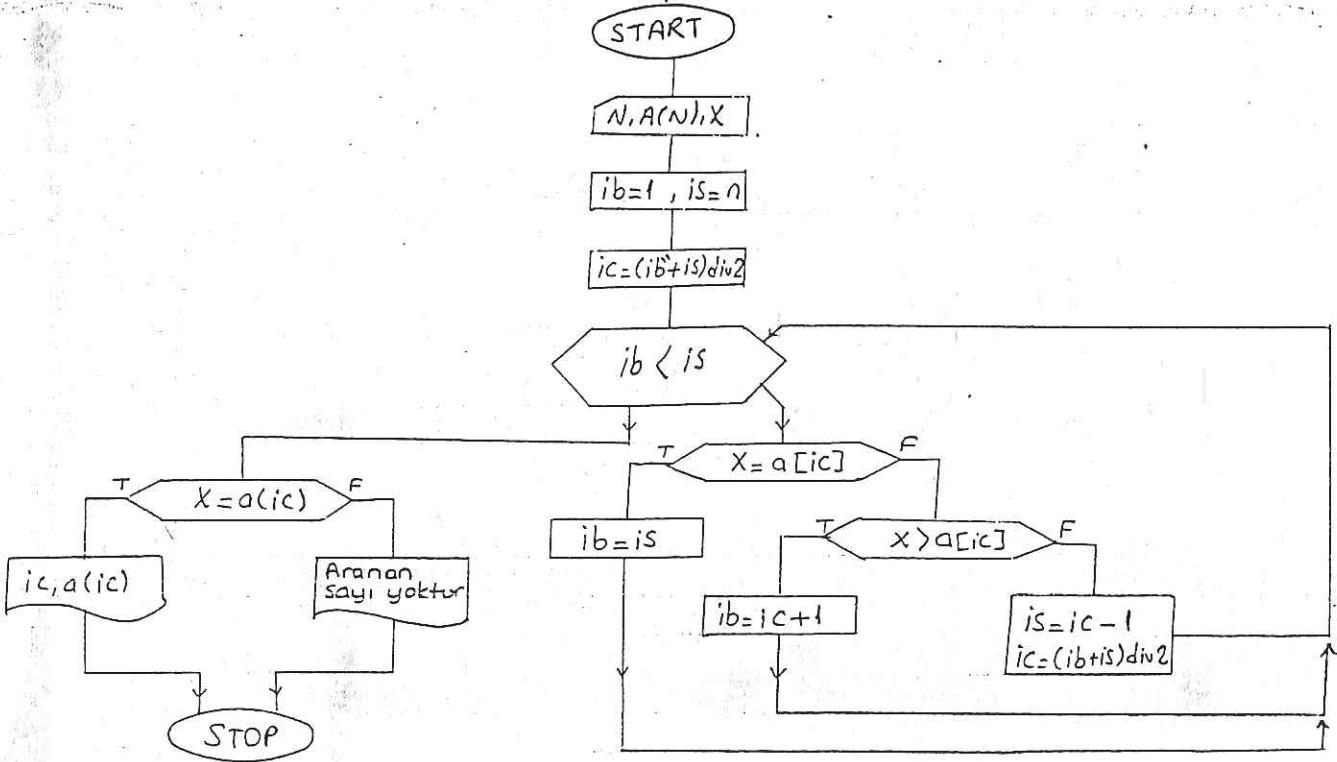
```

Program tekciift (input, output);
uses crt;
var
  i,j,s,n:integer;
  a:array [1..10] of integer;
begin
  clrscr;
  writeln ('A dizisinin eleman sayisini giriniz');
  Readln (n);
  writeln ('A dizisinin elemanlarini giriniz');
  for i:=1 to n do
    Readln (a[i]);
  i:=1; j:=n;
  while i < j do
    begin
      while (i < j) and (odd(a[i])) do
        i:=i+1;
      while (j > i) and (a[j] mod 2 = 0) do
        j:=j-1;
      s:=a[i];
      a[i]:=a[j];
      a[j]:=s;
    end;
end.
  
```

```

i := i + 1;
j := j - 1;
end;
For i : = 1 to n do
Write (A[i]:5);
Readln ;
End.
  
```

Sıralı bir dizide herhangi bir X sayısının olup olmadığını daha kısa yoldan arastırma algoritma ve programı. (Binary Search).



Program Binary (input, output);

uses crt;

var

n, x, i, ib, is, ic : integer;
 a : array [1..10] of integer;

Begin

clrscr;

Writeln ('A dizisinin boyutunu giriniz');

Readln (n);

Writeln ('A dizisini giriniz');

For $i := 1$ to n do

Readln ($A[i]$);

Writeln ('Ardığınız sayıyı giriniz');

Readln (x);

$ib := 1$; $is := n$; $ic := (ib+is) \text{ div } 2$;

while $ib < is$ do

Begin

if $x = a[ic]$ then

$ib := is$

else

Begin

if $x > a[ic]$ then

$ib := ic + 1$

else begin

$is := ic - 1$;

$ic := (ib+is) \text{ div } 2$;

end;

end;

end.

if $x = a[ic]$ then

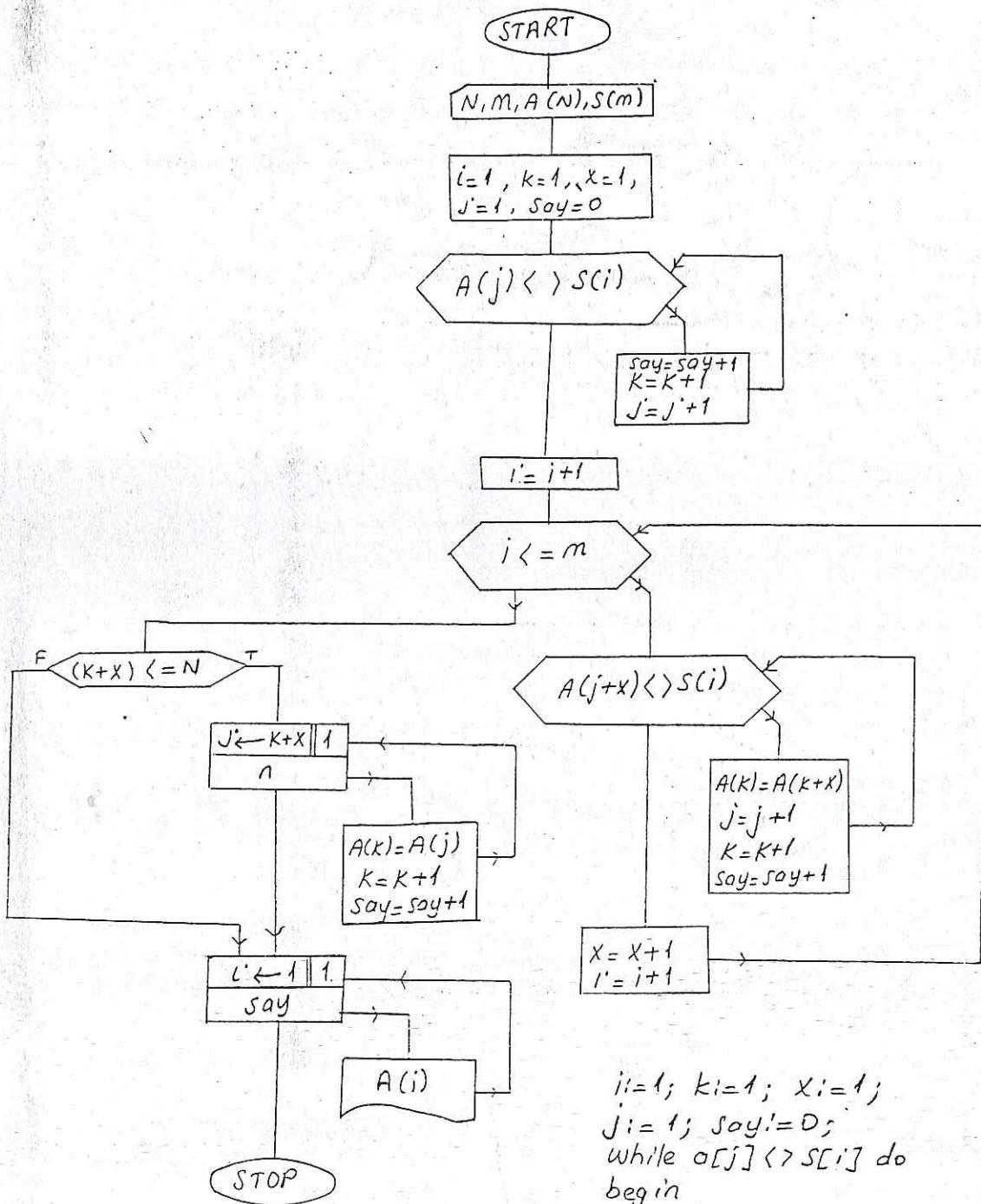
Write (ic:3, a[ic]:3)

Else

Writeln ('aranan sayı yok');

End.

Verilen bir dizide adresi belli olanların silinip, kalonların sıkıştırılmasını sağlayan algoritma ve programı. (Kayıt Silme).



```

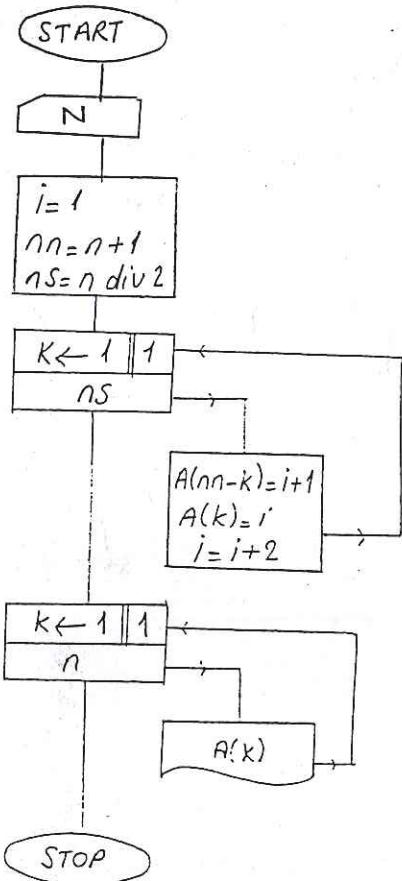
Program kayitsilmebir(input, output);
uses crt;
var
  n,m,i,j,x,k,say : integer;
  s,a : array [1..10] of integer;
begin
  clrscr;
  writeln('A ve S dizilerinin boyutunu gir');
  readln(n,m);
  writeln('A dizisinin elementlerini gir');
  for i:=1 to n do
    read(a[i]);
  writeln('S dizisinin elementlerini gir');
  for i:=1 to m do
    
```

```

    i:=1; k:=1; x:=1;
    j:=1; say:=0;
    while a[j] < > s[i] do
      begin
        k:=k+1;
        j:=j+1;
        say:=say+1;
      end;
      i:=i+1;
    while i <= m do
      begin
        while a[j+x] < > s[i] do
          begin
            a[k]:=a[j];
            k:=k+1;
            j:=j+1;
            say:=say+1;
          end;
        write(a[i]);
      end;
    end;
  
```

(23)

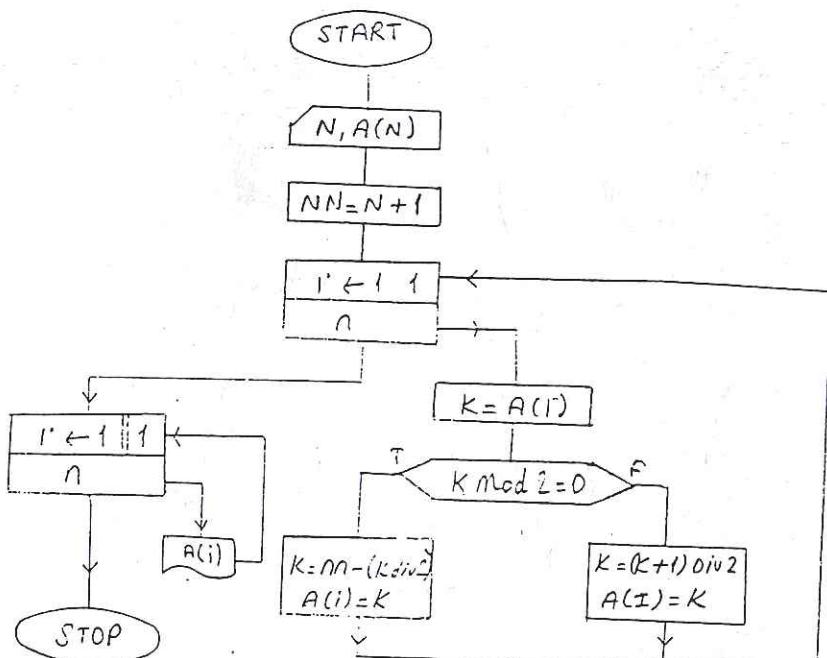
Herhangi bir N sayısına göre $1\ 3\ 5\dots N\dots 4\ 2$ dizisini oluşturan algoritma ve programı.



```

Program Dizgi (input,output);
uses crt;
var
  n,i,nn,ns,k:integer;
  a:array [1..10] of integer;
begin
  clrscr;
  writeln ('Sayiyi gir');
  Readln (n);
  i:= 1;
  nn:=n+1;
  ns:= n div 2;
  for k:=1 to ns do
    begin
      a[nn-k]:= i+1;
      a[k]:= i;
      i:= i+2;
    end;
  for k:= 1 to n do
    write (a[k]:4);
  end.
  
```

Bir N sayısına göre $1\ 3\ 5\dots N\dots 4\ 2$ sekline getirilmiş diziyi $1\ 2\dots N$ (eski haline) haline dönüştüren algoritma ve programı.

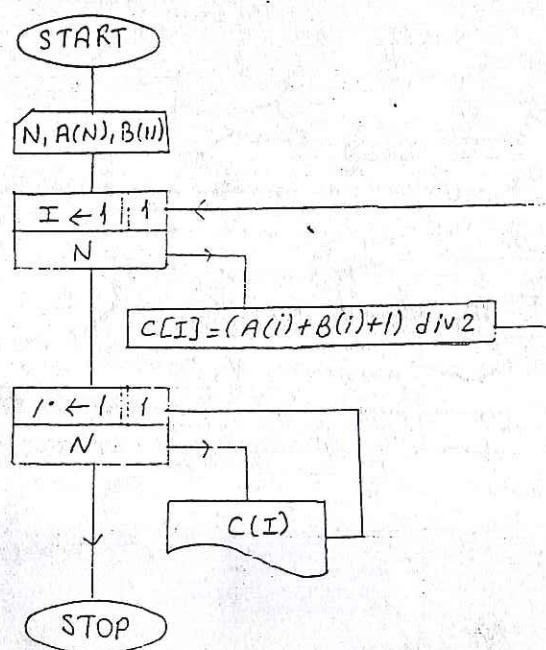


```

Program tersdizgi (input,output);
uses crt;
var
  n,i,nn,k :integer;
  a:array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu girin');
  Readln (n);
  writeln ('Dizi elementlerini giriniz');
  for i:= 1 to n do
    Read (a[i]);
  nn:= n+1;
  for i:= 1 to n do
    begin
      k:= a[i];
      if k mod 2 = 0 then
        begin
          k:= nn - (k div 2);
          a[i]:= k;
        end
      else
        begin
          k:= (k+1) div 2;
        end;
      end;
    
```

end;
for i:= 1 to n do
Write (a[i]:5);
End.

