**26.**

uses crt;

const

MAXS = 10;

var

i, n : integer;

A : array[1..10] of integer;

procedure klik();

begin

i := i-1;

end;

function klek(x : integer) : integer;

begin

if(x = MAXS) then klek := A[x] \* A[1]

else klek := A[x] \* A[x+1];

end;

function klok() : integer;

var

tmp : integer;

begin

if(i = 0) then klok := i

else begin

tmp := i;

klik();

klok := klok() + klek(tmp);

end;

end;

begin

A[1] := 1; A[2] := 2; A[3] := 3; A[4] := 4; A[5] := 5;A[6] := 6; A[10] := 11; A[9] := 9; A[7] := 8; A[8]:=7;

read(n);

i := n;

writeln(klok());

readkey;;

end.

START

klik ()

Klek (x)

INPUT (n)

i := i-1;

if(x = MAXS) then klek := A[x] \* A[1]

else klek := A[x] \* A[x+1];

RETURN

klik

RETURN

Klek (n)

Klok ()

klok

OUTPUT (n)

if(i = 0) then klok := i

else begin

tmp := i;

klik();

klok := klok() + klek(tmp);

end;

END

RETURN

**27.**

uses crt;

function Proses(x : integer) : integer;

begin

if(x <= 1) then Proses := x

else Proses := Proses(x div 2 \* x mod 2) + Proses(x div 2 + x mod 2);

end;

var x:integer;

begin

read(x);

x:=proses(x);

write(x);

readkey;

end.

Proses (x)

OUTPUT (Proses)

Proses (x)

END

INPUT (Proses)

START

if(x <= 1) then Proses := x

else Proses := Proses(x div 2 \* x mod 2) + Proses(x div 2 + x mod 2);

RETURN

**28.**

uses crt;

function noan(n : integer) : integer;

begin

if(n < 4) then noan := n

else noan := noan(n-1) + noan(n-2) + noan(n-4);

end;

var n:integer;

begin

read(n);

n:=noan(n);

write(n);

readkey;

end.

noan

START

END

OUTPUT

noan

INPUT

if(n < 4) then noan := n

else noan := noan(n-1) + noan(n-2) + noan(n-4);

RETURN

**29.**

uses crt;

var

ar : array[1..10] of integer;

procedure mantaps(n : integer);

var

i : integer;

iNi : integer;

temp : integer;

begin

ar[1]:=1; ar[2]:=6; ar[3]:=2; ar[4]:=3; ar[5]:=4; ar[6]:=7; ar[7]:=2; ar[8]:=4; ar[9]:=2; ar[10]:=1;

if(n > 1) then begin

iNi := n;

for i := 1 to n-1 do begin

if(ar[i] < ar[iNi]) then iNi := i;

end;

temp := ar[n];

ar[n] := ar[iNi];

ar[iNi] := temp;

mantaps(n-1);

end;

end;

var

n:integer;

begin

read(n);

mantaps(n-1);

write(n);

readkey;

end.

Mantaps (n)

START

AR = (1,6,2,3,4,7,2,4,2,1)

if(n > 1) then begin

iNi := n;

for i := 1 to n-1 do begin

if(ar[i] < ar[iNi]) then iNi := i;

end;

temp := ar[n];

ar[n] := ar[iNi];

ar[iNi] := temp;

mantaps(n-1);

INPUT

mantaps

OUTPUT

RETURN

END

**30.**

uses crt;

var x,y: integer;

begin

x := 1;

y := 0;

while(x <= 10) do begin

y := y + x;

x := x + x;

end;

writeln(y);

readkey;

end.

