**Program** Lab\_12;

**uses** IniFini, VvodVivod, Conditional;

**var**

a:matrix;

ca:mas;

srA:real;

n:integer;

flag:boolean;

**begin**

read(n);

**if** (n<5) **or** (n>10) **then**

writeln('Error');

**else**

**begin**

VvodMatrix(a, n, Inp);

VvodMatrix(сa, n, Inp);

flag:=ProvMat(a, n);

**if not**(flag) **then**

write(Outp, 'Неккоретно введены данные матрицы!')

**else begin**

flag:=ProvMass(ca, n);

**if not**(flag) **then**

write(Outp, 'Неккоретно введены данные массива!')

**else begin**

srA:=fsrX(a, n);

flag:=check(srA, ca, n);

writeln(Outp, 'Вывод конечных данных:');

**if** flag **then**

**begin**

izm1(a, n, srA);

izm2(a, n, srA);

VivodMatrix(a, n, Outp);

**end**

**else**

**begin**

writeln(Outp, 'Матрица А не изменена');

VivodMatrix(a, n, Outp);

**end**;

**end**;

**end**;

**end**;

**end**.

**Unit** IniFini;

**interface**

**type**

matrix=**array**[1..10,1..10] **of** real;

mas=**array**[1..10] **of** real;

bool=**array**[1..10] **of** boolean;

**var**

Inp, Outp:text;

name:string;

**implementation**

**initialization**

write('Введтите имя файла для считывания: ');

readln(name);

assign(Inp, name+'.txt');

reset(Inp);

assign(Outp,'Out.txt');

rewrite(Outp);

**finalization**

close(Inp);

close(Outp);

**end**.

**Unit** VvodVivod;

**interface**

**uses** IniFini;

//function reading(nx:integer):integer;

**Procedure** VvodMatrix(**var** x:matrix; **var** nx:integer; **var** fx1:text);

**Procedure** VvodMas(**var** xc:mas; **var** nx:integer; **var** fx1:text);

**Procedure** VivodMas(xc:mas; nx:integer; **var** fx2:text);

**Procedure** VivodMatrix(x:matrix; nx:integer; **var** fx2:text);

**implementation**

{function reading(nx:integer):integer;

var

fl:boolean;

begin

fl:=true;

while fl do

begin

writeln('Введите размерность матрицы от 5 до 100 (5)');

read(nx);

if (nx<101) and (nx>4) then

fl:=false;

end;

reading:=nx;

end;}

**Procedure** VvodMatrix(**var** x:matrix; **var** nx:integer; **var** fx1:text); **var**

i, j:integer;

**begin**

//nx:=reading(nx);

**for** i:=1 **to** nx **do**

**for** j:=1 **to** nx **do**

read(fx1, x[i,j]);

**end**;

**Procedure** VvodMas(**var** xc:mas; **var** nx:integer; **var** fx1:text);

i:integer;

**begin**

**for** i:=1 **to** nx **do**

read(fx1, xc[i]);

**end**;

**Procedure** VivodMas(xc:mas; nx:integer; **var** fx2:text);

**var**

i: integer;

**begin**

write(fx2, nx, #13#10, #13#10);

**for** i:=1 **to** nx **do**

write(fx2, xc[i]:5,' ');

**end**;

**Procedure** VivodMatrix(x:matrix; nx:integer; **var** fx2:text);

**var**

i, j:integer;

**begin**

writeln(fx2, #13#10);

**for** i:=1 **to** nx **do**

**begin**

**for** j:=1 **to** nx **do**

write (fx2, x[i,j]:5, ' ');

writeln(fx2);

**end**;

**end**;

**end**.

**Unit** conditional;

**interface**

**uses** IniFini;

**function** ProvMat(x:matrix; nx:integer):boolean;

**function** ProvMass(xc:mas; nx:integer):boolean;

**function** SrArif(x:matrix; nx:integer):real;

**function** check(srX:real; xc:mas; nx:integer):boolean;

**procedure** izm(**var** x:matrix; nx:integer; srX:real);

**implementation**

**function** ProvMat(x:matrix; nx:integer):boolean;

**var**

i, j:integer;

fl:boolean;

**begin**

fl:=true;

i:=1;

j:=1;

**while** (i<=nx)**and**(fl) **do begin**

**while** (j<=nx)**and**(fl) **do begin**

**if** abs(x[i,j])>100 **then**

fl:=false

**else**

j:=j+1;

**end**;

i:=i+1;

**end**;

ProvMat:=fl;

**end**;

**function** ProvMass(xc:mas; nx:integer):boolean;

**var**

i:integer;

fl:boolean;

**begin**

fl:=true;

i:=1;

**while** (i<=nx)**and**(fl) **do begin**

**if** abs(xc[i])>100 **then**

fl:=false

**else**

i:=i+1;

**end**;

ProvMass:=fl;

**end**;

**function** SrArif(x:matrix; nx:integer):real;

**var**

i:integer;

l:real;

**begin**

**for** i:=1 **to** nx **do**

l:=l+x[i,i];

fsrX:=l/nx;

**end**;

**function** check(srX:real; xc:mas; nx:integer):boolean;

**var**

i:integer;

fl:boolean;

**begin**

fl:=true;

i:=1;

**while** (i<=nx) **and** fl **do**

**if** (srX>=xc[i]) **then**

fl:=false

**else**

i:=i+1;

check:=fl;

**end**;

**procedure** izm1(**var** x:matrix; nx:integer; srX:real);

**var**

i, j:integer;

**begin**

**for** i:=1 **to** nx **do**

**for** j:=1 **to** nx **do**

**if** x[i,j]>0 **then**

x[i,j]:=X[i,j]+srX;

**end**;

**procedure** izm2(**var** x:matrix; nx:integer; srX:real);

**var**

i, j:integer;

**begin**

**for** i:=1 **to** nx **do**

**for** j:=1 **to** nx **do**

**if** x[i,j]<0 **then**

x[i,j]:=x[i,j]-srX;

**end;**

**end**.