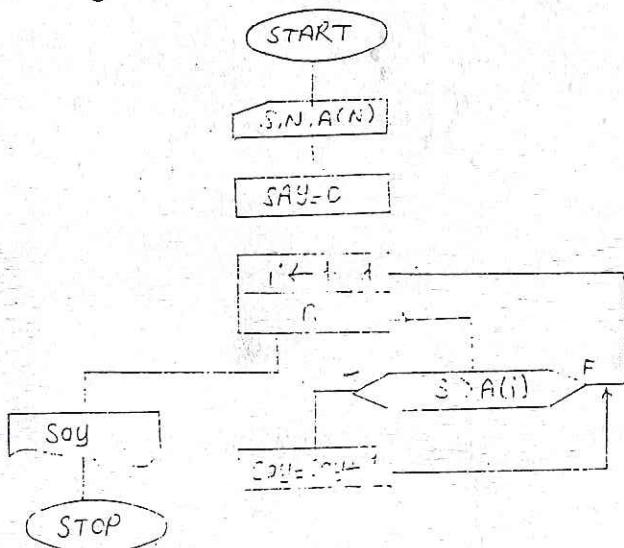
O ve 1 'lerden oluşan diziler veriliyor. C dizisi, A veya B 'den en az birinin 1 olduğu yerde bir oluyor. C dizisini veren algoritma ve program.



```

Program dizi1(input,output);
uses crt;
var
  n,i : integer;
  a,b,c : array [1..10] of integer;
begin
  clrscr;
  writeln('Dizinin boyutu');
  readln(n);
  write('a dizisini gir');
  for i:=1 to n do
    read(a[i]);
  write('b dizisini gir');
  for i:=1 to n do
    read(b[i]);
  for i:=1 to n do
    c[i]:= (a[i]+b[i]+1) div 2;
  for i:=1 to n do
    write(c[i]:3);
  end.
    
```

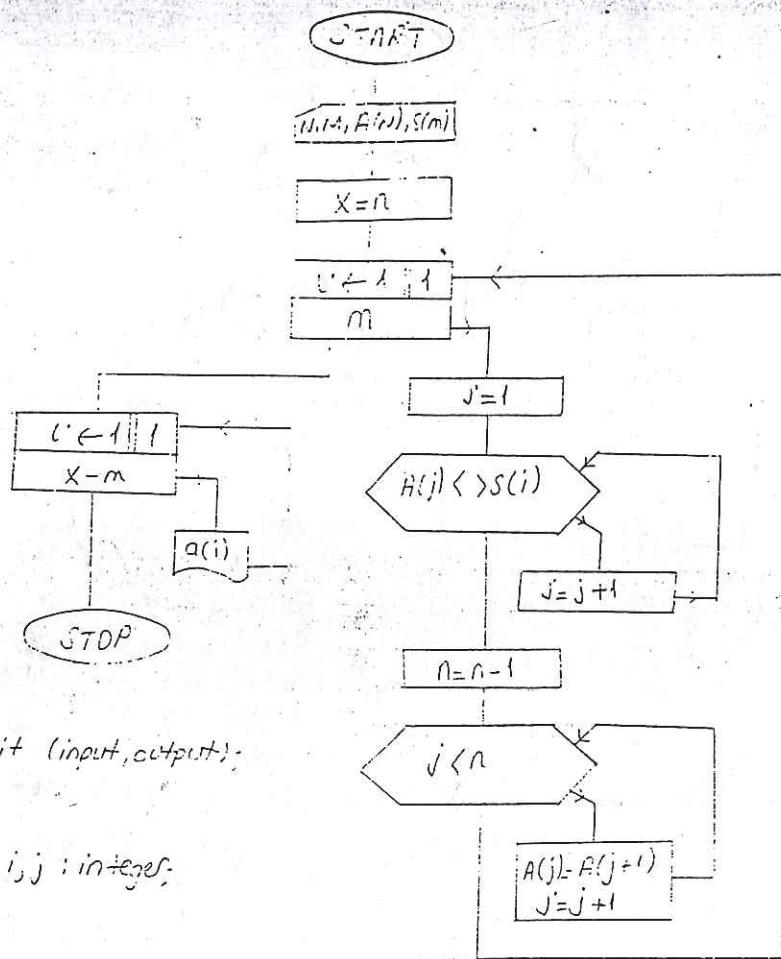
Bir dizide dışardan girilen bir sayının dizinin elemanlarından kaç tanesinden büyük olduğunu bulan algoritma ve programı.



```

Program buyukler(input,output);
uses crt;
var
  n,s,say,i : integer;
  a:array [1..10] of integer;
begin
  clrscr;
  writeln('Dizi boyutunu ve sayiyi gir');
  readln(n,s);
  writeln('Dizi elemanlarini gir');
  for i:=1 to n do
    read(a[i]);
  say:=0;
  for i:=1 to n do
    if s>a[i] then
      say:=say+1;
  writeln(say);
  end.
    
```

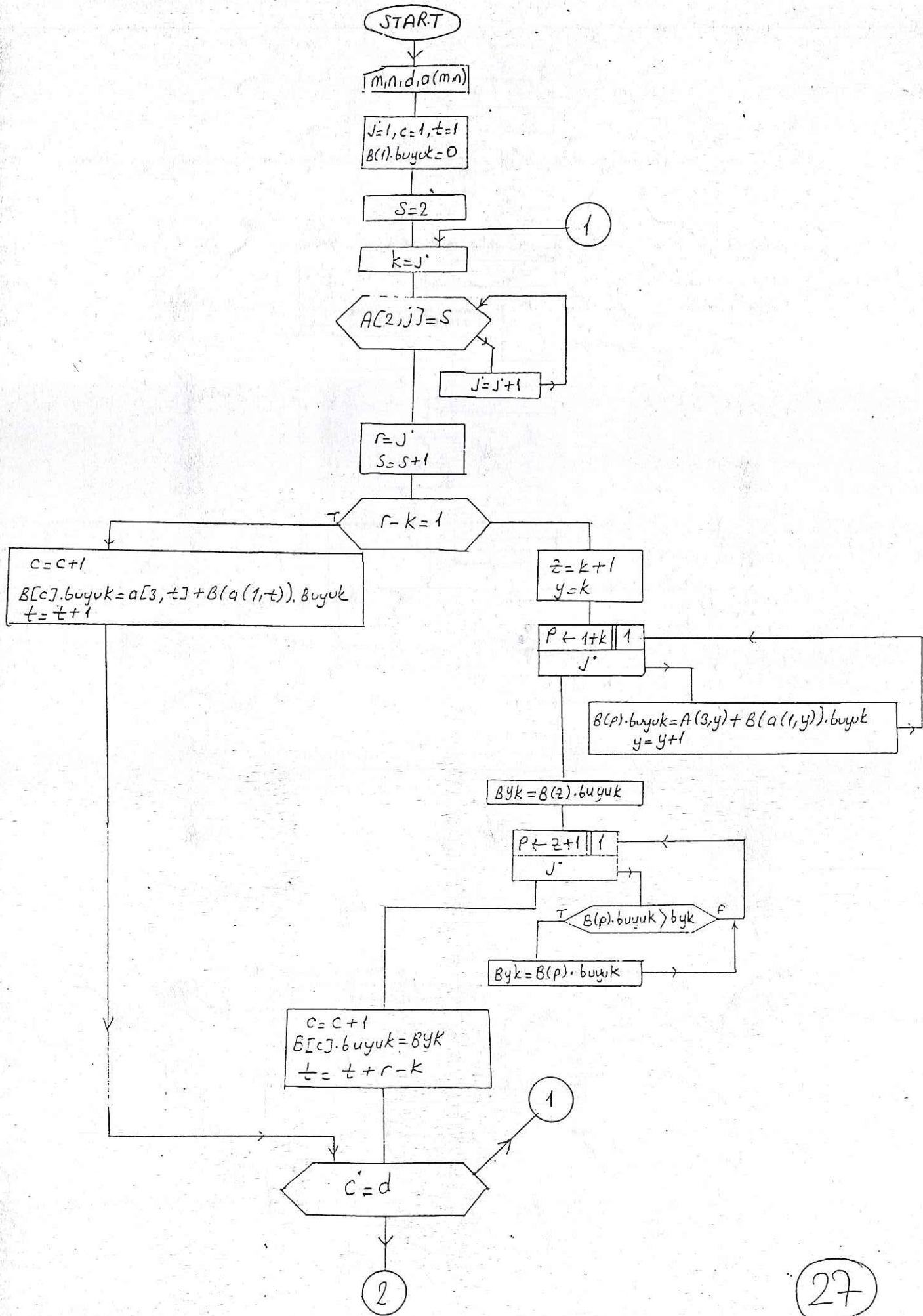
Kayıt silme algoritması ve programı.

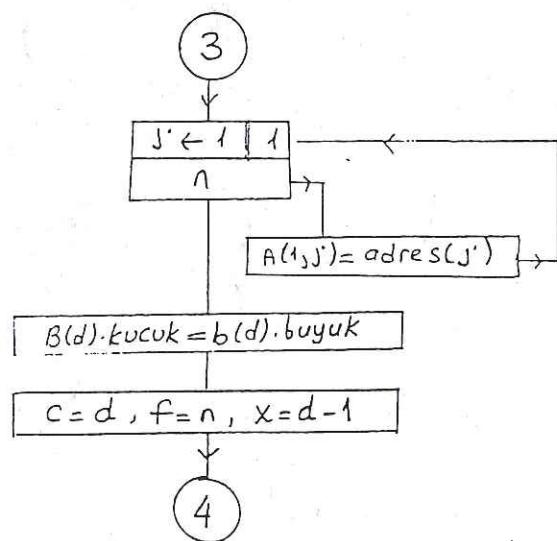
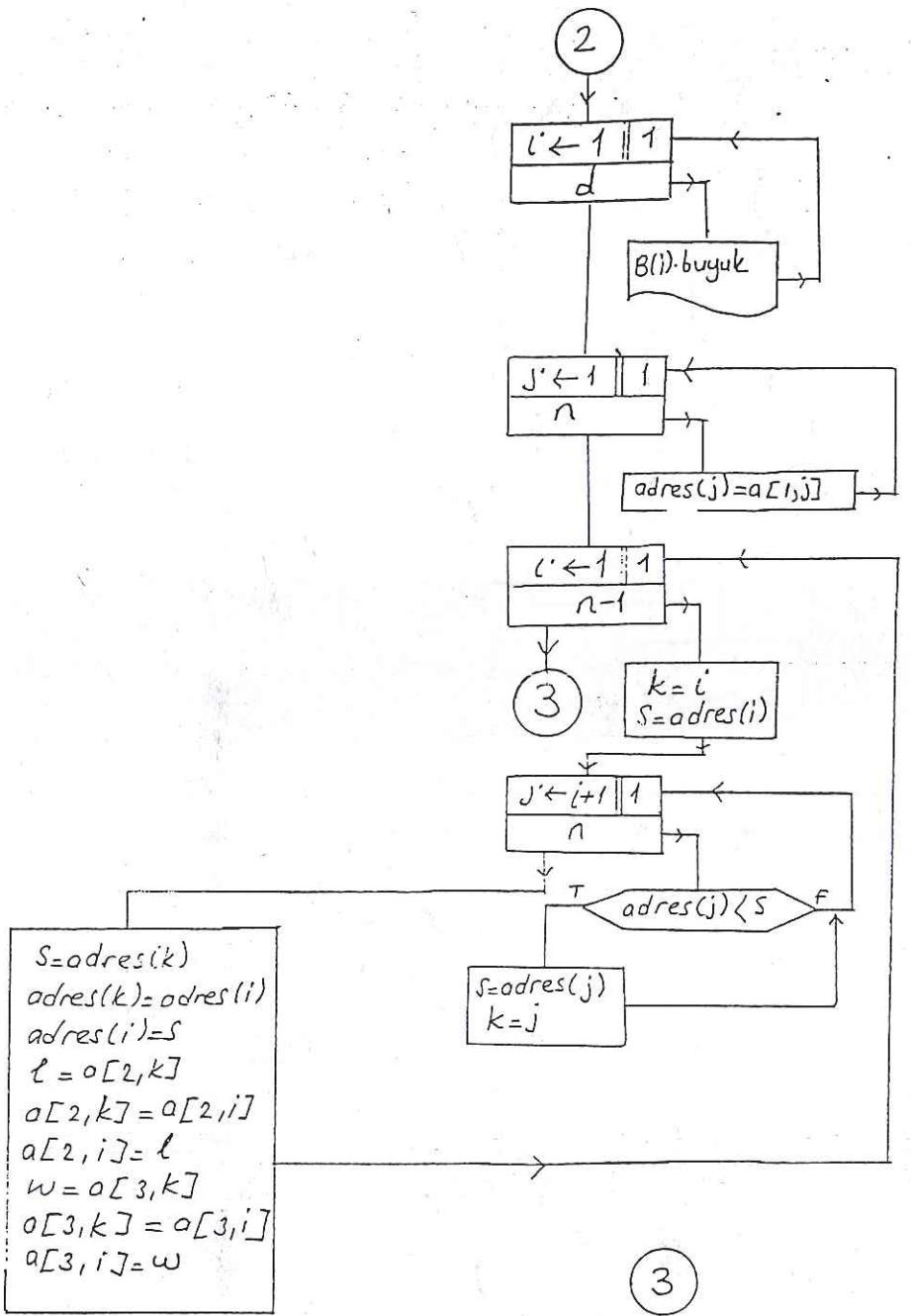


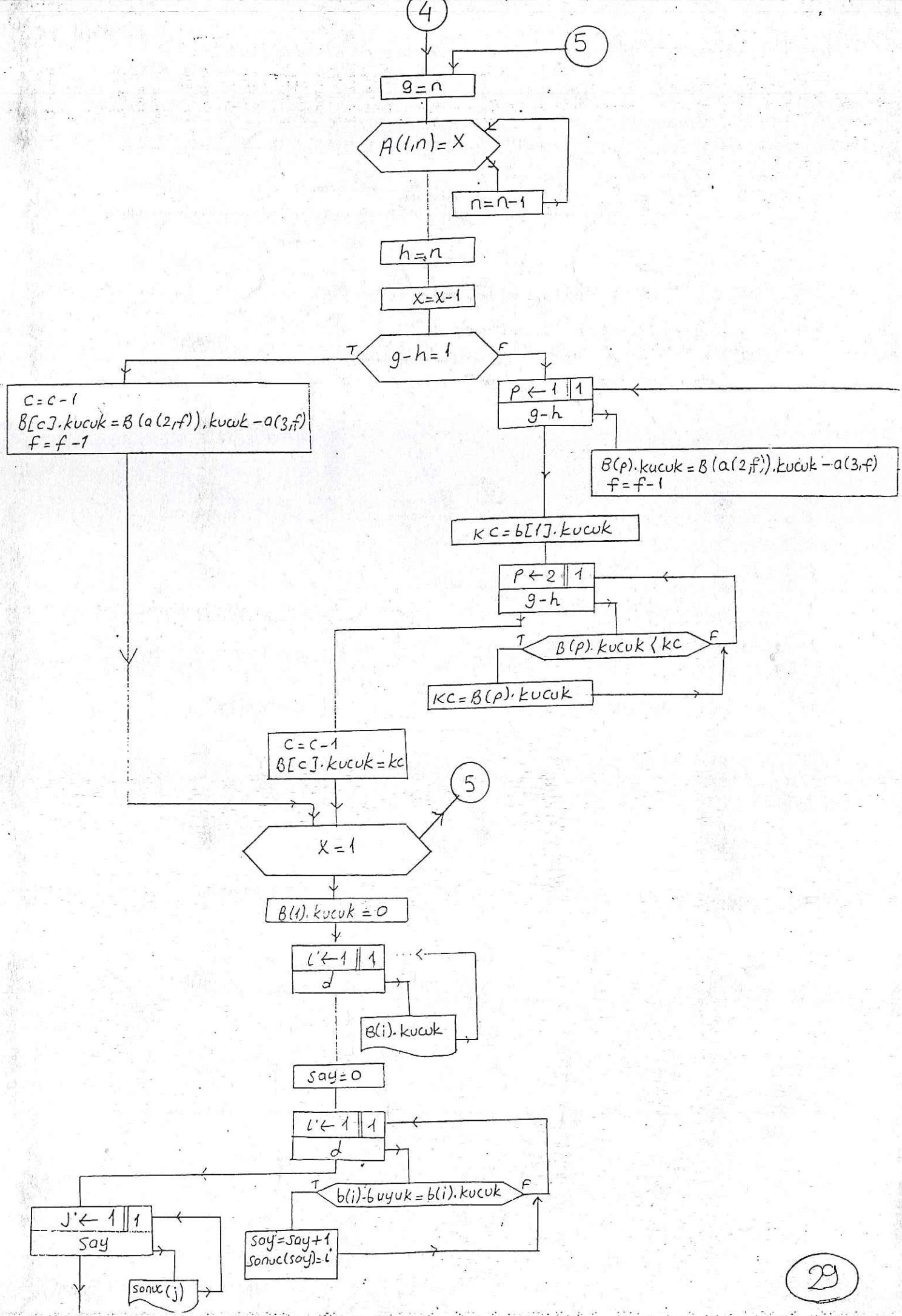
```

Program Kayit (input, output);
uses crt;
var
  x,n,m,say,i,j : integer;
begin
  clrscr;
  writeln ('A ve s dizisinin boyutu');
  Readln (n,m);
  writeln ('a dizisinin elemanları');
  for i:=1 to n do
    Read (a[i]);
  writeln ('s dizisini giriniz');
  for i:=1 to m do
    Read (s[i]);
  x:=n;
  for i:=1 to m do
    begin
      j:=1;
      while a[j] > s[i] do
        j:=j+1;
      n:=n-1;
      while j < n do
        begin
          a[j]:=a[j+1];
          j:=j+1;
        end;
      end;
    end;
  for i:=1 to x-m do
    write (a[i]:5);
  end.
  
