program grey;

uses

strings;

var

{LOG(10^18; 2)= 59.79 < 60 отже max n=10^18-1 < 2^60 отже 60 позицій досить}

dano : string[19];

dano2, zsuv : string[61];

rez : string[61];

f : text;

{зсув в право на­  1}

function str\_shr (s:string) : string;

var

s2 : string;

begin

s2 := copy('0'+s, 1, length(s));

str\_shr:= s2;

end;

{ XOR для рядків 2-ї системи}

function str\_xor (s1:string; s2:string) : string;

var

m1, m2, m3, min1, max1, j, x, y : shortint;

s3 : string;

begin

m1 := length(s1);

m2 := length(s2);

max1 := m1;

min1 := m2;

s3 := s1;

if m2>max1 then

begin

max1 := m2;

s3 := s2;

min1 := m1;

end;

m3 := length(s3);

for j:=1 to min1 do

begin

x := ord(s1[m1+1-j])-48;

y := ord(s2[m2+1-j])-48;

s3[m3+1-j] := chr((x xor y)+48);

end;

str\_xor:= s3;

end;

{перевод 2-чну систему}

function str\_binary (s:string) : string;

var

m1, j, j1, x, y : shortint;

s1, s2, s3 : string;

begin

s1 := s;

m1 := length (s1);

s2:='';

while m1>0 do

begin

x := 0;

j1 := 0;

if ord(s1[1])-48<2 then

begin

x:=ord(s1[1])-48;

j1:=1;

end;

s3:='';

for j := 1+j1 to m1 do

begin

x := x\*10+ord(s1[j])-48;

y:=x div 2;

s3:=s3+chr(y+48);

x:=x mod 2;

end;

Insert(chr(x+48), s2, 1);

s1 := s3;

m1 := length (s1);

end;

str\_binary:= s2;

end;

{основна програма }

begin

{дано}

assign(f, 'grey.dat');

reset(f);

readln(f, dano);

close(f);

{Перевод в код Грея}

{ аналог числової формули Грей (b) =b xor (b shr 1); }

dano2 := str\_binary(dano);

zsuv := str\_shr(dano2);

rez := str\_xor(dano2, zsuv);

{Результат}

assign(f, 'grey.sol');

rewrite(f);

writeln(f, rez);

close(f);

{writeln('B файл grey.sol записано результат. Натисніть Enter.');

readln;}

end.