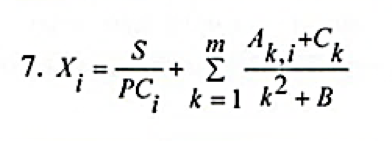
Задание №1 (C++)

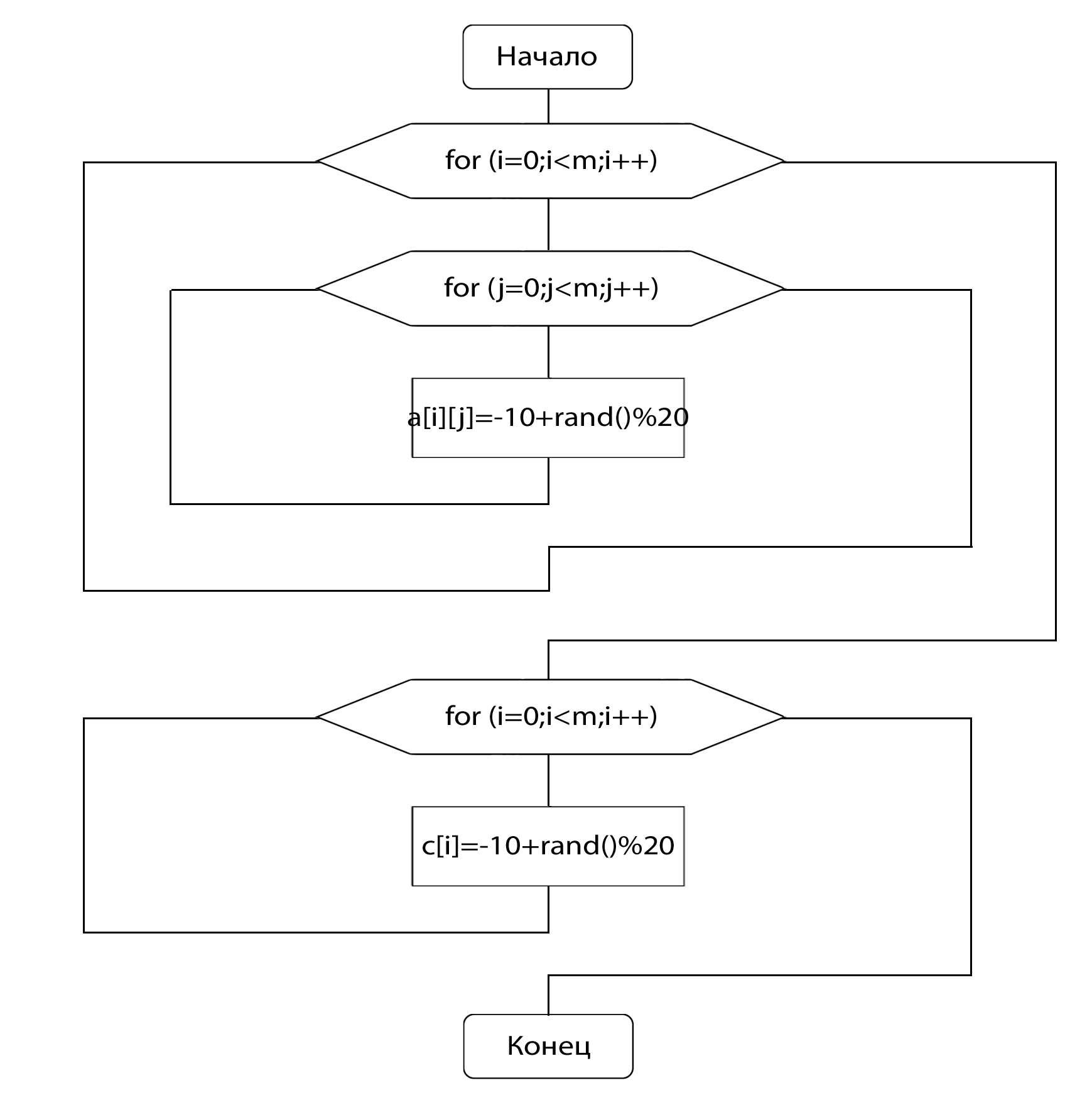
Условие:

Дана переменная В и матрица А, состоящая из m строк и m столбцов. Элементы С образуют одномерный массив из m элементов.