```

Kritik Yol Metodunun algoritması ve programı.







```

until
c=d;
for i:= 1 to d do
  write (b[i].buyuk :3);
  for j:= 1 to n do
    adres[j]:= a[i,j];
    for i:= 1 to n-1 do
      Begin
        k:= i;
        s:= adres[i];
        for j:= i+1 to n do
          Begin
            if adres[j] < s then
              Begin
                s:= adres[j];
                k:= j;
              End;
            s:= adres[k];
            adres[k]:= adres[i];
            adres[i]:= s;
            l:= a[2,k];
            o[2,k]:= a[2,i];
            a[2,i]:= l;
            w:= a[3,k];
            a[3,k]:= o[3,i];
            a[3,i]:= w;
          End;
        End;
      End;
    for j:= 1 to n do
      a[1,j]:= adres[j];
      b[d].kucuk:= b[d].buyuk;
      c:= d; f:= n; x:= d-1;
      Repeat
        g:= n;
        while a[1,n]=x do
          n:= n-1;
        h:= n; x:= x-1;
        if g-h=1 then
          Begin
            c:= c-1;
            b[c].kucuk:= b[a[2,f]].kucuk - a[3,f];
            f:= f-1;
          end
        else
          begin
            for p:= 1 to g-h do
              Begin
                b[p].kucuk:= b[a[2,f]].kucuk - a[3,f];
                f:= f-1;
              End;
            kc:= b[1].kucuk;
            for p:= 2 to g-h do
              Begin
                if b[p].kucuk / kc then

```

```

      kc:= b[p].kucuk;
    End;
    c:= c-1;
    b[c].kucuk:= kc;
  End;
  until
  x:= 1;
  b[1].kucuk:= 0;
  for i:= 1 to d do
    write (b[i].kucuk :3);
  say:= 0;
  for i:= 1 to d do
  Begin
    if b[i].buyuk = b[i].kucuk then
      Begin
        say:= say+1;
        sonuc[say]:= i;
      End;
    End;
    for j:= 1 to say do
      write (sonuc[j]:3);
  End.

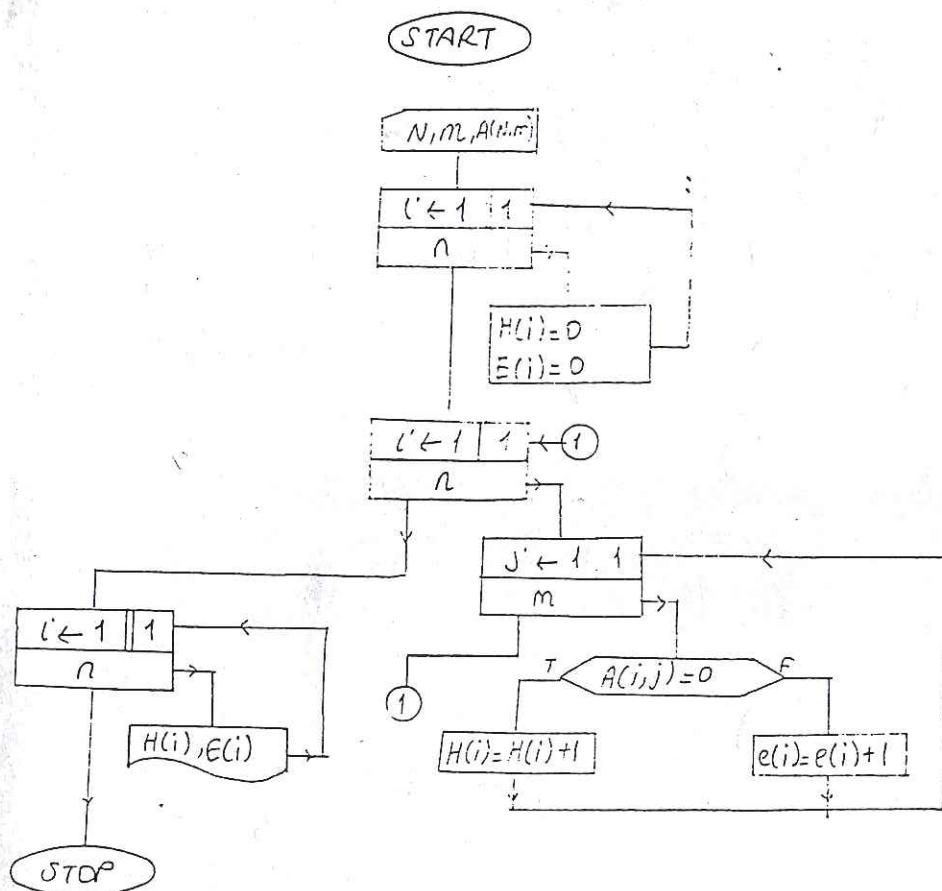
```

```

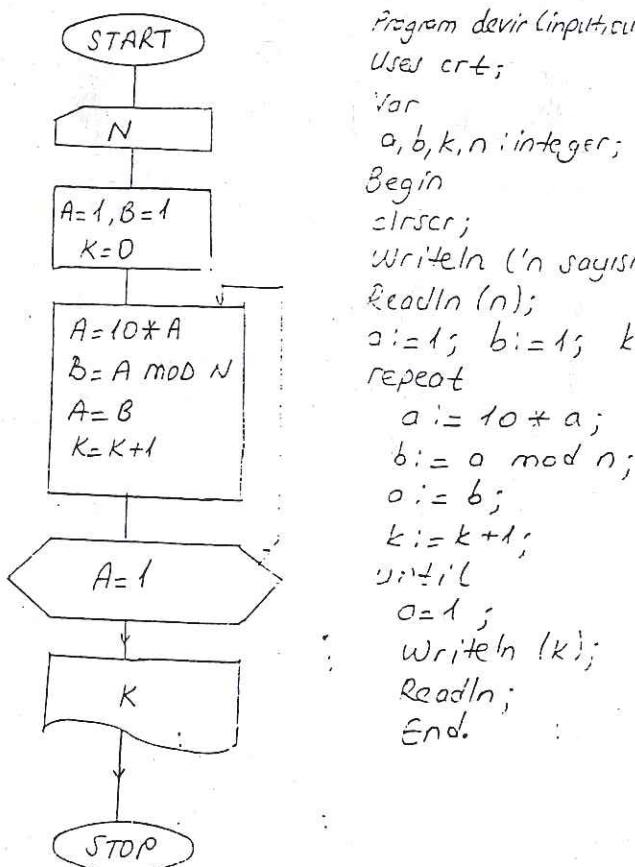
Program Meral (input, output);
uses crt;
type
  sr=record
    buyuk:integer;
    kucuk:integer;
  end;
var
  w,l,x,h,g,f,say,m,n,kc,j,c,t,byk,p,k,s,z,r,y,i,d:integer;
  a:array [1..3, 1..15] of integer;
  b:array [1..15] of sr;
  odres:array [1..20] of integer;
  Sonuc:array [1..10] of integer;
Begin
  clrscr;
  Writeln ('matrisin boyutunu giriniz');
  Readln (m,n);
  Writeln ('matrisin elemanlorsini giriniz');
  For i:=1 to m do
  Begin
    for j := 1 to n do
      Read (a[i,j]); Readln; End;
    Writeln ('Dugum sayisini giriniz');
    Readln (d);
    j:=1; c:=1; t:=1;
    b[1].buyuk:=0; s:=2;
    Repeat
      k:=j;
      While a[2,j]=s do
        j:=j+1; r:=j; s:=s+1;
      If r-k=1 then
      Begin
        c:=c+1;
        b[c].buyuk:=a[3,t]+b[a[1,t]].buyuk;
        t:=t+1;
      End
      Else
      Begin
        z:=k+1; y:=k;
        for p:=1+k to j do
        Begin
          b[p].buyuk:=a[3,y]+b[a[1,y]].buyuk;
          y:=y+1;
        End;
        BYK:=b[z].buyuk;
        for p:=1+z to j do
        Begin
          if b[p].buyuk > byk then
            byk:=b[p].buyuk;
        End;
        c:=c+1;
        b[c].buyuk:=byk;
        t:=t+r-k;
      End;
    End;
  End;

```

200 kişiye 30 soruluk bir anket uygulanıyor. Cevaplar evet ve hayır'dan oluşuyor. Evetler 1, hayır'lar 0 ile gösteriliyor. Buna göre her soruya verilen evet ve hayır sayılarını bir diziye yazdırın algoritma ve programı.

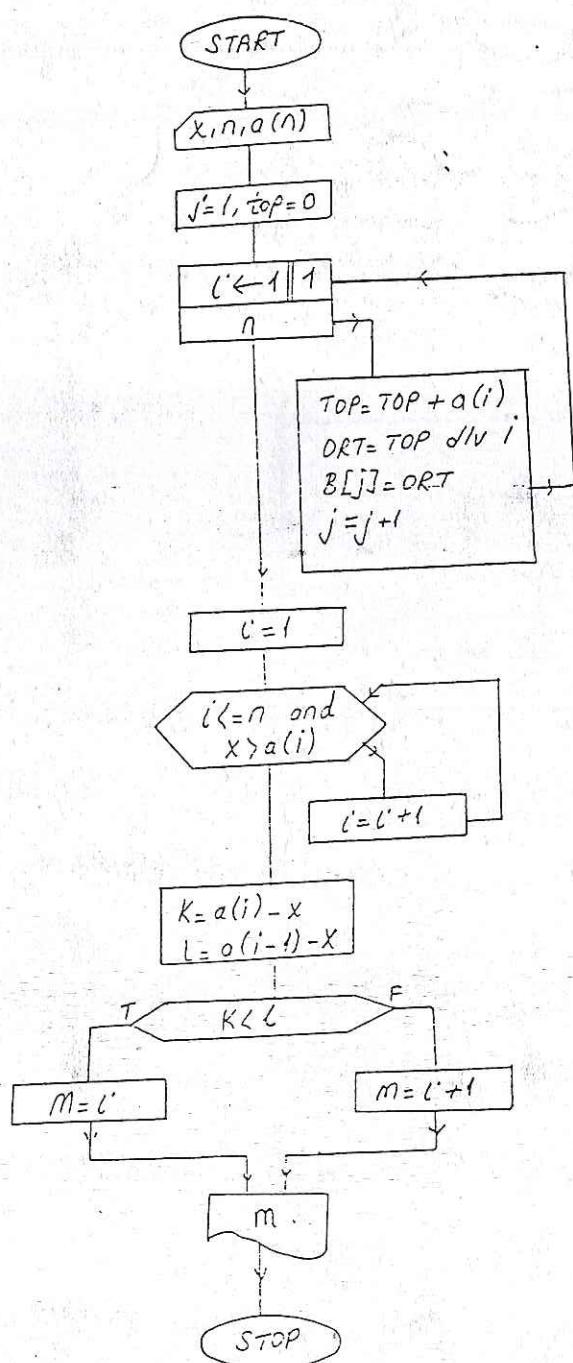


$\frac{1}{n}$ 'in kaç basamak sonra devrettigini bulan algoritma ve programı.



Program devir (input, output);
uses crt;
Var
a, b, k, n : integer;
Begin
clrscr;
writeln ('n sayisini giriniz');
readln (n);
a:=1; b:=1; k:=0;
repeat
 a := 10 * a;
 b := a mod n;
 a := b;
 k := k + 1;
until a=1;
writeln (k);
readln;
End.

Küçükten büyüğe sıralı bir $A(N)$ dizisinin elemanları toplanarak ortalamaya değerleri alınıyor. Ortalama belli bir X değerine yakın olana dek devam ediliyor. Kas elemanın ortalamaya girdiğini bulan algoritma, ve programı.



```

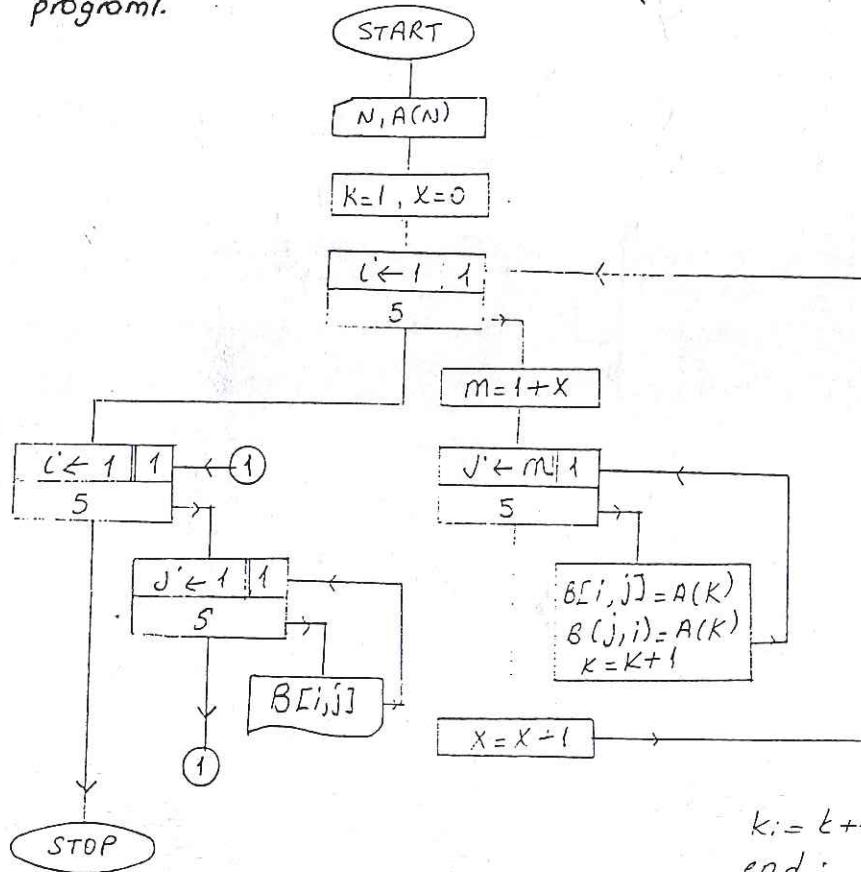
Program Sabri (input, output);
Uses crt;
Var
  n, i, j, top, ort, x, k, l, m: integer;
  a, b: array [1..10] of integer;
Begin
  clrscr;
  writeln ('Dizi boyutunu giriniz');
  Readln (n);
  writeln ('Dizi elementlerini giriniz');
  For i := 1 to n do
    Read (a[i]);
  writeln ('x sayisini gir');
  Readln (x);
  j := 1; top := 0;
  for i := 1 to n do
  Begin
    top := top + a[i];
    ort := top div i;
    b[j] := ort;
    j := j + 1;
  end;
  i := 1;
  while (i <= n) and (x > a[i]) do
  begin
    i := i + 1;
    k := a[i] - x;
    l := a[i-1] - x;
    if k < l then
      m := i
    else
      m := i + 1;
    writeln (m);
    Readln;
  End.
  