X = 1

Y = 0

START

While x<=10

y = y + x

**False**

x= x + x

INPUT (y)

END

**31.**

uses crt;

var

ar : array[1..10] of integer;

a, b, c, i : integer;

begin

ar[1]:=4; ar[2]:=5; ar[3]:=10; ar[4]:=5; ar[5]:=51; ar[6]:=33; ar[7]:=49; ar[8]:=64; ar[9]:=2; ar[10]:=7;

a := -1;

for i := 1 to 10 do begin

if a = -1 then a := i

else if ar[i] > ar[a] then a := i;

end;

b := -1;

for i := 1 to 10 do begin

if i <> a then begin

if b = -1 then b := i

else if ar[i] > ar[b] then b := i;

end;

end;

c := -1;

for i := 1 to 10 do begin

if (i <> a) and (i <> b) then begin

if c = -1 then c := i

else if ar[i] > ar[c] then c := i;

end;

end;

writeln(a, ' ', b, ' ', c);

readkey;

end.

c := -1;

for i := 1 to 10 do begin

if (i <> a) and (i <> b) then begin

if c = -1 then c := i

else if ar[i] > ar[c] then c := i;

AR = (4, 5, 10, 5, 51, 33, 49, 64, 2, 7)

START

a := -1;

for i := 1 to 10 do begin

if a = -1 then a := i

else if ar[i] > ar[a] then a := i;

b := -1;

for i := 1 to 10 do begin

if i <> a then begin

if b = -1 then b := i

else if ar[i] > ar[b] then b := i;

OUTPUT (a,b,c)

END

**32.**

uses crt;

function meong(x: longint):integer;

begin

if (x = 0) then

meong := 0

else if (x mod 2 = 1) and ((x div 2) mod 2 = 1) then

meong := meong((x div 2) div 2) + 1

else

meong := meong(x + 1) + 1;

end;

begin

writeln(meong(888));

readkey;

end.

Meong ()

END

OUTPUT (meong(888))

meong

START

if (x = 0) then

meong := 0

else if (x mod 2 = 1) and ((x div 2) mod 2 = 1) then

meong := meong((x div 2) div 2) + 1

else

meong := meong(x + 1) + 1;

RETURN

**33.**

uses crt;

var

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

function get : integer;

var

m: integer;

left, right: integer;

begin

arr[1]:=303; arr[2]:=304; arr[3]:=365; arr[4]:=454; arr[5]:=487; arr[6]:=6; arr[7]:=12; arr[8]:=15; arr[9]:=78; arr[10]:=90; arr[11]:=155; arr[12]:=169; arr[13]:=183; arr[14]:=205; arr[15]:=209; arr[16]:=218; arr[17]:=5; arr[18]:=269; arr[19]:=282; arr[20]:=287;

if (arr[1] < arr[20]) then

get := 1

else begin

left := 1;

right := 20;

while (left < right) do

begin

m := (left+right) div 2;

if (arr[1] <= arr[m]) then

left := m+1

else

right := m;

end;

get := left;

end;

end;

begin

write(get);

readkey;

end.

START

ARR =(303, 304, 365, 454, 487, 6, 12, 15, 78, 90, 155, 169, 183, 205, 209, 218, 5, 269, 282, 287)

GET

GET ()

OUTPUT

if (arr[1] < arr[20]) then

get := 1

else begin

left := 1;

right := 20;

while (left < right) do

begin

m := (left+right) div 2;

if (arr[1] <= arr[m]) then

left := m+1

else

right := m;

end;

get := left;

end;

END

RETURN

**34.**

uses crt;

Var

ar : array[1..1000] of Boolean;

i,j : integer;

begin

for i := 1 to 1000 do

ar[i] := false;

for i := 1 to 1000 do begin

j := i;

while(j <= 1000) do begin

ar[j] := not(ar[j]);

j := j + i;

end;

end;

writeln(i);

writeln(j);

readln;

end.