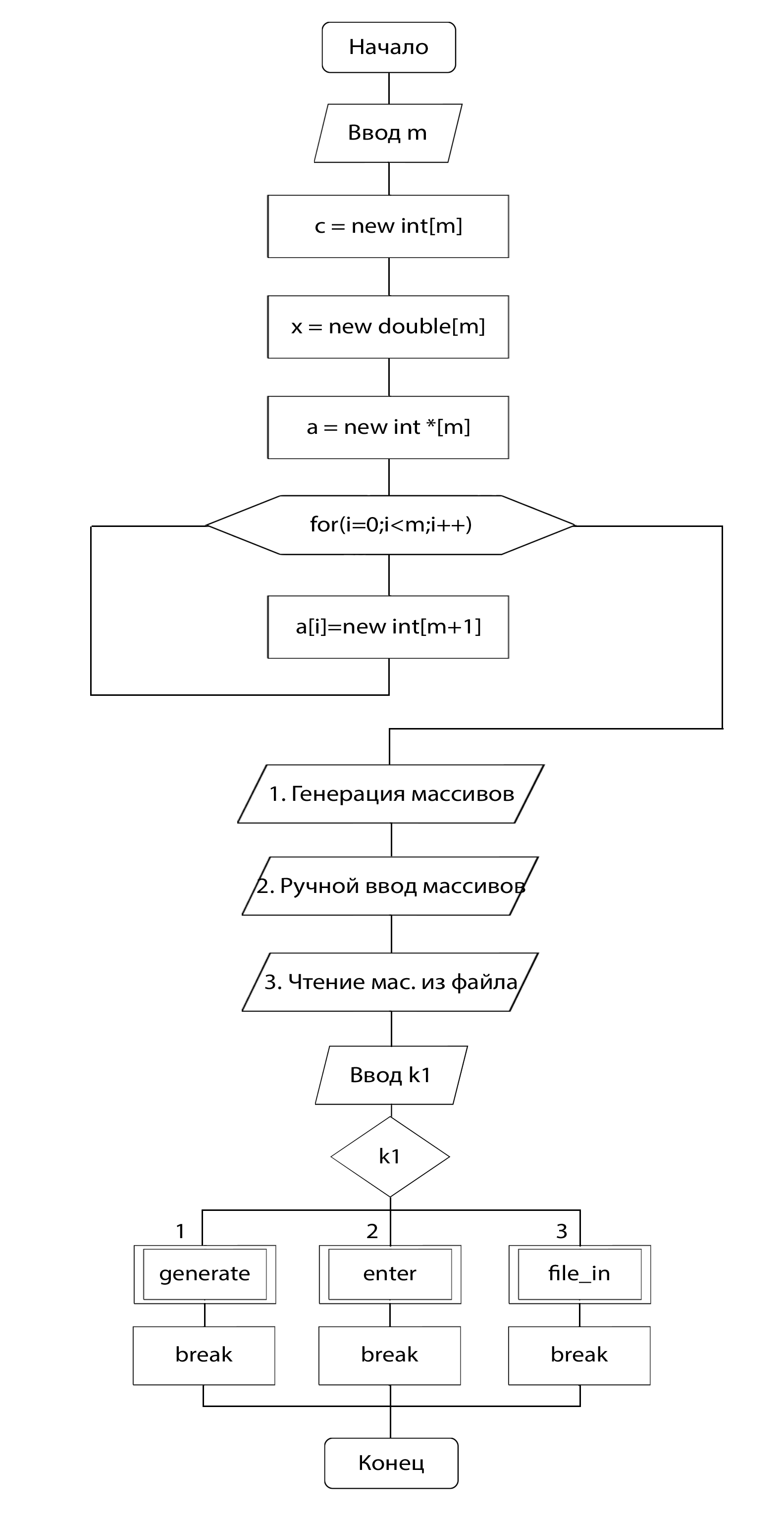
Требуется составить алгоритм и программу вычисления значений элементов одномерного массива Х1,Х2,..., X m (когда вычисляется Хi, Р равно Х i -1, для первого элемента массива Р=1).

S — сумма элементов главной диагонали матрицы А, вычисляемая в алгоритме. 

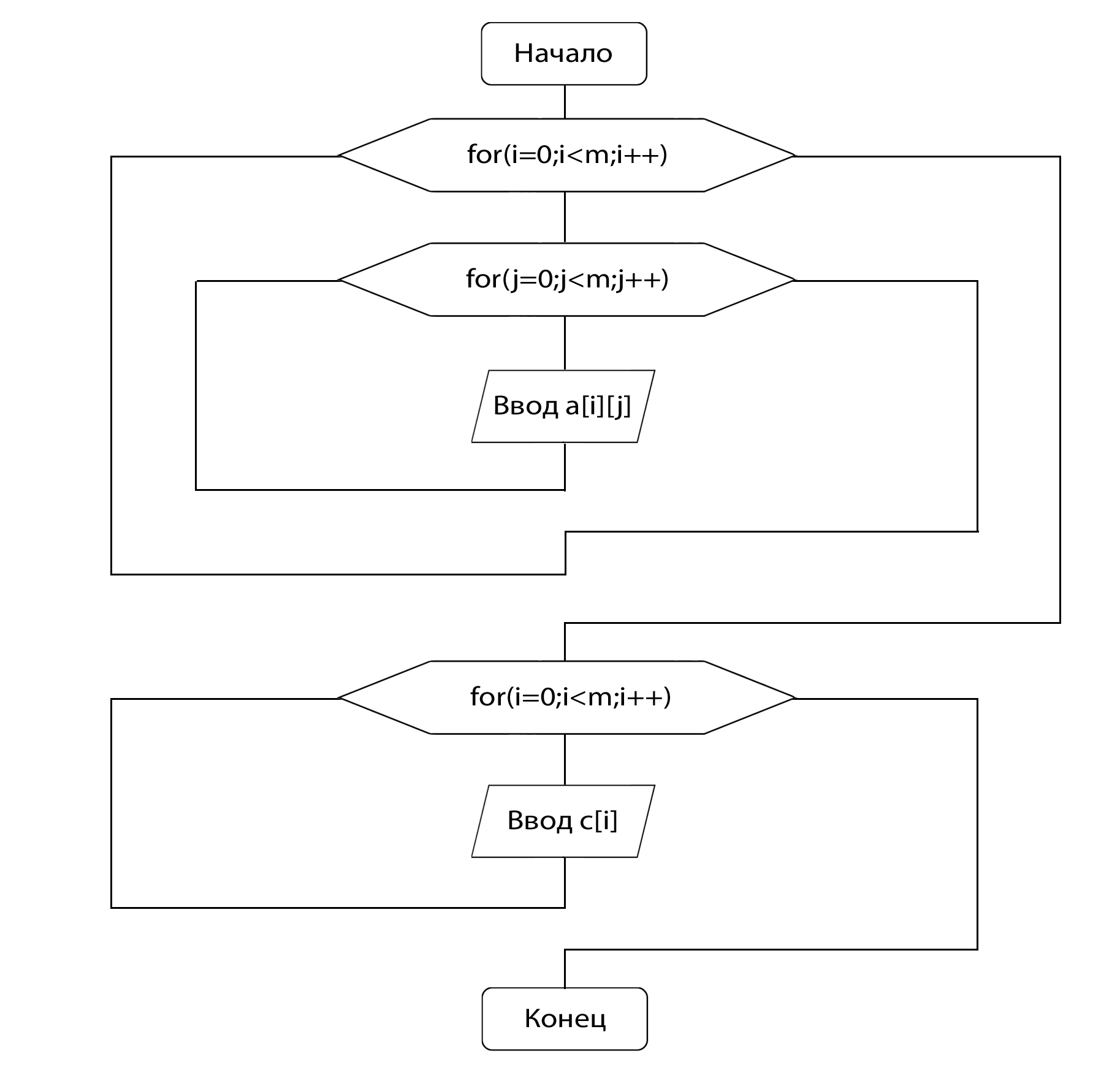
Алгоритм процедуры generate:



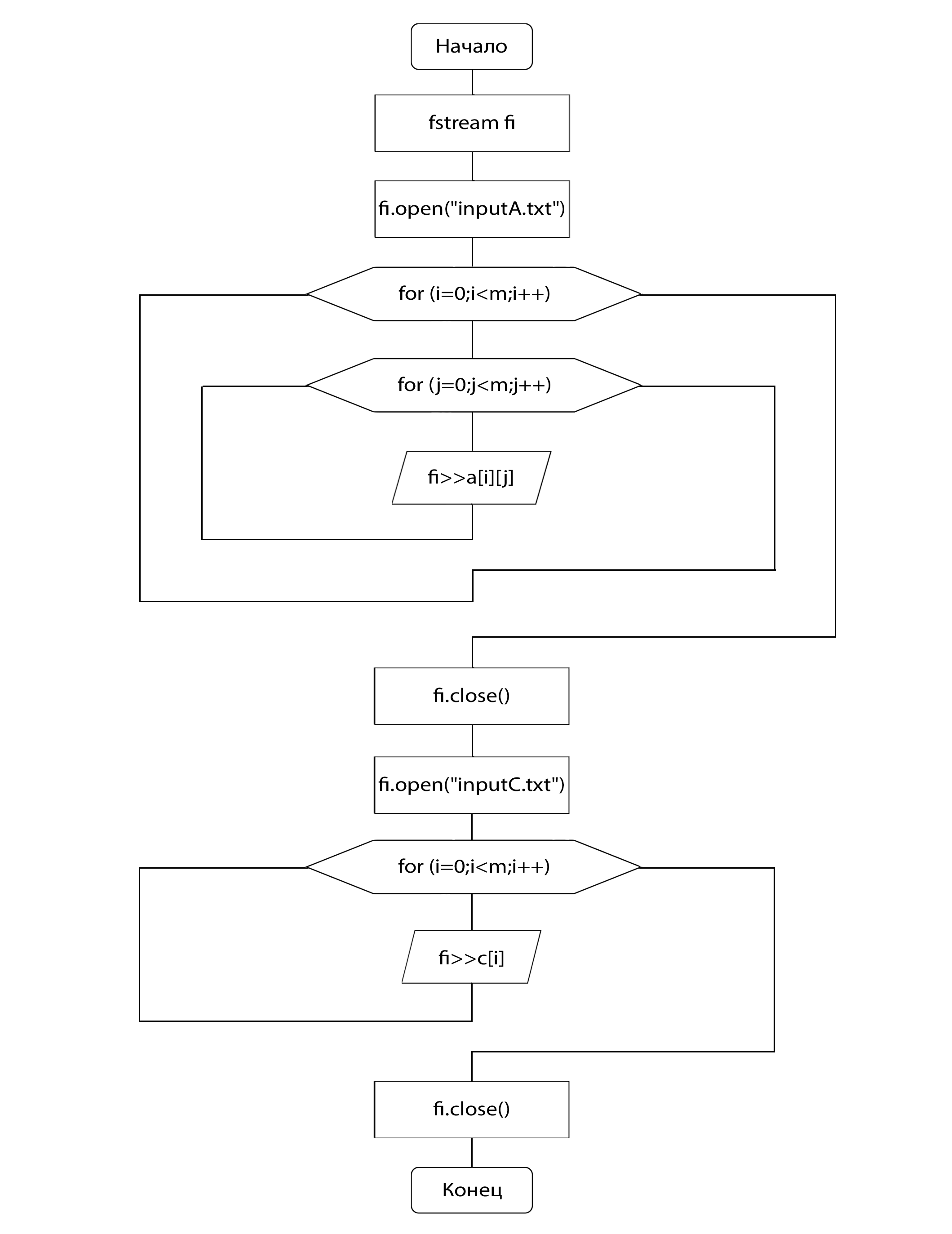
Алгоритм процедуры initarray:



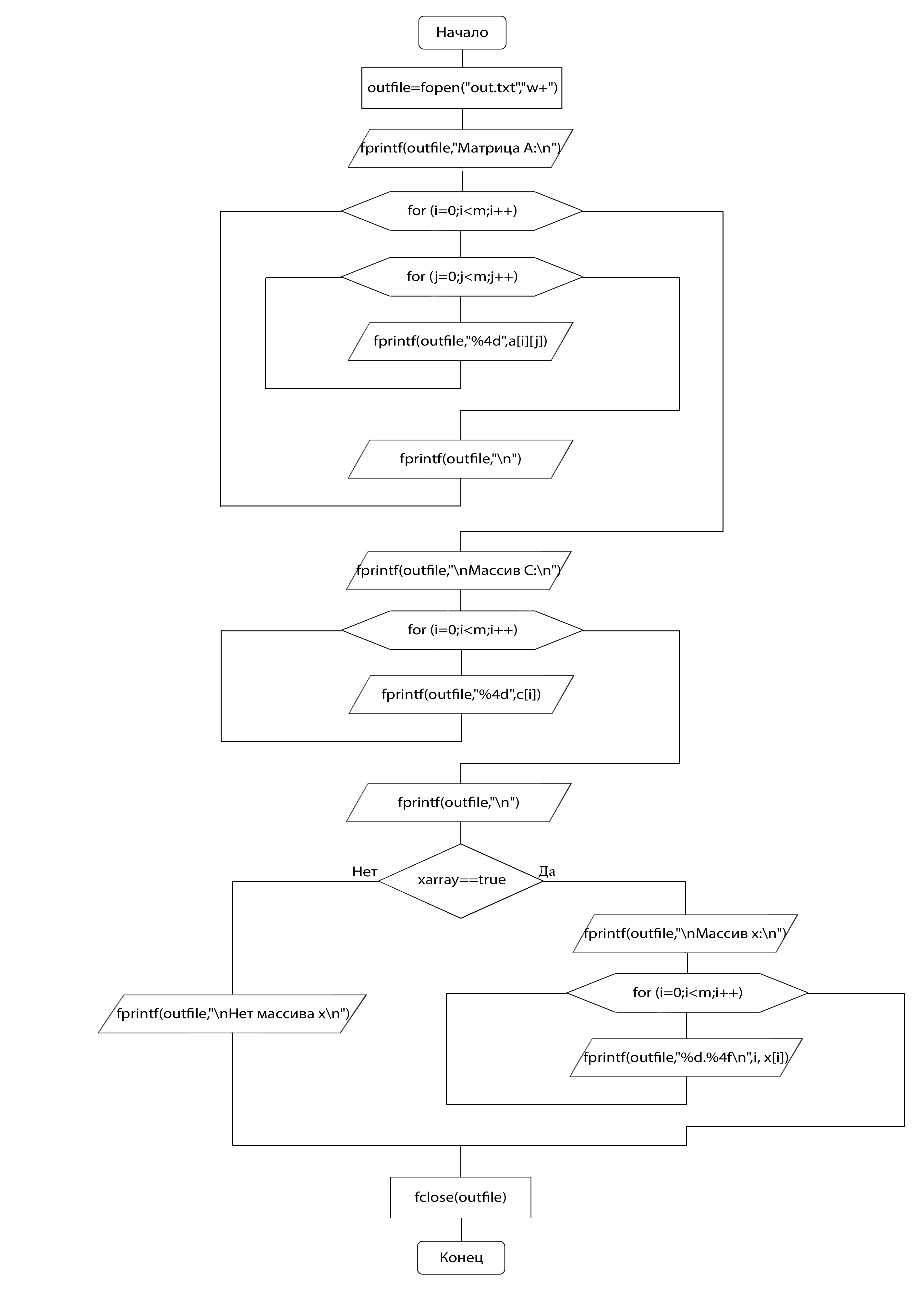
Алгоритм процедуры enter:



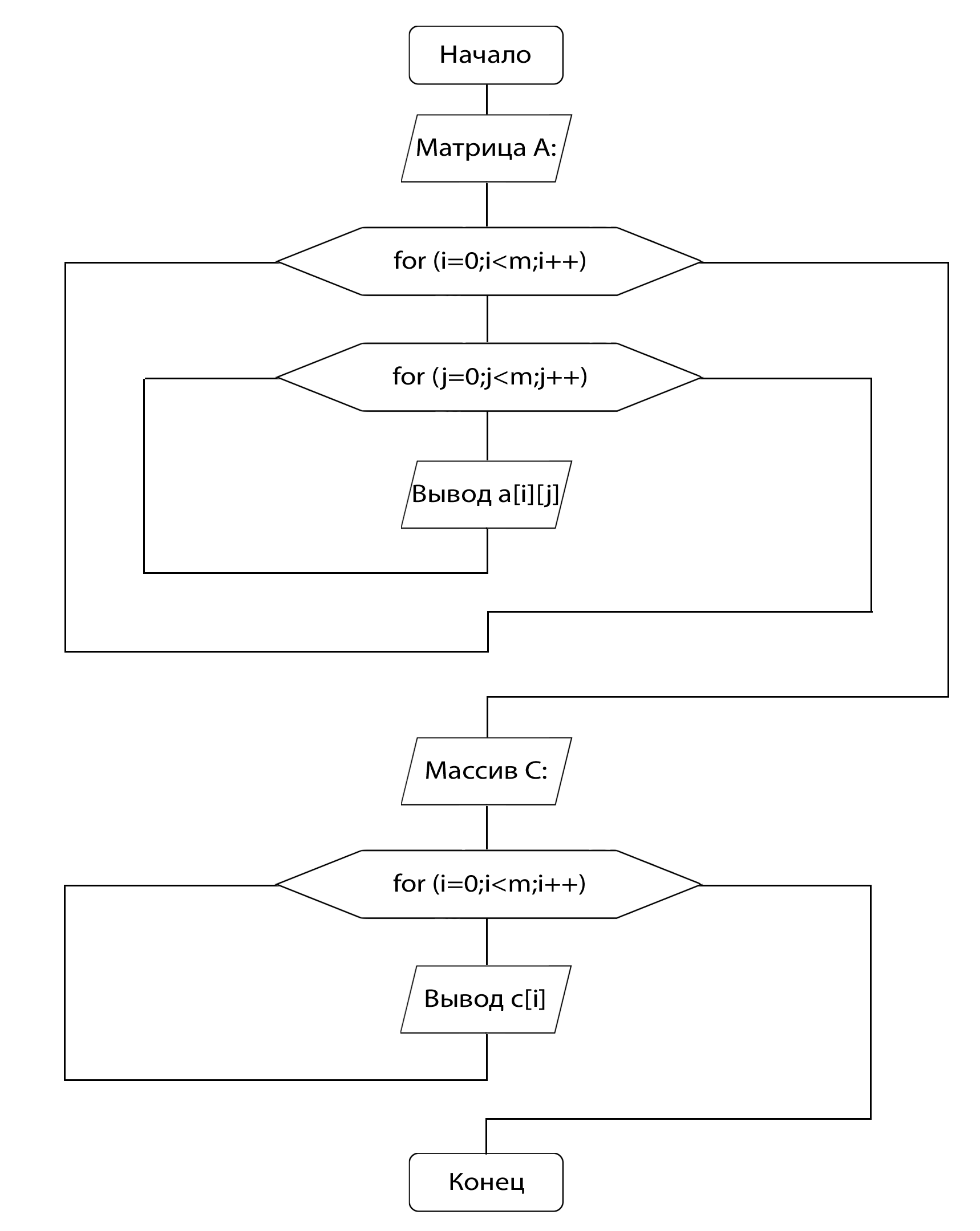
Алгоритм процедуры file\_in:



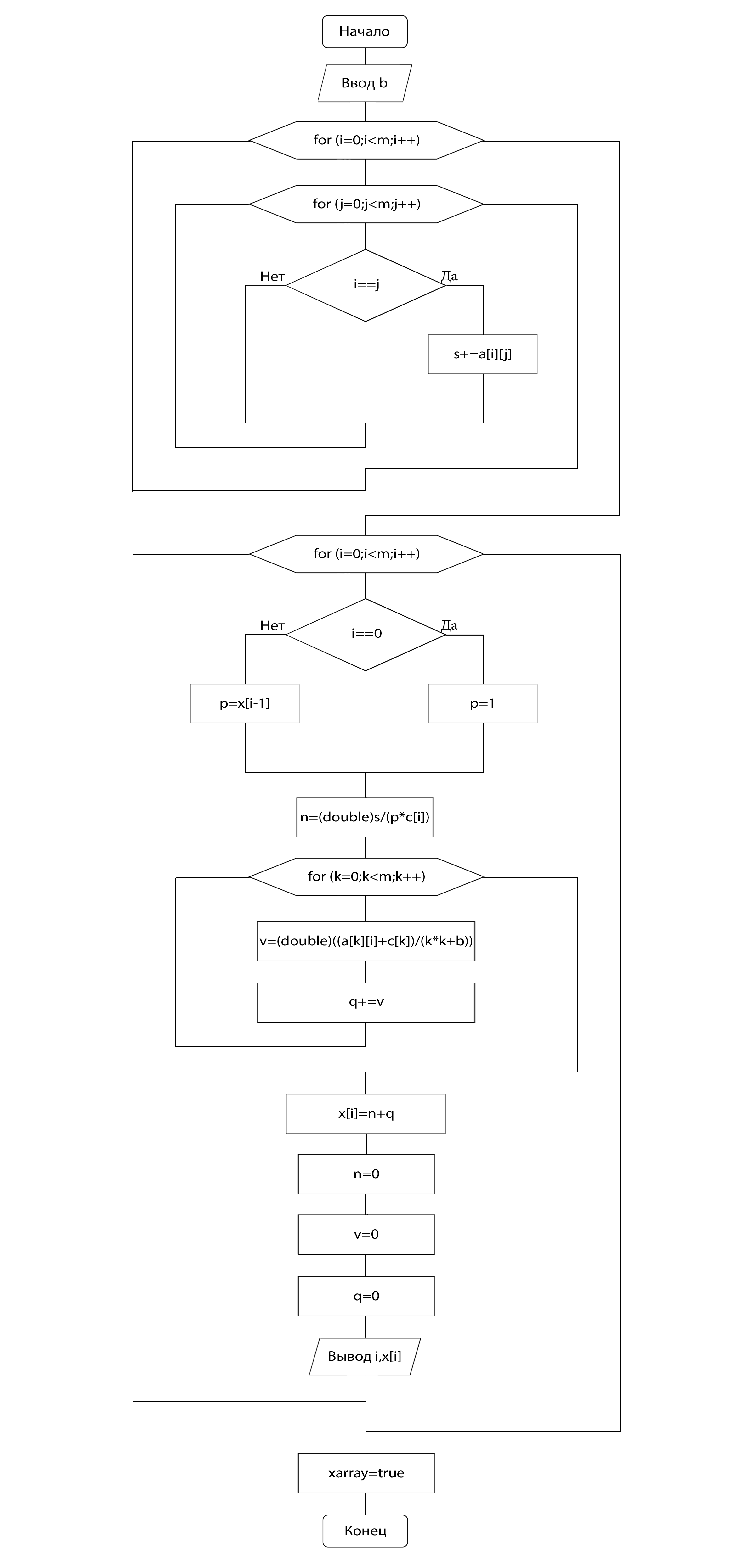
Алгоритм процедуры file\_out:



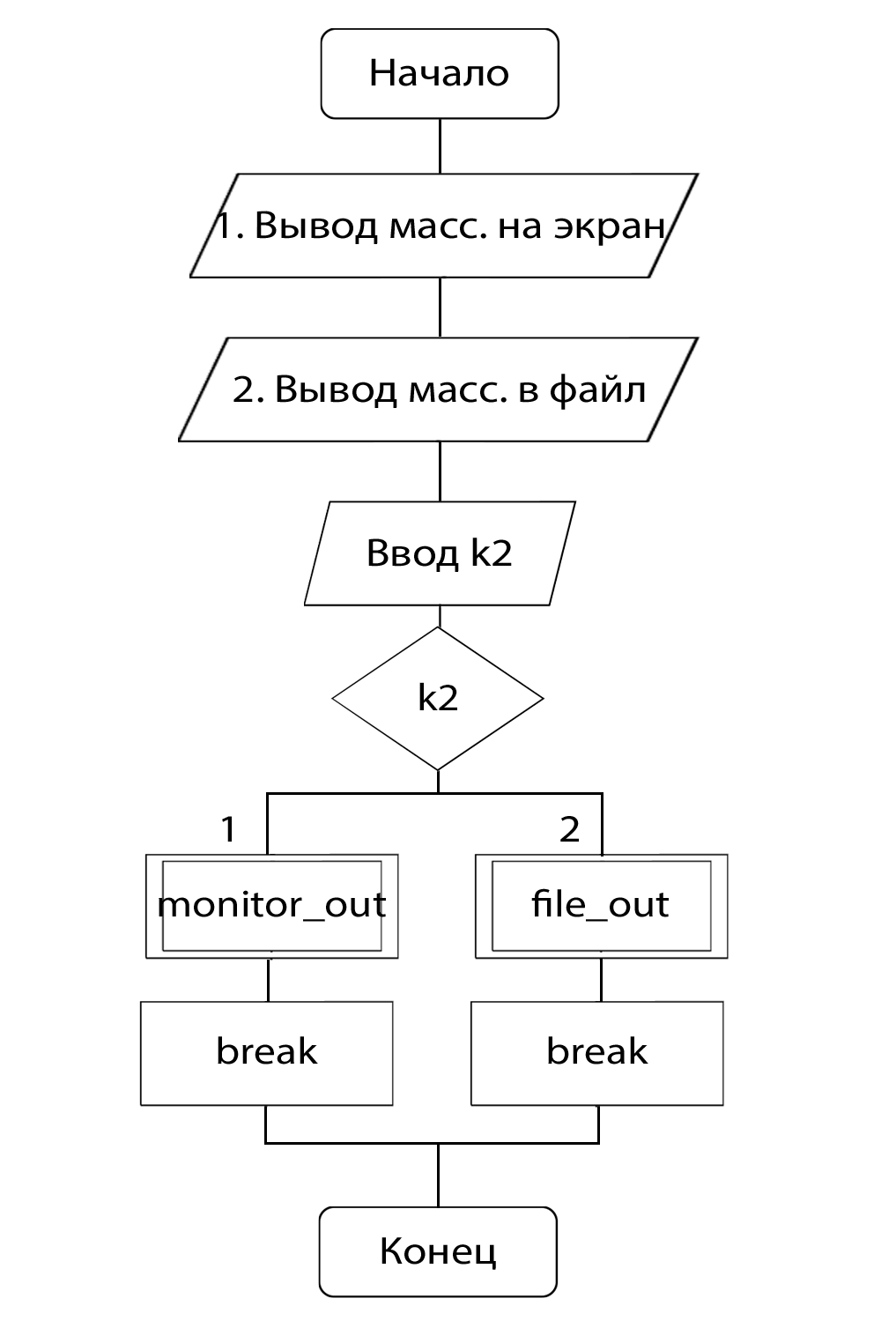
Алгоритм процедуры monitor\_out:



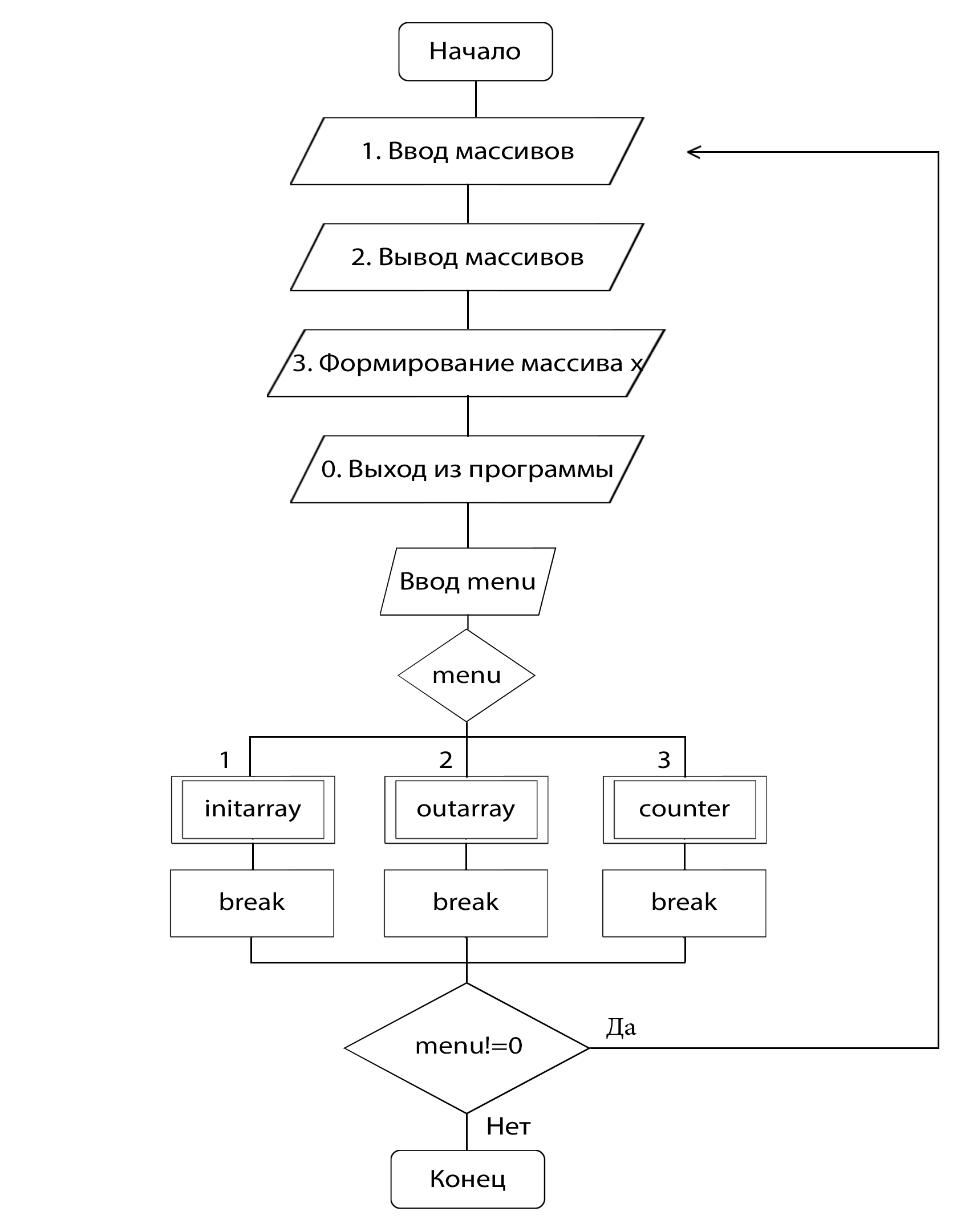
Алгоритм процедуры counter:



Алгоритм процедуры outarray:



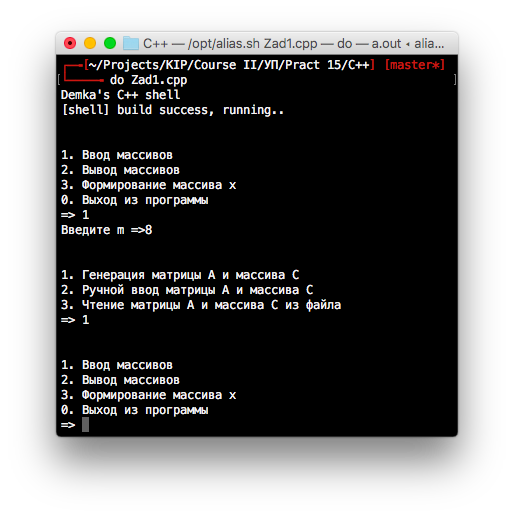
Алгоритм основной программы:



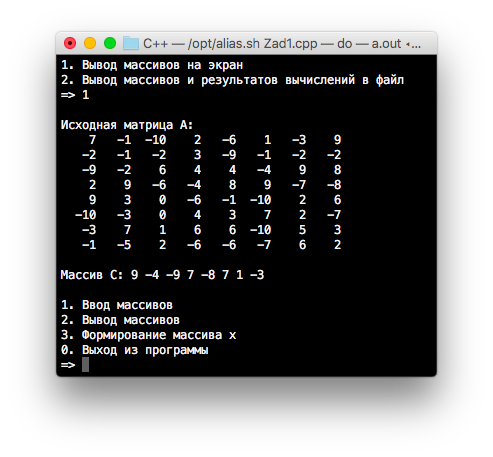
Исходный код программы:

#include **<iostream>**#include **<iomanip>**#include **<fstream>  
  
using namespace** std;  
  
**int** i, j, t, s, m, b, k, k1, k2, menu, \*c, \*\*a;  
**double** p, q, n, v, \*x;  
**bool** xarray = **false**;  
FILE \*outfile;  
  
*//Генерация массивов***void** generate(){  
  
 srand((**unsigned**)time(0));  
 **for** (i=0;i<m;i++)  
 **for** (j=0;j<m;j++)  
 a[i][j]=-10+rand()%20;  
  
 **for** (i=0;i<m;i++)  
 c[i]=-10+rand()%20;  
  
}  
  
*//Ручный ввод массивов***void** enter(){  
  
 cout<<**"<Заполняем матрицу А>\n"**;  
 **for**(i=0;i<m;i++)  
 **for**(j=0;j<m;j++){  
 cout<<**"Введите элемент ["**<<i<<**"]["**<<j<<**"]: "**;  
 cin>>a[i][j];  
 }  
  
 cout<<**"\n\n<Заполняем массив С>\n"**;  
 **for**(i=0;i<m;i++){  
 cout<<**"Введите элемент ["**<<i<<**"]: "**;  
 cin>>c[i];  
 }  
  
}  
  
*//Ввод массивов из файла***void** file\_in(){  
  
 fstream fi;  
 fi.open(**"inputA.txt"**);  
 **for** (i=0;i<m;i++)  
 **for** (j=0;j<m;j++)  
 fi>>a[i][j];  
 fi.close();  
  
  
 fi.open(**"inputC.txt"**);  
 **for** (i=0;i<m;i++)  
 fi>>c[i];  
 fi.close();  
  
}  
  
*//Вывод массивов на экран***void** monitor\_out(){  
  
 cout<<**"\nИсходная матрица А:"**;  
 **for** (i=0;i<m;i++){  
 cout<<**"\n"**;  
 **for** (j=0;j<m;j++)  
 cout<<setw(5)<<a[i][j];  
 }  
  
 cout << **"\n\nМассив С: "**;  
 **for** (i=0;i<m;i++)  
 cout<<c[i]<<**" "**;  
}  
  
*//Меню выбора ввода***void** initarray(){  
  
 cout<<**"Введите m =>"**; cin>>m;  
  
 c = **new int**[m];  
 x = **new double**[m];  
 a = **new int** \*[m];  
 **for**(i=0;i<m;i++)  
 a[i]=**new int**[m+1];  
  
  
 cout<<**"\n\n1. Генерация матрицы А и массива С\n2. Ручной ввод матрицы А и массива С\n3. Чтение матрицы А и массива С из файла\n=> "**;  
 cin>>k1;  
 **switch** (k1)  
 {  
 **case** 1:  
 generate();  
 **break**;  
  
 **case** 2:  
 enter();  
 **break**;  
  
 **case** 3:  
 file\_in();  
 **break**;  
 }  
  
}  
  
*//Процедура формирования массива x по образцу***void** counter(){  
  
 cout<<**"Введите переменную b =>"**;  
 cin>>b;  
  
 **for** (i=0;i<m;i++)  
 **for** (j=0;j<m;j++)  
 **if** (i==j)  
 s+=a[i][j];  
  
 cout<<**"\nМассив x на выходе:\n"**;  
  