```

```

yeni[sat,3]:=a[i,j];
end;
end;
for i:=1 to sat do
begin
for j:=1 to 3 do
write(yeni[i,j]);
writeln;
End;
End.

```

15 elemanlı bir dizi ile 5×5 'lik simetrik bir matris oluşturulan algoritma ve programı.



```

Program simetrik (input, output);
uses crt;
var
  m, k, x, n, i, j: integer;
  a: array [1..15] of integer;
  b: array [1..5, 1..5] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu gir');
  Readln (n);
  writeln ('Dizi elemanlarını gir');
  for i:=1 to n do
    Read(a[i]);
  k:=1; x:=0;
  for i:=1 to 5 do
  begin
    m:=1+x;
    for j:=1 to 5 do
    begin
      b[i,j]:=a[k];
      b[j,i]:=a[k];
      k:=k+1;
    end;
  end;
end.

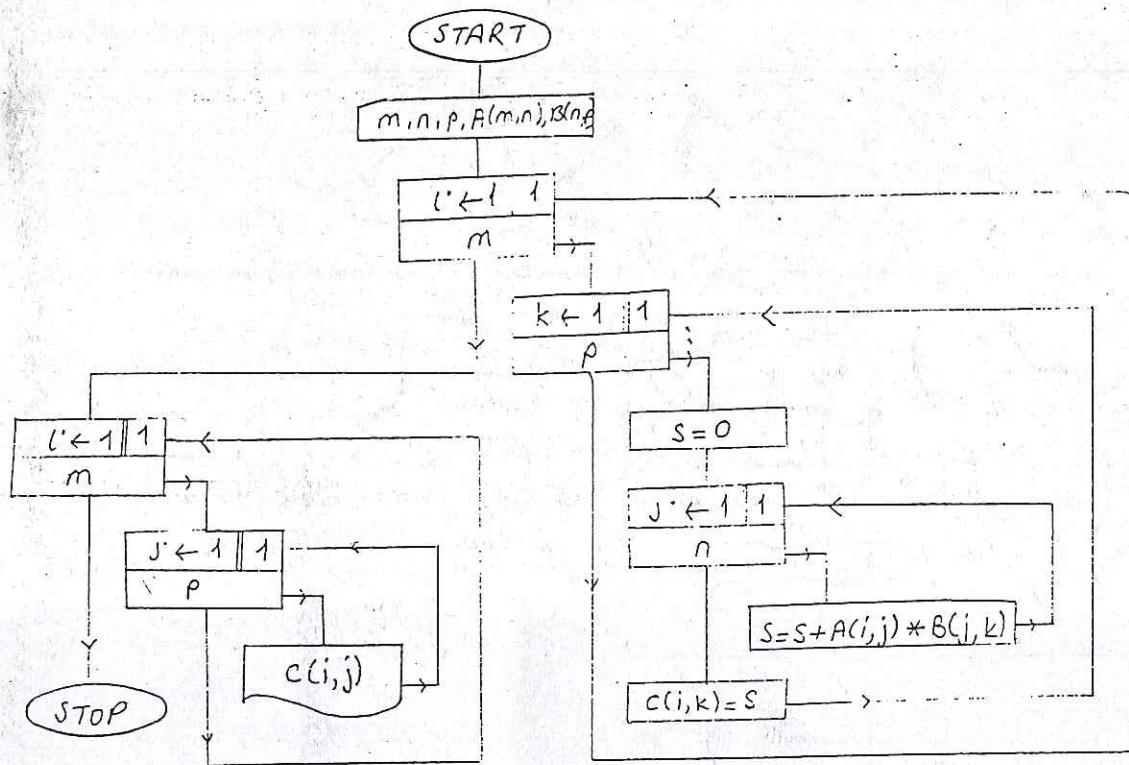
```

```

k:= k+1;
end;
x:= x+1;
end;
for i:=1 to 5 do
begin
  for j:= 1 to 5 do
  write (b[i,j]:3);
  writeln;
End;
End.

```

iki matrisin çarpımı algoritması ve programı.



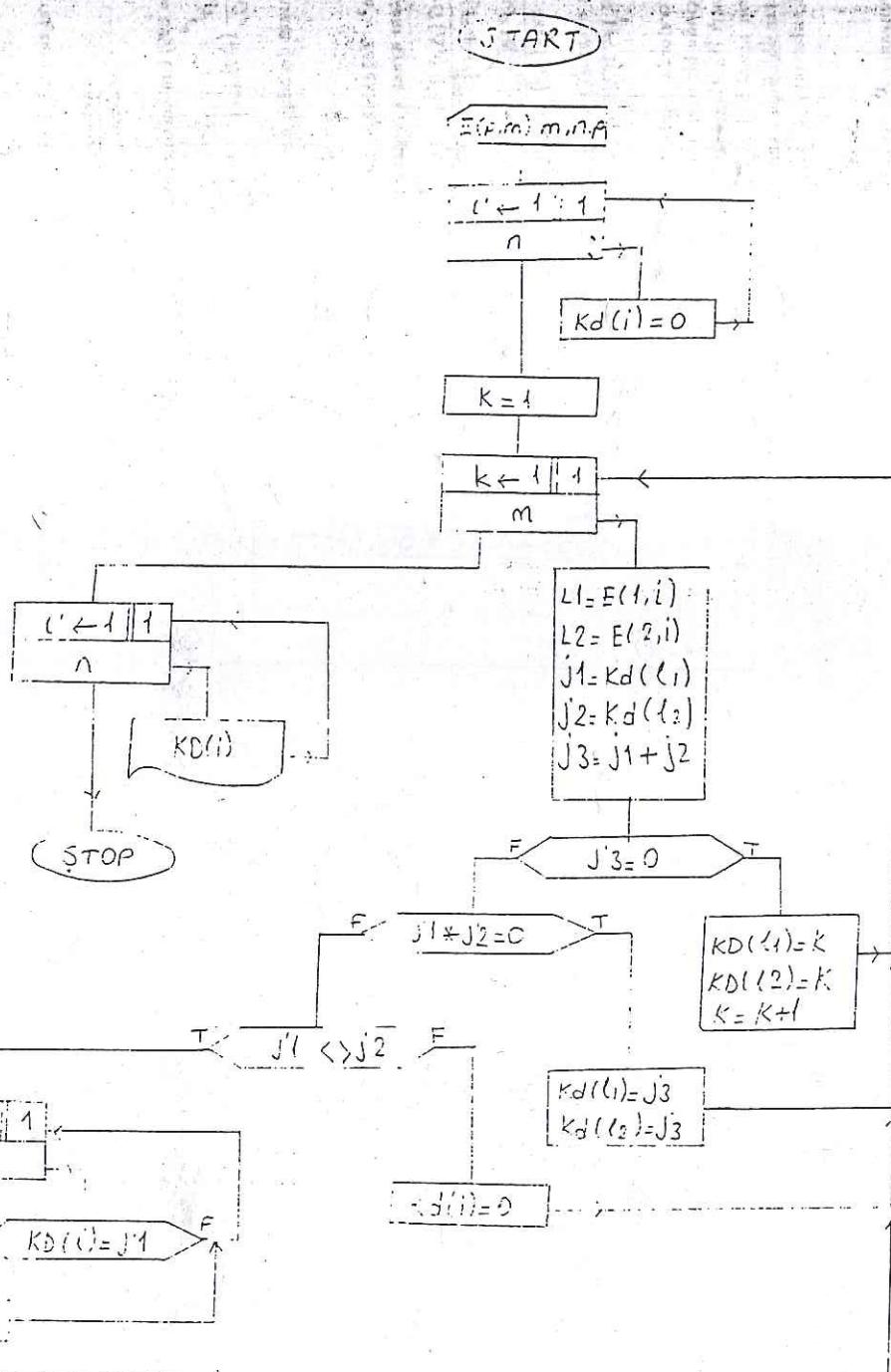
```

Program carpomatris( input,output);
Uses crt;
Var
  m,n,p,s,k,i,j :integer;
  a,b,c:array [1..10, 1..10] of integer;
Begin
  clrscr;
  writeln (' m,n ve p sayılarını giriniz');
  Readln (m,n,p);
  writeln ('a matrisini gir ');
  For i:=1 to m do
  Begin
    For j:=1 to n do
    Read (a[i,j]); Readln;
    End;
  writeln ('b matrisini girin ');
  For j:=1 to n do
  Begin
    For i:=1 to p do
    Read (b[i,j]); Readln; End;
    For i:=1 to m do
    For k:=1 to p do
    Begin
      s:=0;
      for j:=1 to n do
      s:= s+(a[i,j]*b[j,k]);
      c[i,k]:=s;
    End;
    For i:=1 to m do
    Begin
  
```

```

    For j:= 1 to p do
    write (c[i,j]:4);
    writeln;
    End;
    End.
  
```

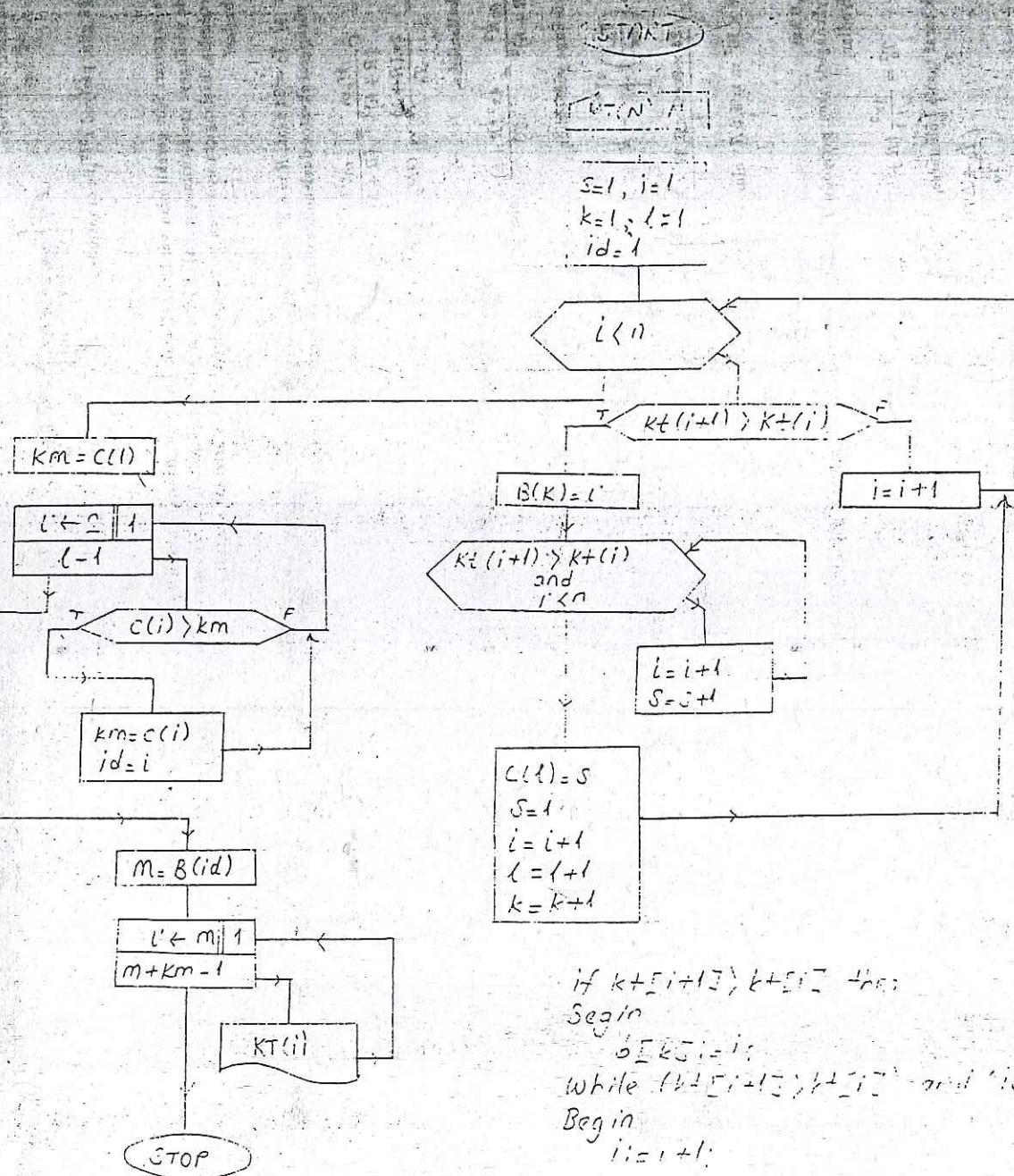
Minimum Spanning Tree algoritmosu ve programı.



end;
end;
for i:=1 to n do
Write (Kd[i]);
End.

Program minimum input, output uses crt;
var
k, i, n, j, m, l, j1, j2, j3 : int;
p, l1, l2 : integer;
kd:array [1..10] of integer;
e:array [1..3, 1..15] of integer;
Begin
clrscr;
writeln ('matrisin boyunu gir');
readln (p,m);
writeln ('matris elementlerini');
for i:=1 to p do
Begin
for j:=1 to m do
Read (e[i,j]); readln;
end;
writeln ('Dügüm sayisini gir');
readln (n);
for i:=1 to n do
kd[i]:=0; k:=1;
for i:=1 to m do
Begin
l1:=e[1,i];
l2:=e[2,i];
j1:=kd[l1];
j2:=kd[l2];
j3:=j1+j2;
if j3=0 then
begin
kd[l1]:=k;
kd[l2]:=k;
k:=k+1;
end
else
begin
if j1*j2=0 then
begin
kd[l1]:=j3;
kd[l2]:=j3;
end
else
begin
if j1 < j2 then
begin
for l:=1 to n do
Begin
if kd[l]:=j1 then
kd[l]:=j2
else
kd[l]:=0;
end;
end;

En fazla yükseltmenin olduğu aralığı bulan algoritma ve programı.



```

Program yükseltme(input,output);
uses crt;
var
  n,i,s,k,l,id,km,m:integer;
  t+:array [1..30] of integer;
  c,s:array [1..10] of integer;
begin
  clrscr;
  writeln('Dizi boyutunu giriniz:');
  readln(n);
  writeln('Elemanlarini gir');
  for i:=1 to n do
    read(t[i]);
  s:=1; l:=1; k:=1; id:=1; i:=1;
  while i<n do
    begin
    
```

```

      if k+t[i+1]>t[i] then
        begin
          i:=i+1;
          while t[i+1]>t[i] and i<n do
            begin
              i:=i+1;
              s:=s+1;
            end;
          id:=i+1;
          end;
        km:=c[i];
        for i:=m to m+km-1 do
          write(k+t[i]:5);
        end;
      end;
    end;
    i:=i+1;
    end;
    km:=c[i];
    for i:=l to l+k-1 do
      write(k+t[i]:5);
    end;
    l:=i+1;
    k:=k+1;
    id:=i+1;
    end;
    m:=b[id];
    for i:=m to m+km-1 do
      write(k+t[i]:5);
    end;
  end;
end.