START

begin

for i := 1 to 1000 do

ar[i] := false;

for i := 1 to 1000 do begin

j := i;

while(j <= 1000) do begin

ar[j] := not(ar[j]);

j := j + i;

end;

end;

END

**35.**

var

isi : array[1..10] of integer ;

function X(l, r, v : integer) : longint;

var

temp : integer;

begin

isi[1]:=-4; isi[2]:=1; isi[3]:=7; isi[4]:=9; isi[5]:=0;

isi[6]:=1; isi[7]:=2; isi[8]:=4; isi[9]:=3; isi[10]:=-1;

if(l = r) then X := isi[l]

else begin

temp := (l + r) div 2;

if(isi[temp] > v) then begin

X := X(l, temp, v);

end else

X := X(temp + 1, r, v);

end;

end;

var

hasil,a,b,c : integer;

begin

read(a);

read(b);

read(c);

hasil:=x(a,b,c);

write(hasil);

readln;

readln;

end.

**36.**

var

isi : array[1..10] of integer ;

function X(l, r, v : integer) : longint;

var

temp : integer;

begin

isi[1]:=-4; isi[2]:=1; isi[3]:=7; isi[4]:=9; isi[5]:=0; isi[6]:=1; isi[7]:=2; isi[8]:=4; isi[9]:=3; isi[10]:=-1;

if(l = r) then X := isi[l]

else begin

temp := (l + r) div 2;

if(isi[temp] > v) then begin

X := X(l, temp, v);

end else

X := X(temp + 1, r, v);

end;

end;

var

hasil,a,b,c : integer;

begin

read(a);

read(b);

read(c);

hasil:=x(a,b,c);

write(hasil);

readln;

readln;

end.

**35-36.**

X (l,r,v)

ISI = (-4,1,7,9,0,1,2,4,3,-1)

INPUT (x)

START

begin

if(l = r) then X := isi[l]

else begin

temp := (l + r) div 2;

if(isi[temp] > v) then begin

X := X(l, temp, v);

end else

X := X(temp + 1, r, v);

end;

X

Output (x)

RETURN

END

**37-38.**

uses crt;

var

daebak : array [0..2015] of char;

w, x, y, z, i : integer;

procedure saranghae(x : integer; len : integer);

var

i : integer;

store : char;

begin

store := daebak[x];

for i := len-1 downto 0 do begin

if (i = 0) then

daebak[x+((i+1) mod len)] := store

else

daebak[x+((i+1) mod len)] := daebak[x+i];

end;

end;

procedure anyeong(arr\_sz : integer; part\_sz : integer);

var

i : integer;

begin

for i:= 0 to (arr\_sz div part\_sz)-1 do

begin

saranghae(i \* part\_sz, part\_sz);

end;

end;

begin

readln(w, x, y, z);

oppa(w);

for i := 1 to x do

anyeong(y, z);

for i := 0 to w-1 do

begin

if(i mod 4 = 0) and (i <> 0) then write(' ');

write(daebak[i]);

end;

writeln;

readkey;

end.

saranghae

END

OUTPUT

oppa(w);

for i := 1 to x do

anyeong(y, z);

for i := 0 to w-1 do

begin

if(i mod 4 = 0) and (i <> 0) then write(“ “);

write(daebak[i]);

INPUT (w,x,y,z)

START

oppa

anyeong

saranghae

var

i : integer;

store : char;

begin

store := daebak[x];

for i := len-1 downto 0 do begin

if (i = 0) then

daebak[x+((i+1) mod len)] := store

else

daebak[x+((i+1) mod len)] := daebak[x+i];

end;

end;

RETURN

anyeong

var

i : integer;

begin

for i:= 0 to (arr\_sz div part\_sz)-1 do

begin

saranghae(i \* part\_sz, part\_sz);

end;

end;

RETURN

**.**

oppa

var

i : integer;

begin

for i := 0 to n-1 do begin

if(i mod 4 = 0) then daebak[i] := 'T'

else if(i mod 4 = 1) then daebak[i] := 'O'

else if(i mod 4 = 2) then daebak[i] := 'K'

else daebak[i] := 'I';

end;

end;

RETURN

**39-40.**

uses crt;

Var

s, t: string;

x, l, r, k: byte;

temp: char;

begin

readln(s);

k := 0;

x := 0;

l := 1;

r := length(s);

t := '';

while (l <= r) do

begin

if (k mod 2) = 0 then

begin

temp:=s[r];

r := r - 1;

end

else

begin

temp := s[l];

l := l + 1;

end;

k := k + 1;

x := (x + ord(temp) - ord('A')) mod 26;

t := t + chr(x + ord('A'));

end;

writeln(t);

readkey;

end.