```

(37)

Pascal üçgenini veren algoritma ve programı.

Program pascal üçgeni (input, output),

uses crt;

var

$p, i, j, k : integer;$

$a : array [1..40, 1..40] of integer;$

Begin

clrscr;

writeln ('p sayisini giriniz');

readln (p);

$n := \lfloor \frac{p}{2} \rfloor - 1;$

for $i := 1$ to p do

for $j := 1$ to n do

$a[i,j] := 0;$

$i := 1, p; j := 1;$

$i := 2;$

while $i <= p$ do

begin

for $j := 1$ to $n-2$ do

begin

$k := j+1;$

$a[i,k] := a[i-1,j] + a[i-1,j+2];$

end;

$i := i+1;$

end;

$a[p,1] := 1;$

$a[p,n] := 1;$

for $i := 1$ to p do

begin

for $j := 1$ to n do

writeln ($a[i,j]:3$);

writeln;

end;

end.

START

p

$n = \lfloor \frac{p}{2} \rfloor - 1$

$i \leftarrow 1 - 1$

p

$j \leftarrow 1 - 1$

n

$A(i,j) = 0$

$A[i,j] = 1$

$i = 2$

$i \leftarrow p$

$j \leftarrow 1 - 1$

$n - 2$

$k = j + 1$

$A[i,k] = A[i-1,j] + A[i-1,j+2]$

$A(p,1) = 1$

$A(p,n) = 1$

$i \leftarrow 1 - 1$

p

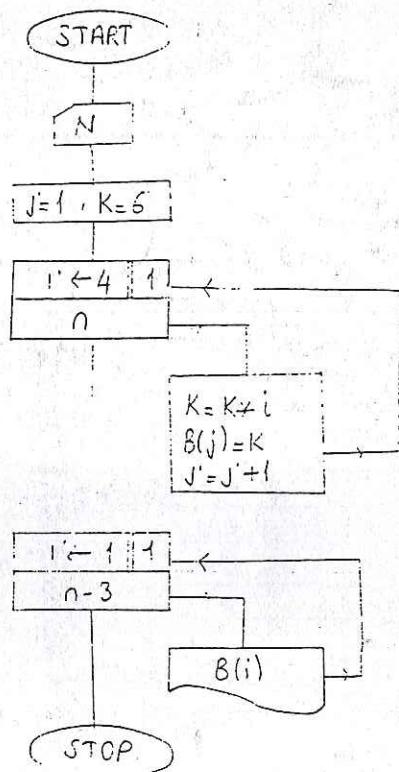
$j \leftarrow 1 - 1$

n

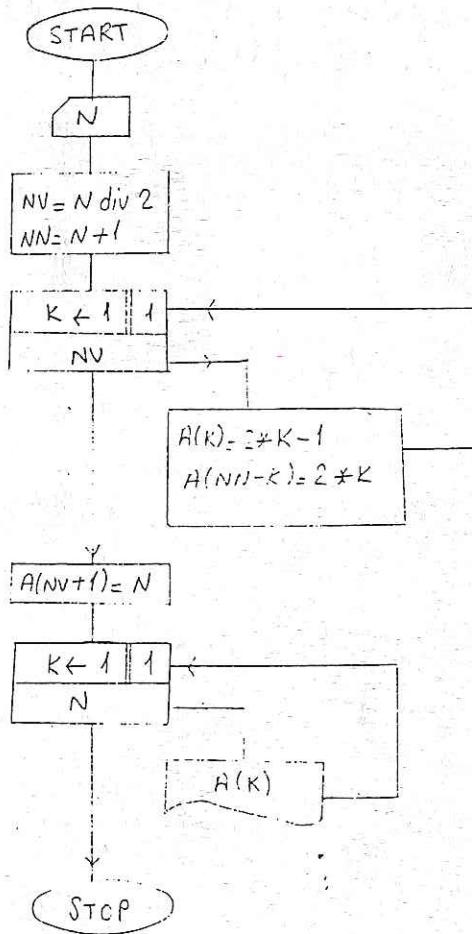
$A(p,i)$

END

3-25 sayıları arasındaki sayıların faktöriyellerini hesaplayıp bir dizİYE yerlestire algoritma ve programı



Verilen bir N sayısına göre $1 \ 3 \ 5 \ 7 \dots N \ 6 \ 4 \ 2$ dizisini oluşturan algoritma ve programı.



Program faktör (input, output);

uses crt;

var

$n, j, i, k : integer;$
 $b : array[1..30] of integer;$

Begin

clrscr;

writeln ('n sayısını gir');

Readln(n);

$j := 1; k := b;$

for $i := 4$ to n do

Begin

$k := k * i;$

$b[j] := k;$

$j := j + 1;$

end;

for $i := 1$ to $n - 3$ do

write ($b[i]$);

End.

Program dizigiiki (input, output);

uses crt;

var

$n, nn, nv, k : integer;$
 $a : array [1..10] of integer;$

Begin

clrscr;

writeln ('n sayısını gir');

Readln(n);

$nv := n div 2;$

$nn := n + 1;$

for $k := 1$ to nv do

Begin

$a[nv + 1] := n;$

$a[nn - k] := 2 * k;$

End;

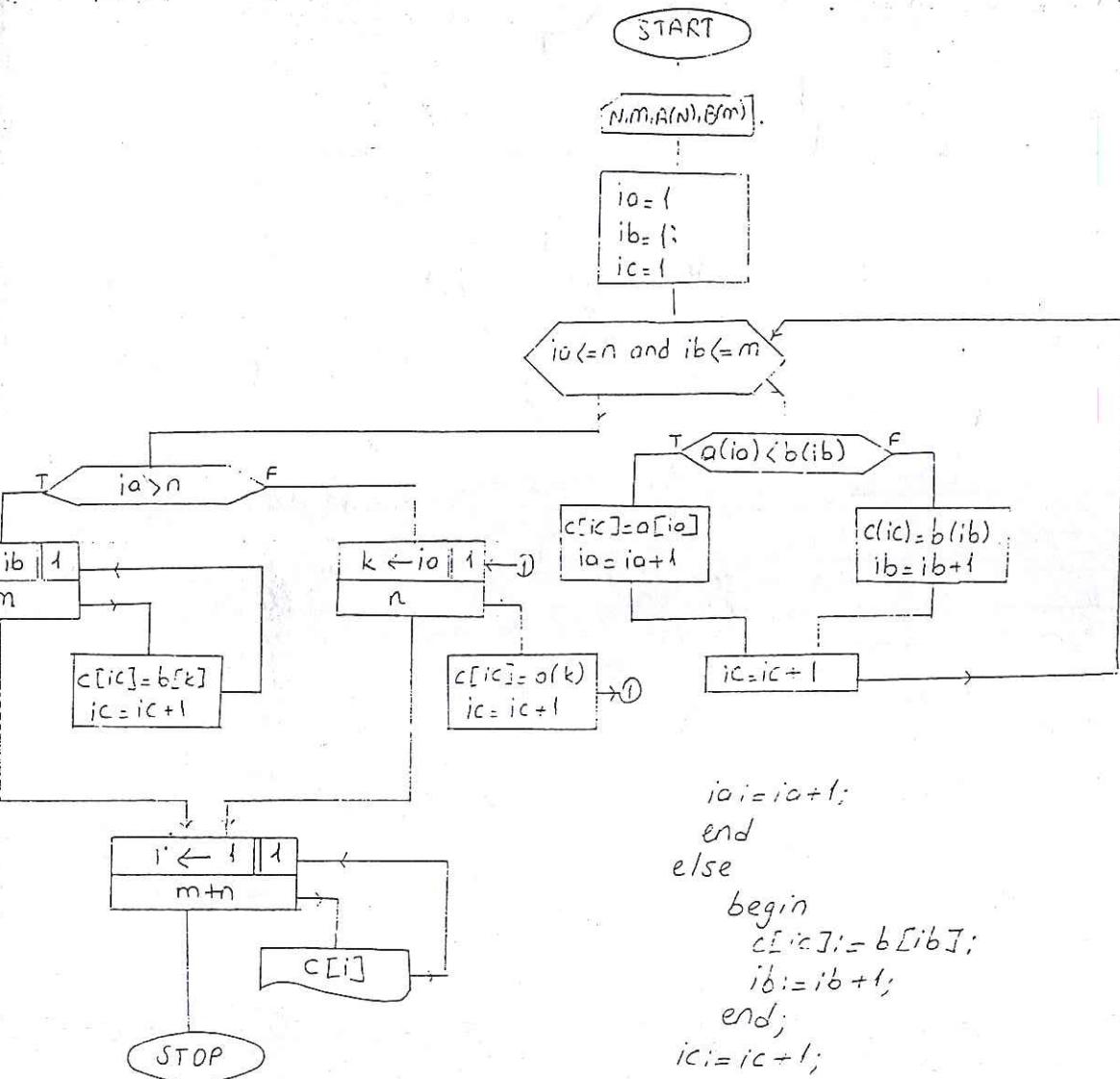
$a[nv + 1] := n;$

for $k := 1$ to n do

write ($a[k]$);

End.

Küçükten büyüğe sıralı N elemanlı ve m elemanlı iki ayrı dizinin verildiğinde bu dizileri kullanarak $N+m$ elemanlı sıralı tek bir dizisi elde eden algoritma ve programı. (merge sort).



```

Program merge (input, output);
uses crt;
var
  m, io, ib, ic, n, i, k : integer;
  a, b : array [1..10] of integer;
  c:array [1..20] of integer;
begin
  clrscr;
  writeln ('Dizilerin boyalarini giriniz');
  Readln (n,m);
  writeln ('A dizisini gir');
  for i := 1 to n do
    Read (A[i]);
  writeln ('B dizisini gir');
  for i := 1 to m do
    Read (B[i]);
  io:=1; ib:=1; ic:=1;
  while (io<=n) and (ib<=m) do
  begin
    if a[io]< b[ib] then
      begin
        c[ic]:=a[io];
        io:=io+1;
        ic:=ic+1;
      end
    else
      begin
        c[ic]:=b[ib];
        ib:=ib+1;
        ic:=ic+1;
      end;
  end;
  for i := 1 to m+n do
    write (c[i]);
end.

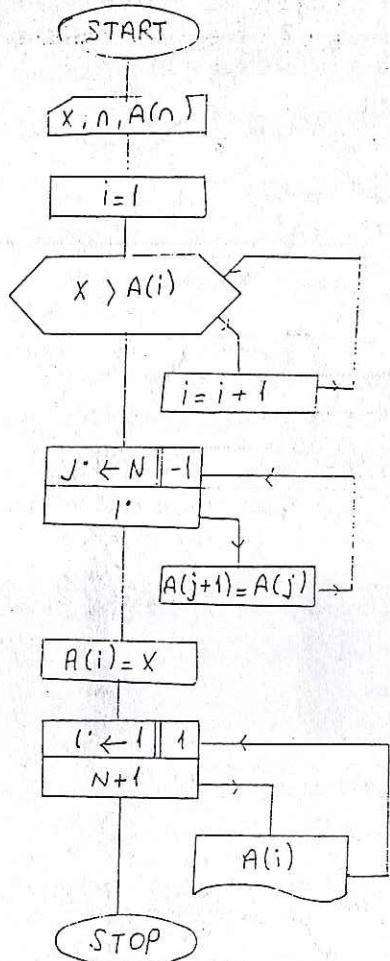
```

```

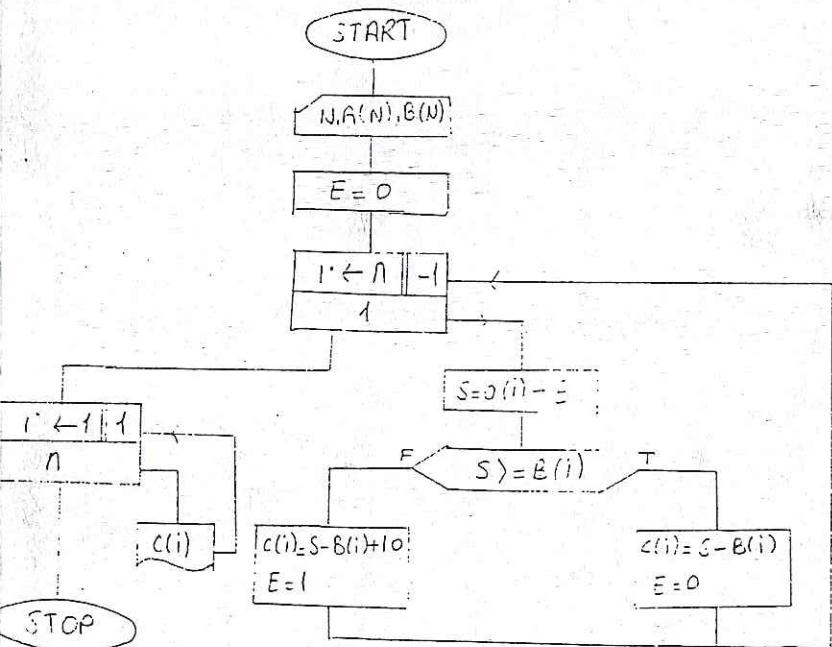
  io := io + 1;
end
else
begin
  c[ic] := b[ib];
  ib := ib + 1;
end;
ic := ic + 1;
end;
if io > n then
begin
  for k := ib to m do
  begin
    c[ic] := b[k];
    ic := ic + 1;
  end;
end
else
begin
  for k := io to n do
  begin
    c[ic] := a[k];
    ic := ic + 1;
  end;
end;
for i := 1 to m+n do
  write (c[i]);
End.

```

Verilen küçükten büyüğe sıralı bir dizeye bir sayının yerleştirilmesini sağlayan algoritma ve programı.



İki sayının farkını bulan algoritma ve programı.



```

begin
c[i]:=s-b[i]+10;
e:=1;
end;
end;
For i:=1 to n do
Write(c[i]);
    
```

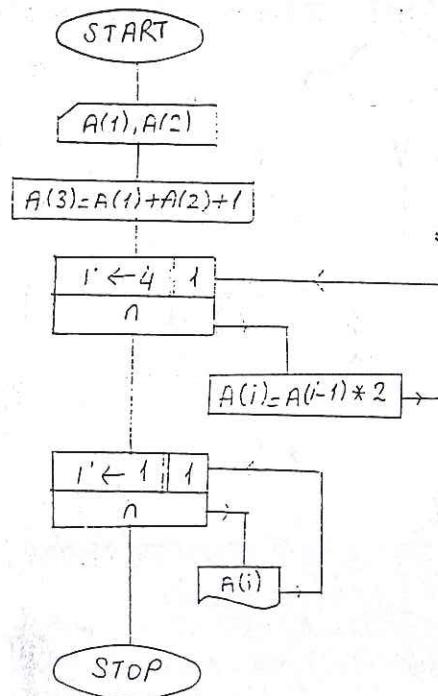
Program yerlesim (input, output);
uses crt;
var
n, i, j, x: integer;
a: array [1..15] of integer;
Begin
clrscr;
writeln ('Dizi boyutu ve sayı');
readln (n, x);
writeln ('Dizi elemanları');
for j := 1 to n do
read (a[j]);
i := 1;
while x > a[i] do
i := i + 1;
for j := n downto i do
a[j+1] := a[j];
a[i] := x;
for i := 1 to n+1 do
write (a[i]:5);
End.

Program cikarma (input, output);
uses crt;
var
s, e, n, i: integer;
a, b, c: array [1..10] of integer;
Begin
clrscr;
writeln ('Dizi boyutunu gir');
readln (n);
writeln ('A dizisini gir');
for i := 1 to n do
read (a[i]);
writeln ('B dizisini gir');
for i := 1 to n do
read (b[i]);
e := 0;
for i := n downto 1 do
begin
s := a[i] - e;
if s >= b[i] then
begin
c[i] := s - b[i];
e := 0;
end

(41)

Verilen bir dizinin her bir elemanı kendinden önce gelen elemanların toplamının bir fazlasına eşittir. 1. ve 2. elemanı verilen bir dizinin ilk 20 elemanın değerlerini bulup dizeye yerlestiren algoritma ve programı.

i)



Program tanzimbir (input, output);

uses crt;

var

i, n : integer;

a:array [1..20] of integer;

Begin

clrscr;

Writeln ('Dizinin ilk iki elemanını gir');

Readln (A[1], A[2]);

Writeln ('n·sayısını gir');

Readln (n);

A[3]:= A[1]+A[2]+1;

for i:= 4 to n do

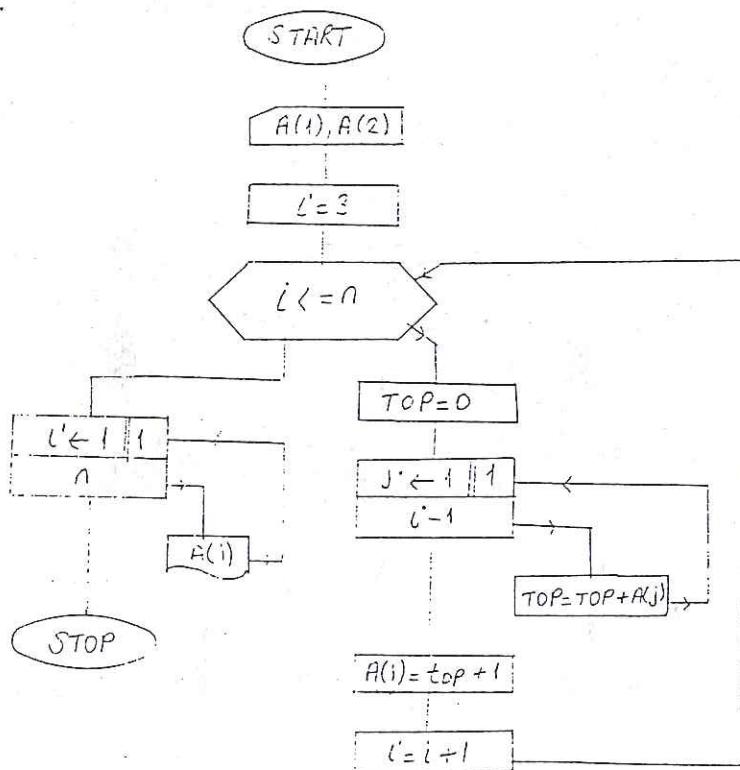
A[i]:= A[i-1]*2;

for i:= 1 to n do

Write (A[i]:6);

End.

ii)



Program tanzimiki (input, output);

uses crt;

var

i, j, n, top : integer;

a:array [1..20] of integer;

Begin

clrscr;

Writeln ('Dizinin ilk iki elemanını gir');

Readln (a[1]; a[2]);

Writeln ('n sayısını giriniz');

Readln (n);

i:=3;

while i <= n do

Begin

top := 0;

for j := 1 to i-1 do

begin

top := top + a[j];

end;

a[i]:= top + 1;

i:= i+1;

end;

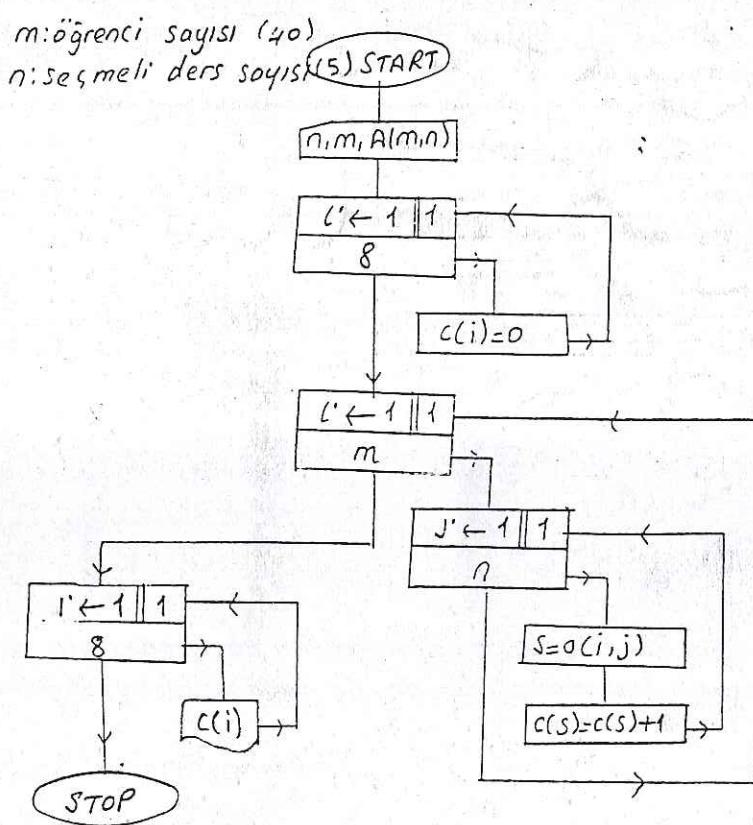
For i := 1 to n do

Write (a[i]:4);

Readln;

End.

40 kişilik bir sınıfta öğrenciler kredilerini doldurabilmek için 5 ders almak zorundadırlar. Bu dersler 8 ayrı ders arasından seçilebilmektedir. Dersler 1 ile 8 arasında numaralandırılmıştır. Bu sınıfın seçtiği dersler 40×5 'lik bir matris olarak bilgisayara girileceğine göre her dersin kaç öğrenci tarafından alındığını bulan algoritmo ve programı.



Program dersler (input, output);

uses crt;

var

i, m, n, j, s : integer;

a:array [1..40,1..5] of integer;

c:array [1..8] of integer;

Begin

clrscr;

writeln ('matris boyutu');

readln (m,n);

writeln ('matrisi gir');

for i:= 1 to m do

begin

for j:= 1 to n do

read (a[i,j]);

readln;

for i:= 1 to 8 do

c[i]:=0;

for i:= 1 to m do

for j:= 1 to n do

begin

s := a[i,j];

c[s]:=c[s]+1;

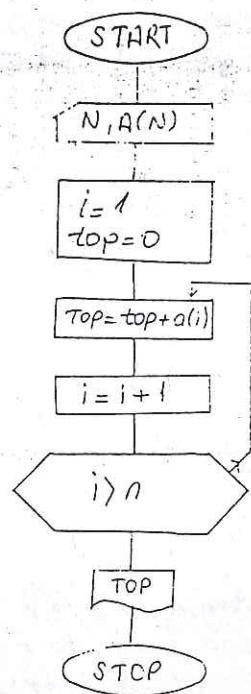
end;

for i:= 1 to 8 do

writeln (c[i]:5);

end.

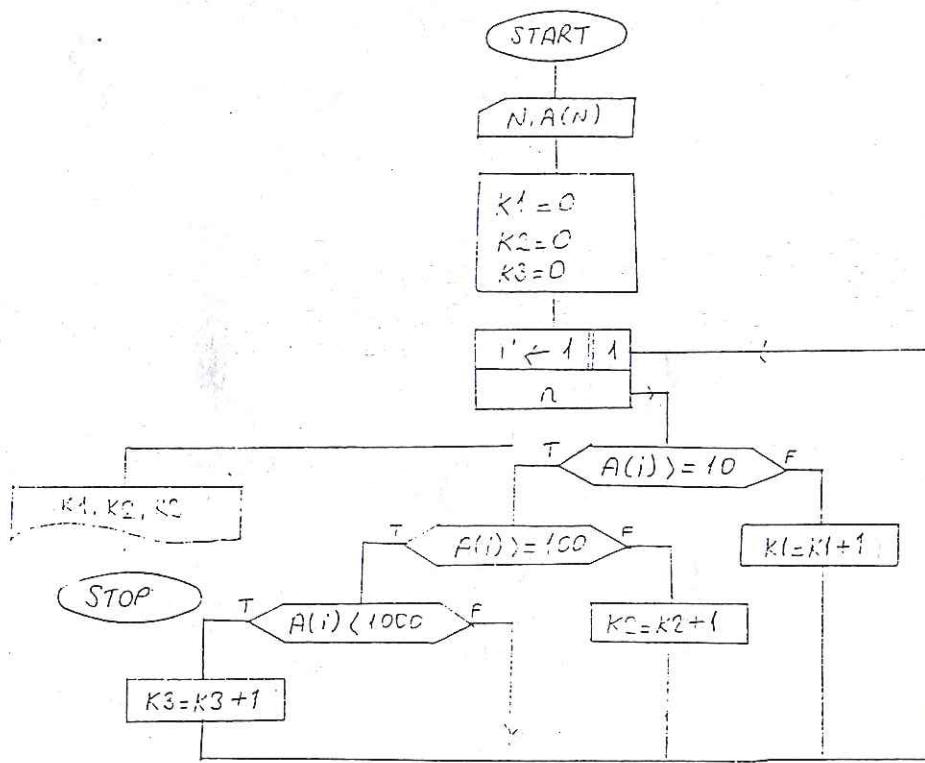
iii)



```

Program toplu (input, output);
uses crt;
var
  i,n,top : integer;
  a:array [1.. 10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu gir');
  readln (n);
  writeln ('Dizi elementlerini gir');
  for i:=1 to n do
    read (a[i]);
  top := 0;
  i := 1;
  repeat
    top := top + a[i];
    i := i + 1;
  until i > n;
  writeln (top);
end.
  
```

Tamsayılarından oluşan bir dizi veriliyor. Bu dizi elementlerinden kaç tanesinin bir basamaklı, kaç tanesinin iki basamaklı, kaç tanesinin de üç basamaklı olduğunu bulan algoritma ve programı.



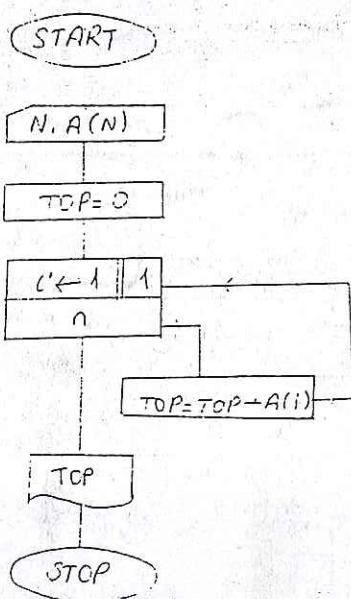
```

Program bosmak (input, output);
uses crt;
var
  k1,k2,k3,n,i : integer;
  a:array [1.. 10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu gi');
  readln (n);
  writeln ('Dizi elementlerini');
  for i:=1 to n do
    read (a[i]);
  k1 := 0; k2 := 0; k3 := 0;
  for i:=1 to n do
    begin
      if a[i] > 10 then
        begin
          if a[i] > 100 then
            begin
              if a[i] < 1000 then
                k3 := k3 + 1;
              end
            end
          else
            k2 := k2 + 1;
        end
      else
        k1 := k1 + 1;
    end;
  write (k1;i3,k2;i3,k3;i3);
end.
  
```

(41)

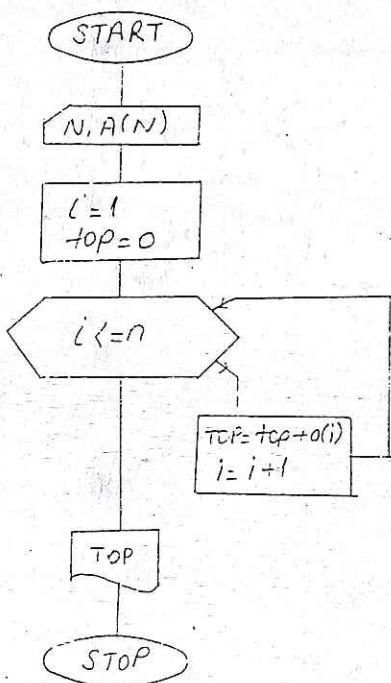
Bir dizinin elemanlarının toplamlarını bulan algoritma ve programı.

i)



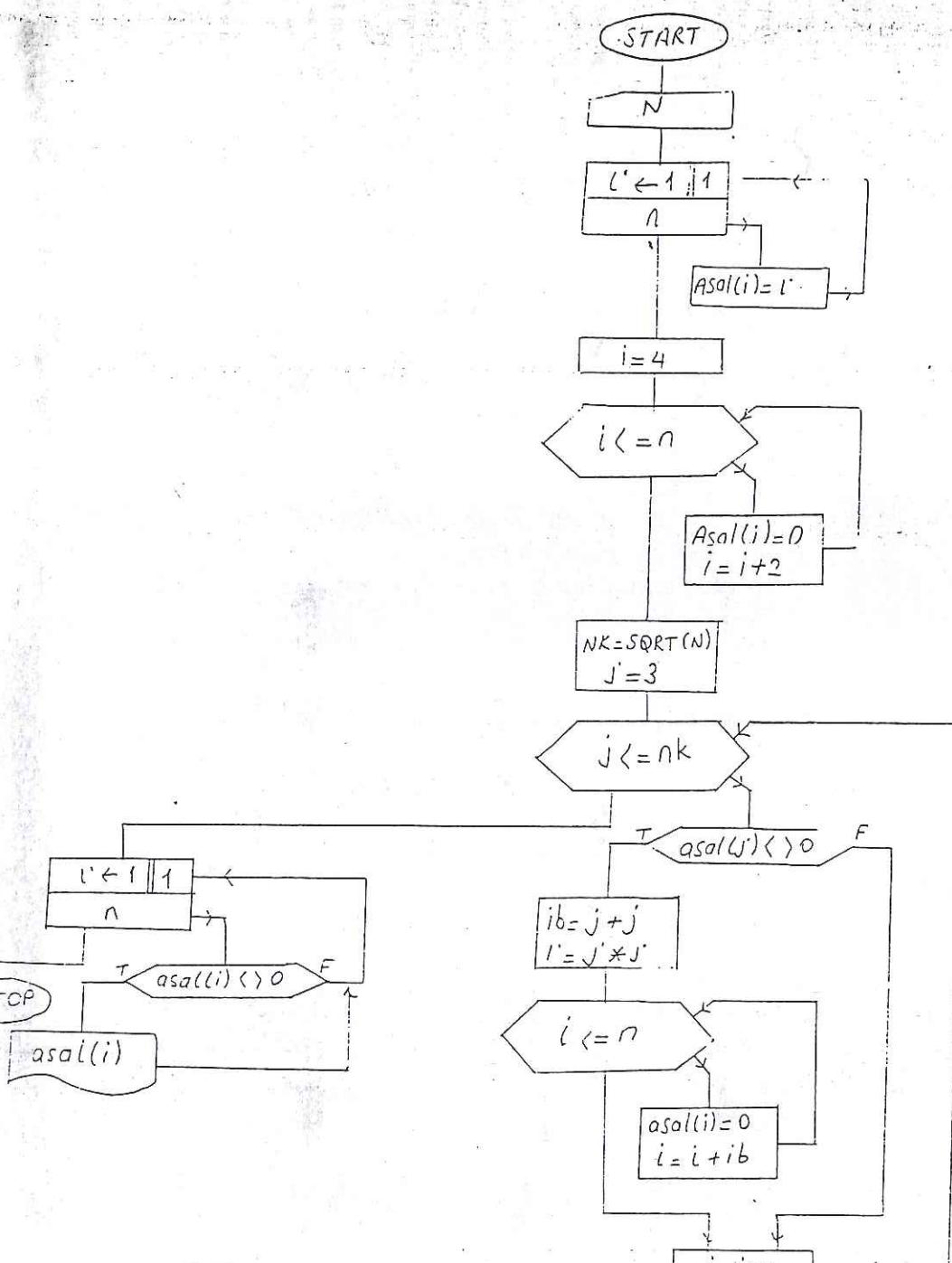
Program dizi toplama (input, output);
uses crt;
var
i,n,top : integer;
a:array [1..10] of integer;
Begin
clrscr;
Writeln ('Dizi boyutunu giriniz');
Readln (n);
Writeln ('Dizi elemanlarını gir');
For i:=1 to n do
Read (a[i]);
TOP := 0;
For i:=1 to n do
TOP := TOP + a[i];
Writeln (top);
End.

ii)



Program dizi (input, output);
uses crt;
var
i,n,top : integer;
a:array [1..10] of integer;
Begin
clrscr;
Writeln ('Dizi boyutunu gir');
Readln (n);
Writeln ('Dizi elemanlarını gir');
For i:=1 to n do
Read (a[i]);
TOP := 0; i := 1;
While i <= n do
Begin
top := top + a[i];
i := i + 1;
End;
Writeln (top);
End.