**.**

START

INPUT (s)

k := 0;

x := 0;

l := 1;

r := length(s);

t := '';

while (l <= r) do

begin

if (k mod 2) = 0 then

begin

temp := s[r];

r := r – 1;

end

else

begin

temp := s[l];

l := l + 1;

end;

k := k + 1;

x := (x + ord(temp) - ord('A')) mod 26;

t := t + chr(x + ord('A'));

OUTPUT (t)

END

**41-42.**

uses crt;

Var

Q : array[1..15] of integer ;

P : array[0..15] of integer;

i, a, b,total : integer;

begin

Q[1]:=1; Q[2]:=5; Q[3]:=2; Q[4]:=3; Q[5]:=4; Q[6]:=6; Q[7]:=2; Q[8]:=7; Q[9]:=3; Q[10]:=4; Q[11]:=6; Q[12]:=0; Q[13]:=-1; Q[14]:=2; Q[15]:=3;

P[0] := 0;

for i := 1 to 15 do begin

P[i] := P[i-1] + Q[i];

end;

readln (a,b);

total := 0; // baris-1

for i := a to b do begin // baris-2

total := total + Q[i]; // baris-3

end; // baris-4

writeln(total);

readkey;

end.

START

Q : array[1..15] of integer = (1,5,2,3,4,6,2,7,3,4,6,0,-1,2,3);

P : array[0..15] of integer;

i, a, b,total : integer;

P[0] := 0;

for i := 1 to 15 do begin

P[i] := P[i-1] + Q[i];

end;

INPUT (a,b)

total := 0;

for i := a to b do begin

total := total + Q[i];

end;

OUTPUT (total)

END

**43.**

uses crt;

var

isi : array[1..10] of integer;

i : integer;

procedure Whatsup(l, r : integer);

var

X : integer;

begin

X := isi[l];

isi[l] := isi[r];

isi[r] := X;

end;

procedure naoooon(l, r : integer);

var

ini : integer;

kiri, kanan : integer;

begin

if(l < r) then begin

kiri := l;

kanan := r;

ini := isi[(kiri + kanan) div 2];

while(kiri < kanan) do begin

while(isi[kiri] > ini) do kiri := kiri + 1;

while(isi[kanan] < ini) do kanan := kanan - 1;

if(kiri < kanan) then Whatsup(kiri, kanan);

end;

naoooon(l, kanan);

naoooon(kanan+1, r);

end;

end;

begin

isi[1] := 5; isi[2] := 10; isi[3] := 18; isi[4] := 1; isi[5] := 7; isi[6] := 9; isi[7] := 3; isi[8] := 8; isi[9] := 100; isi[10] := 29;

naoooon(1,10);

for i := 1 to 9 do

write(isi[i], ' ');

writeln(isi[10]);

readln;

end.

**44.**

uses crt;

var

isi : array[1..10] of integer;

i : integer;

procedure Whatsup(l, r : integer);

var

X : integer;

begin

X := isi[l];

isi[l] := isi[r];

isi[r] := X;

end;

procedure naoooon(l, r : integer);

var

ini : integer;

kiri, kanan : integer;

begin

if(l < r) then begin

kiri := l;

kanan := r;

ini := isi[(kiri + kanan) div 2];

while(kiri < kanan) do begin

while(isi[kiri] > ini) do kiri := kiri + 1;

while(isi[kanan] < ini) do kanan := kanan - 1;

if(kiri < kanan) then Whatsup(kiri, kanan);

end;

naoooon(l, kanan);

naoooon(kanan+1, r);

end;

end;

begin

isi[1] := 5; isi[2] := 10; isi[3] := 18; isi[4] := 1; isi[5] := 7;

isi[6] := 9; isi[7] := 3; isi[8] := 8; isi[9] := 100; isi[10] := 29;

naoooon(3,6);

for i := 1 to 9 do

write(isi[i], ' ');

writeln(isi[10]);

readln;

end.