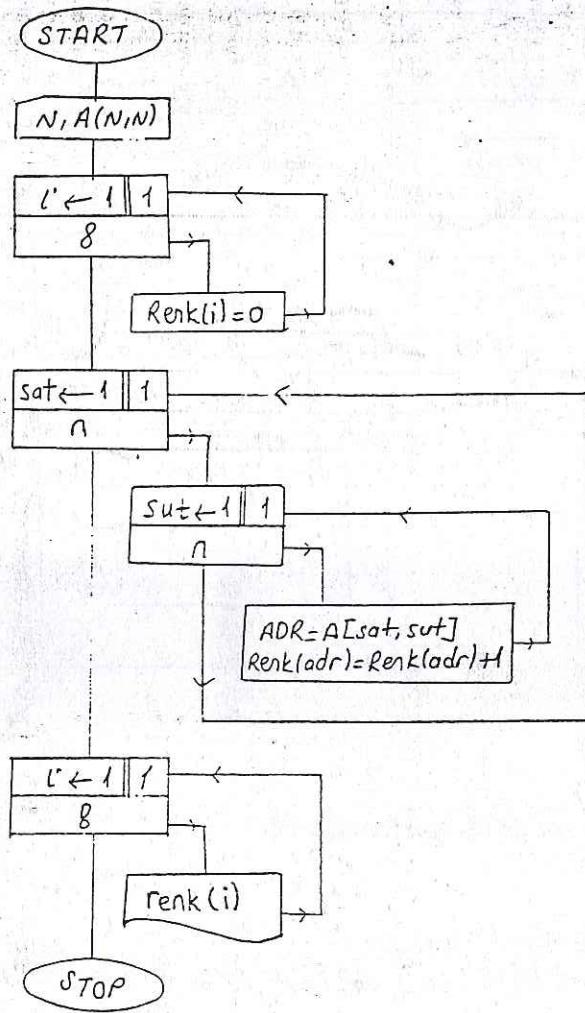
Girilen bir sayıya kadar olan asal sayıları veren algoritma ve programı.



```

Program asalsayi(input, output);
uses crt;
var
  i, j, nk, ib: integer;
  asal: array [1..100] of integer;
begin
  clrscr;
  writeln('n sayısını gir');
  Readln(n);
  for i := 1 to n do
    asal[i] := i;
  i := 4;
  while i <= n do
    begin
      asal[i] := 0;
      i := i + 2;
    end;
  nk := trunc(sqrt(n));
  j := 3;
  while j <= nk do
    begin
      if asal[j] <> 0 then
        begin
          ib := j + j;
          i := j * j;
          while i <= n do
            begin
              asal[i] := 0;
              i := i + ib;
            end;
          j := j + 2;
        end;
      for i := 1 to n do
        begin
          if asal[i] <> 0 then
            write(asal[i], ' ');
        end;
      end;
    end;
  end.
  
```

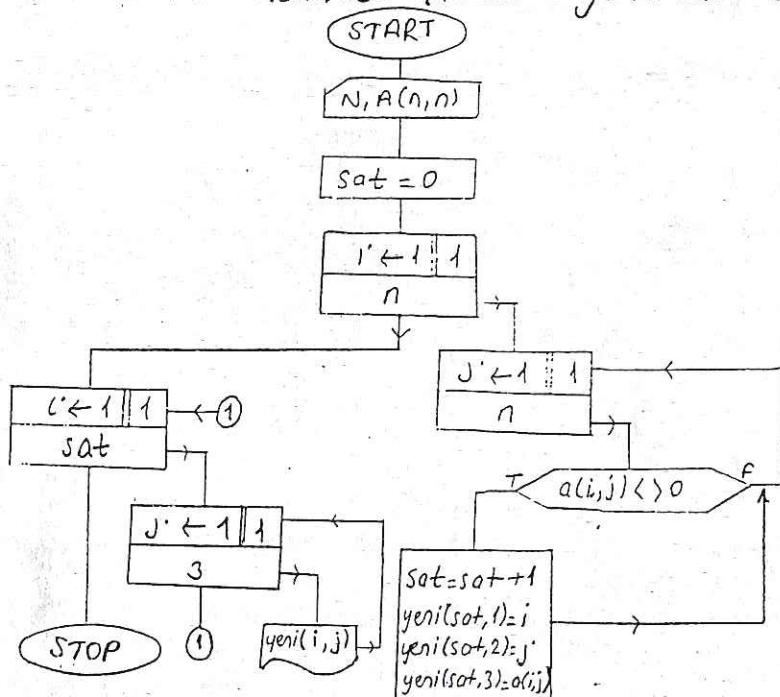
Renklerin 1-8 arası sayılarla gösterildiği 50'ye 50'lik bir resimde her rengin kaç defa kullanıldığını bulan algoritma ve programı.



```

Program Boyama(input, output);
Uses crt;
Var
  n, i, sat, sut, adr: integer;
  a: array [1..50, 1..50] of integer;
  Renk: array [1..8] of integer;
Begin
  clrscr;
  Writeln ('Matris boyutunu giriniz');
  Readln (n);
  Writeln ('Matrisi gir');
  For sat := 1 to n do
    Begin
      for sut := 1 to n do
        Read (a[sat, sut]); readln;
    End;
  For i := 1 to 8 do
    Renk[i] := 0;
  For sat := 1 to n do
    Begin
      for sut := 1 to n do
        begin
          adr := a[sat, sut];
          renk[adr] := renk[adr] + 1;
        end;
    End;
  For i := 1 to 8 do
    Write (Renk[i]; 3);
  End.
  
```

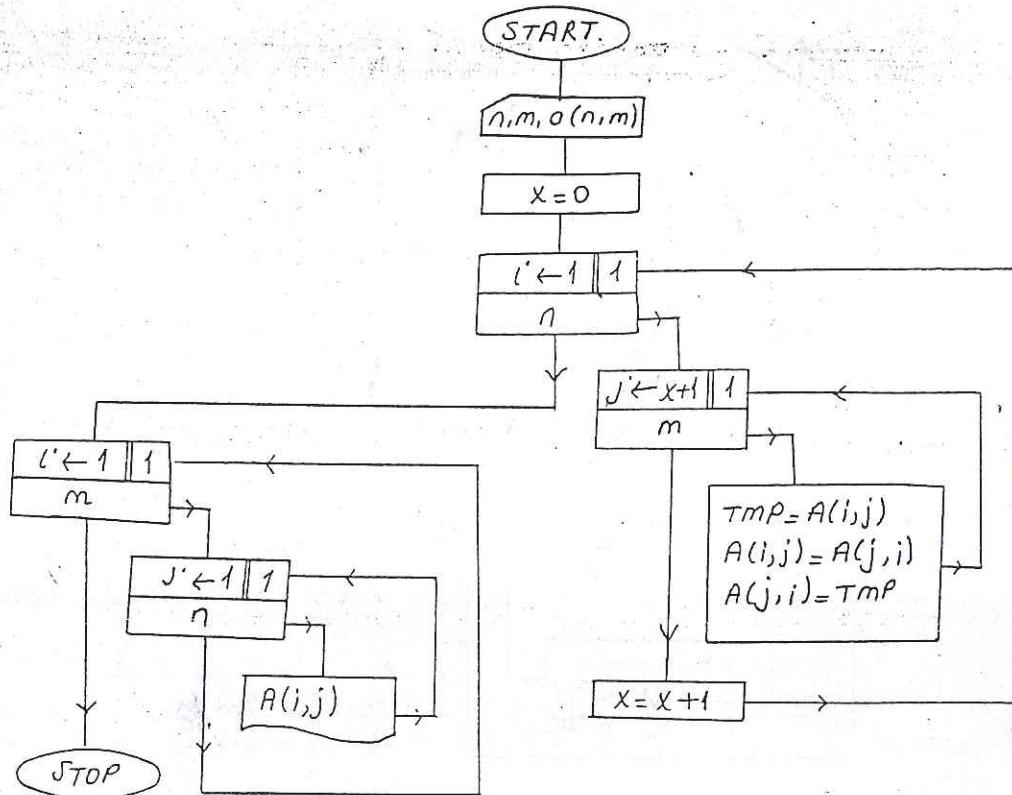
Elemanlarının yarısından çoğu sıfır olan matrise "sparse matris" denilir. Verilen bir sparse matrisin sıfırdan farklı değerlerini saklamak için öyle bir yapı kurunuz ki istenilen bir satır ve sütundaki değerin bu yeni yapı kullanılarak elde edilebilmesi mümkün olsun. Bu işlemin algoritması ve programı.



```

Program sparse(input, output);
uses crt;
var
  n, i, j, sat: integer;
  yeni, a: array [1..10, 1..10] of integer;
begin
  clrscr;
  Writeln ('Matrisin boyutunu gir');
  Readln (n);
  Writeln ('Matris elemanlarını gir');
  for i := 1 to n do begin
    for j := 1 to n do
      Read (a[i, j]); readln;
    end;
  sat := 0;
  for i := 1 to n do
    for j := 1 to n do
      begin
        if a[i, j] > 0 then
          begin
            sat := sat + 1;
            yeni[sat, 1] := i;
            yeni[sat, 2] := j;
            yeni[sat, 3] := a[i, j];
          end;
      end;
  end.
  
```

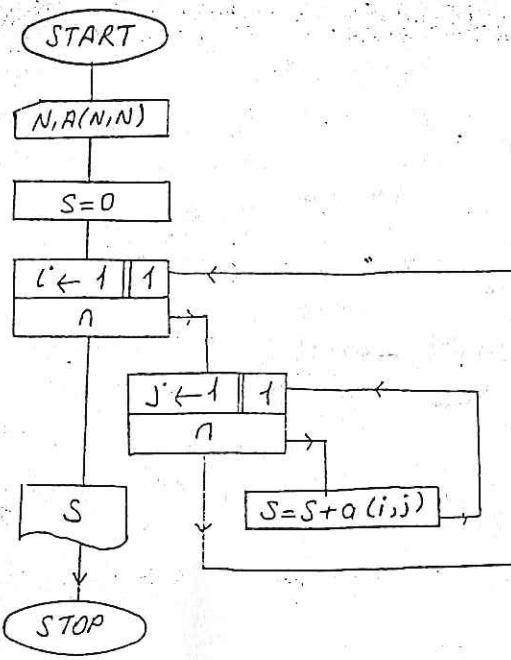
Bir matrisin transpozesini bulan algoritma ve programı.



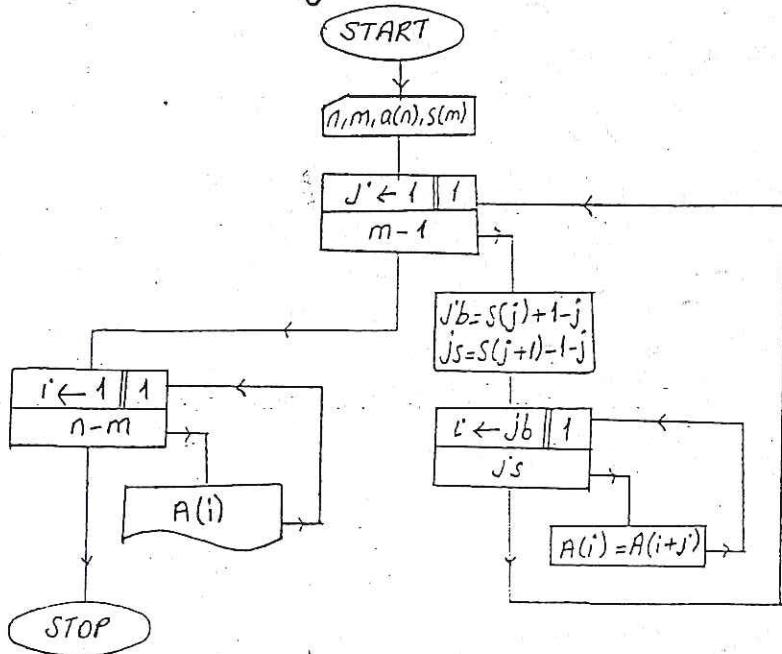
Program transpose (input,output);
Uses crt;
Var

```
n,m,x,i,j,tmp:integer;
a:array [1..10, 1..10] of integer;
Begin
clrscr;
Writeln ('matrisin boyutu');
Readln (n,m);
Writeln ('matrisin elementlari');
For i:=1 to n do
Begin
For j:=1 to m do
Read (a[i,j]); Readln; End;
x:=0;
for i:=1 to n do
Begin
tmp:=a[i,j];
a[i,j]:=a[j,i];
a[j,i]:=tmp;
End;
x:=x+1;
End;
For i:=1 to m do
Begin
For j:=1 to n do
Write (a[i,j]:5);
Writeln;
End;
End.
```

Elemenləri sadece 0 və 1'den olusun $N \times N$ 'lik bir matriste kəs elementin 1'den olustugunu bulan algoritma ve programı.



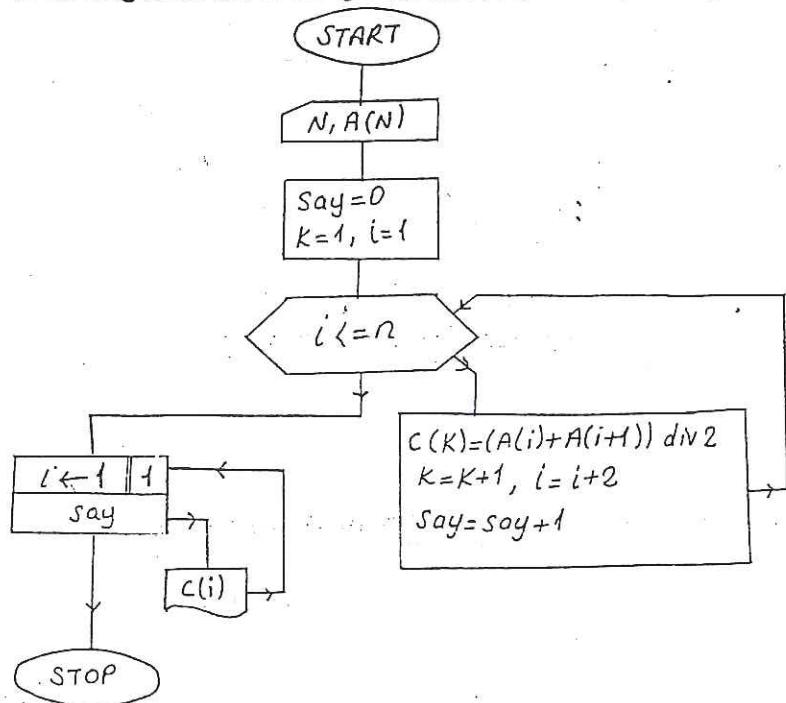
Kayıt silen algoritma ve programı.



Program birsayisi (input, output);
Uses crt;
Var
 $n, i, j, s : integer;$
 $a : array [1..10, 1..10] of integer;$
Begin
clrscr;
writeln ('matriç boyutu');
readln (n);
writeln ('matriçin elementləri');
for $i := 1$ to n do
begin
for $j := 1$ to n do
read (a[i, j]); readln; End;
 $s := 0$;
for $i := 1$ to n do
for $j := 1$ to n do
 $s := s + a[i, j]$;
writeln ('s');
readln;
End.

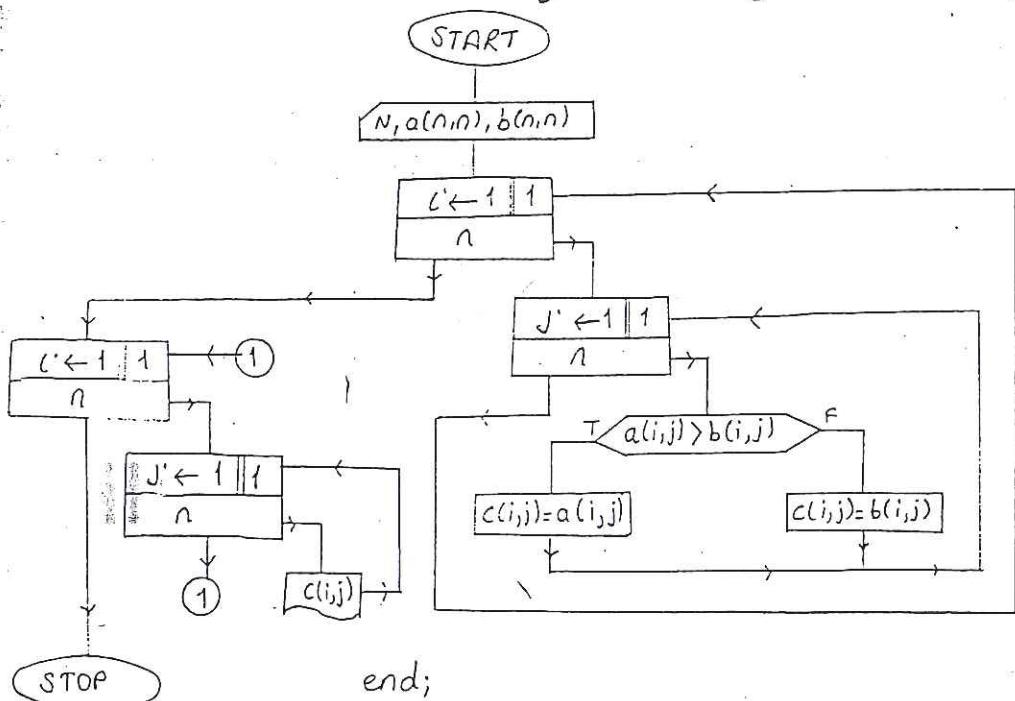
Program koyik (input, output);
Uses crt;
Var
 $n, m, i, j, jb, js : integer;$
 $a, s : array [1..10] of integer;$
Begin
clrscr;
writeln ('a ve s dizilərinin boyutu');
readln (n, m);
writeln ('a dizisinin elementləri');
for $i := 1$ to n do
read (a[i]);
writeln ('s dizisinin elementləri');
for $i := 1$ to m do
read (s[i]);
for $j := 1$ to $m-1$ do
begin
 $jb := s[j] + 1 - j$;
 $js := s[j+1] - 1 - j$;
for $i := jb$ to js do
 $a[i] := a[i+j]$;
End;
for $i := 1$ to $n-m$ do
write (a[i]:3);
End.

50 elementli bir dizinin 1. ve 2. elemanları toplamının yarısı başka bir dizinin ilk elemanı, 3. ve 4. elemanları toplamının yarısı 2. elemanı... şeklindedir. Bu işlemi yapan algoritma ve programı.



Program diz (input, output);
 Uses crt;
 Var
 i, n, sayik: integer;
 a, c: array [1..10] of integer;
 Begin
 clrscr;
 Writeln ('dizinin eleman sayisi');
 Readln (n);
 Writeln ('dizi elemanları');
 For i := 1 to n do
 Read (a[i]);
 Say := 0; k := 1; i := 1;
 While i <= n do
 Begin
 c[k] := (a[i] + a[i + 1]) div 2;
 k := k + 1;
 i := i + 2;
 say := say + 1;
 End;
 for i := 1 to say do
 Write (c[i]: 3);
 End;

iki tane $N \times N$ 'lik matris veriliyor. Bu matrisleri karşılaştırarak aynı satır ve sütundaki elemanlardan büyük olanını aynı satır ve numaradaki C matrisine yazdırın algoritma ve programı.

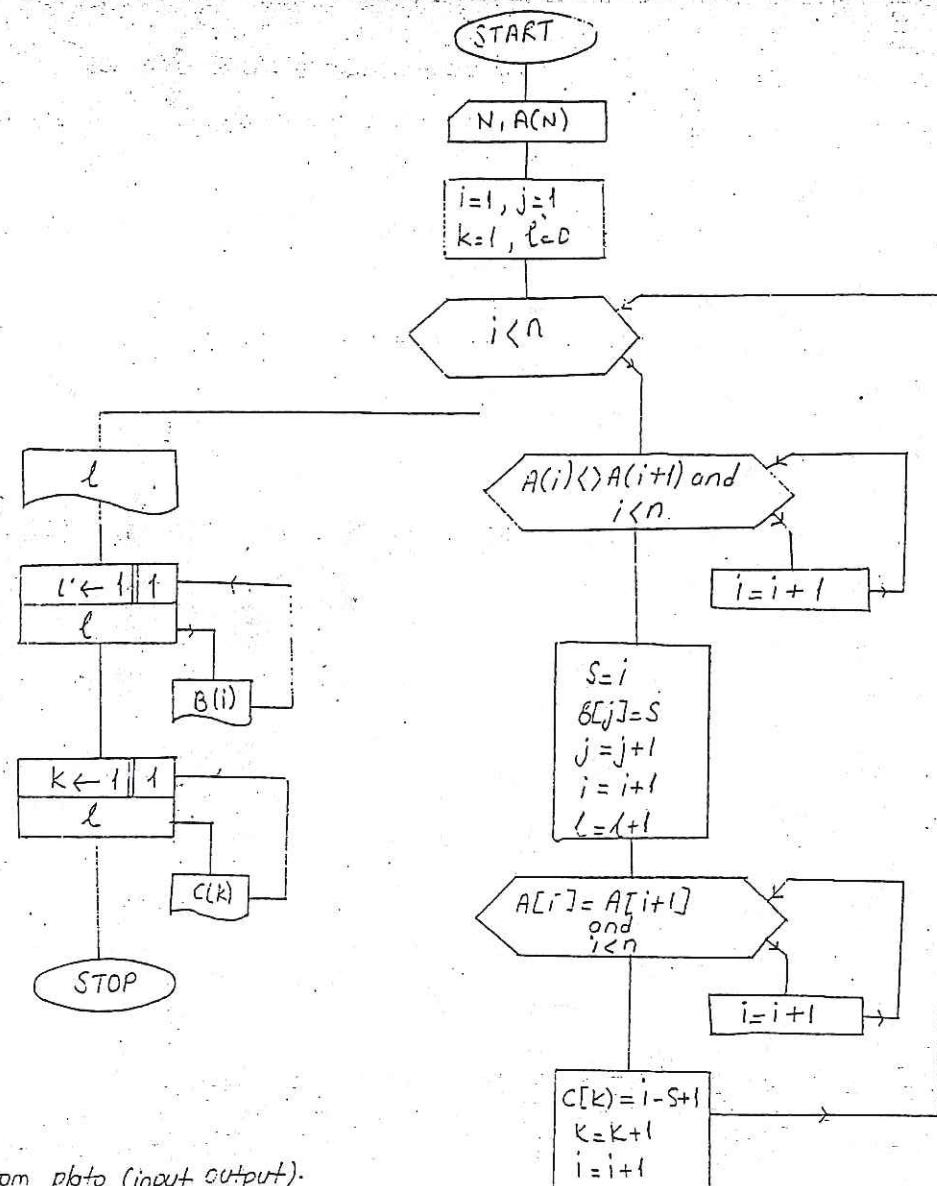


```

    end;
    for i := 1 to n do
    Begin
    for j := 1 to n do
    Write (c[i, j]: 2);
    Writeln;
    End;
    End.
    
```

Program yenimat (input, output);
 Uses crt;
 Var
 i, j, n: integer;
 a, b, c: array [1..10, 1..10] of integer;
 Begin
 clrscr;
 Writeln ('matris boyutu');
 Readln (n);
 Writeln ('a matrisi');
 For i := 1 to n do
 Begin
 for j := 1 to n do
 Read (a[i, j]); Readln; E.
 Writeln ('b matrisi');
 For i := 1 to n do
 Begin
 for j := 1 to n do
 Read (b[i, j]); Readln; E.
 For i := 1 to n do
 For j := 1 to n do
 Begin
 if a[i, j] > b[i, j] then
 c[i, j] := a[i, j];
 End;
 End;

Plato algoritması ve programı.



Program plato (input, output);

uses crt;

var

i,j,k,l,n,s:integer;

a,b,c:array [1..15] of integer;

begin

clrscr;

writeln('a dizisinin boyutu');

readln(n);

writeln('a dizisinin elemanları');

for i:=1 to n do

rand(751);

i:=1; j:=1; k:=1; l:=0;

while i < n do

begin

while (a[i]<>a[i+1]) and (i<n) do

i:=i+1;

s:=i;

b[j]:=s;

j:=j+1;

end;

while (a[i]>=a[i+1]) and (i<n) do

i:=i+1;

c[k]:=i-s+1;

k:=k+1;

i:=i+1;

end;

writeln(l);

for i:=1 to l do

write(b[i]:4);

for k:=1 to l do

write(c[k]:7);

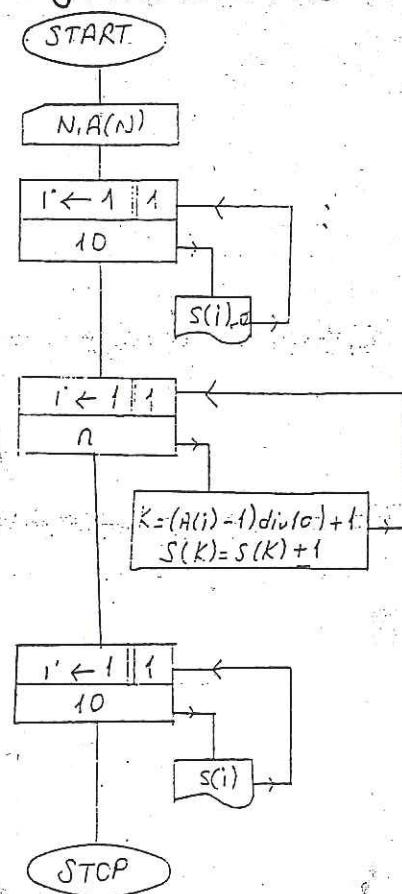
end.

$\ell \rightarrow$ plato sayısı

$B \rightarrow$ Başlangıç dizisi

$C \rightarrow$ Uzunluk dizisi

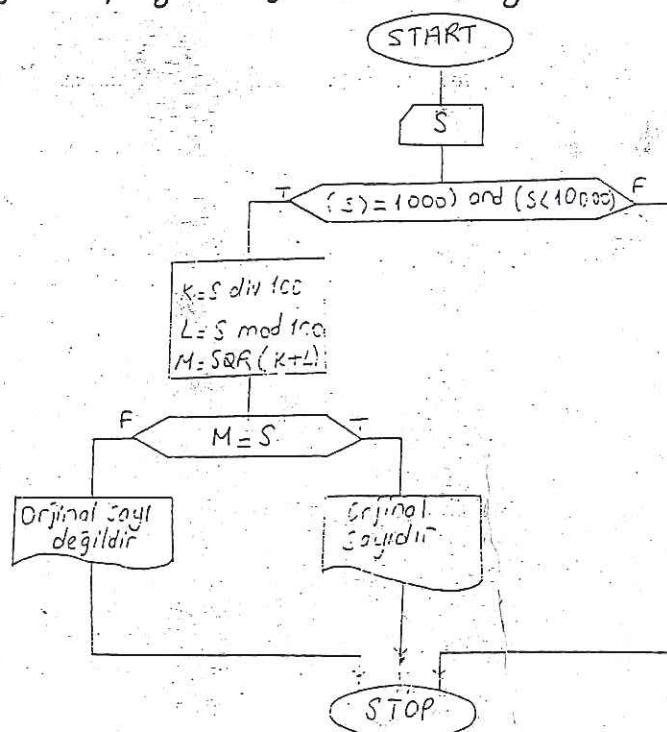
Bir kasabada yaşları 1..100 arasında değişen 400 kişi bulunuyor. Bu kişilerin yaşlarını (1..10, 11..20, 21..30, ..., 91..100) grupları olmak üzere bu 10 gruptan birine yerlestiren algoritma ve programı.



```

Program yasgrubu (input, output);
uses crt;
var
  n, k, i : integer;
  a, s : array [1..10] of integer;
begin
  clrscr;
  writeln ('Dizi boyutunu gir');
  readln (n);
  writeln ('Dizi elementleri');
  for i := 1 to n do
    read (a[i]);
  for i := 1 to 10 do
    s[i] := 0;
  for i := 1 to n do
  begin
    k := ((a[i]-1) div 10) + 1;
    s[k] := s[k]+1;
  end;
  for i := 1 to 10 do
    write (s[i]:3);
  end.
  
```

Dört basamaklı sayılarından ilk iki basamağı ile son iki basamığının toplamının karesi, sayının kendisine eşit olan sayılar orijinal sayı denir. Girilen bir sayının orijinal olup olmadığını saptayan algoritma ve programı.



```

program orjinalsayi (input, output);
uses crt;
var
  m, s, k, l : integer;
begin
  clrscr;
  writeln ('Sayiyi giriniz');
  readln (s);
  if (s)=1000 and (s<10000) then
  begin
    k := s div 100;
    l := s mod 100;
    m := SQR (k+l);
    if m=s then
      writeln ('sayi orjinaldir')
    else
      writeln ('sayi orjinal degildir');
    end
  else
    writeln ('sayi orjinal degildir');
  end.
  
```