**43-44.**

Whatsup

START

var

X : integer;

begin

X := isi[l];

isi[l] := isi[r];

isi[r] := X;

end;

Whatsup

naooooon

RETURN

isi[1] := 5;

isi[2] := 10;

isi[3] := 18;

isi[4] := 1;

isi[5] := 7;

isi[6] := 9;

isi[7] := 3;

isi[8] := 8;

isi[9] := 100;

isi[10] := 29;

naooooon

var

ini : integer;

kiri, kanan : integer;

begin

if(l < r) then begin

kiri := l;

kanan := r;

ini := isi[(kiri + kanan) div 2];

while(kiri < kanan) do begin

while(isi[kiri] > ini) do kiri := kiri + 1;

while(isi[kanan] < ini) do kanan := kanan - 1;

if(kiri < kanan) then Whatsup(kiri, kanan);

end;

naoooon(l, kanan);

naoooon(kanan+1, r);

end;

end;

naoooon(1,10)

for i := 1 to 9 do

OUTPUT isi[i]

END

RETURN

**45.**

uses crt;

Var

ar : array[1..5] of integer;

i : integer;

procedure S(a,b : integer);

var

temp : integer;

begin

temp := ar[a];

ar[a] := ar[b];

ar[b] := temp;

end;

begin

for i := 1 to 5 do read(ar[i]);

S(3,4);

S(4,1);

S(5,2);

S(5,1);

for i := 1 to 4 do begin

write(ar[i], ' ');

end;

writeln(ar[5]);

readkey;

end.

**.**

START

S (a,b)

var

temp : integer;

begin

temp := ar[a];

ar[a] := ar[b];

ar[b] := temp;

end;

S

for i := 1 to 5 do

INPUT (ar[i])

S(3,4);

S(4,1);

S(5,2);

S(5,1);

RETURN

for i := 1 to 4 do

OUTPUT (ar[i]

END

**46.**

uses crt;

var

sum, i, j, n, c : integer;

begin

readln(n);

sum:=0;

for i:=2 to n do

begin

c:=0;

j:=i;

while (j>0) do

begin

if (j mod 2 = 1) then c:=c+1;

j:=j div 2;

end;

if (c=1) then sum:=sum+1;

end;

writeln(sum);

readkey;

end.

START

INPUT

SUM = 0

for i:=2 to n do begin

c:=0; j:=i;

while (j>0) do begin

if (j mod 2 = 1) then c:=c+1; j:=j div 2;

end;

C =1

SUM = SUM + 1

OUTPUT

END

**47.**

uses crt;

var

n,m : longint;

jml : int64;

begin

read(n,m);

while (n>0) or (m>0) do

begin

jml:=jml+n\*m;

dec(n);dec(m);

end;

writeln(jml);

readkey;

end.

OUTPUT (jml)

dec(n);dec(m);

jml:=jml+n\*m;

INPUT (n.m)

START

while (n>0) or (m>0) do

END

**48.**

uses crt;

var

n,i : integer;

s,k : string;

begin

k:='';

readln(s);

n:=length(s);

for i:=n downto 1 do

begin

k:=k+s[i];

end;

if s=k then writeln('PALINDROM')

else writeln('Bukan PALINDROM');

readkey;

end.

START

END

BUKAN PALINDROM

PALINDROM

S=K

n:=length(s);

for i:=n downto 1 do

begin

k:=k+s[i];

end;

INPUT

**false**