 **for** (i=0;i<m;i++){  
 **if** (i==0)  
 p=1;  
 **else** p=x[i-1];  
 n=(**double**)s/(p\*c[i]);  
 **for** (k=0;k<m;k++){  
 v=(**double**)((a[k][i]+c[k])/(k\*k+b));  
 q+=v;  
 }  
  
 x[i]=n+q;  
 n=0;  
 v=0;  
 q=0;  
 cout<<i<<**". "**<<x[i]<<**"\n"**;  
 }  
  
 xarray=**true**;  
  
}  
  
*//Процедура вывода в файл***void** file\_out(){  
  
 outfile=fopen(**"out.txt"**,**"w+"**);  
 fprintf(outfile,**"Матрица А:\n"**);  
 **for** (i=0;i<m;i++){  
 **for** (j=0;j<m;j++)  
 fprintf(outfile,**"%4d"**,a[i][j]);  
 fprintf(outfile,**"\n"**);  
 }  
   
 fprintf(outfile,**"\nМассив C:\n"**);  
 **for** (i=0;i<m;i++)  
 fprintf(outfile,**"%4d"**,c[i]);  
 fprintf(outfile,**"\n"**);  
  
 **if** (xarray==**true**){  
 fprintf(outfile,**"\nСформированный массив x:\n"**);  
 **for** (i=0;i<m;i++)  
 fprintf(outfile,**"%d. %4f\n"**,i, x[i]);  
 }  
 **else** fprintf(outfile,**"\nМассив x не сформирован\n"**);  
   
 fclose(outfile);  
  
}  
  
*//Меню выбора вывода***void** outarray(){  
  
 cout<<**"\n\n1. Вывод массивов на экран\n2. Вывод массивов и результатов вычислений в файл\n=> "**;  
 cin>>k2;  
 **switch** (k2)  
 {  
 **case** 1:  
 monitor\_out();  
 **break**;  
  
 **case** 2:  
 file\_out();  
 **break**;  
 }  
  
}  
  
**int** main(){  
  
 **do** {  
 cout<<**"\n\n1. Ввод массивов\n2. Вывод массивов\n3. Формирование массива x\n0. Выход из программы\n=> "**;  
 cin>>menu;  
 **switch** (menu)  
 {  
 **case** 1:  
 initarray();  
 **break**;  
  
 **case** 2:  
 outarray();  
 **break**;  
  
 **case** 3:  
 counter();  
 **break**;  
  
 }  
  
 }  
 **while** (menu!=0);  
  
cout<<**"\n"**;  
**return** 0;  
}

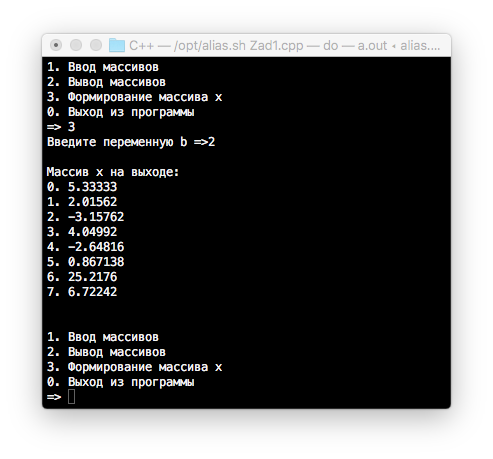
Скриншоты программы:

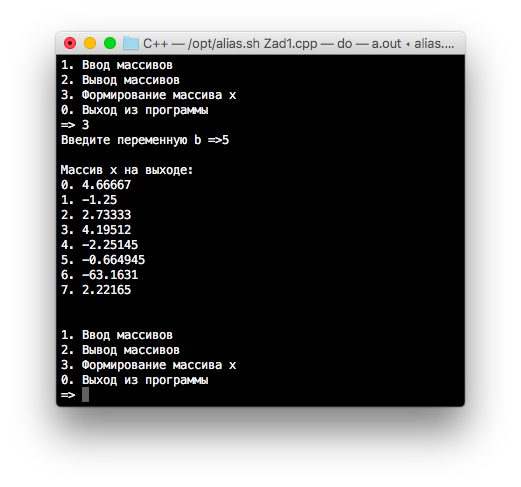
а) Ввод исходных массивов через генератор случайных чисел

б) Вывод массивов на экран

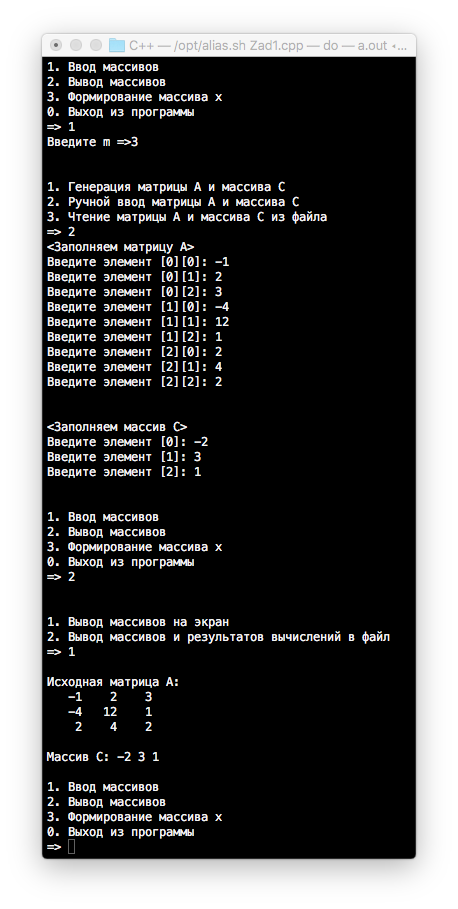


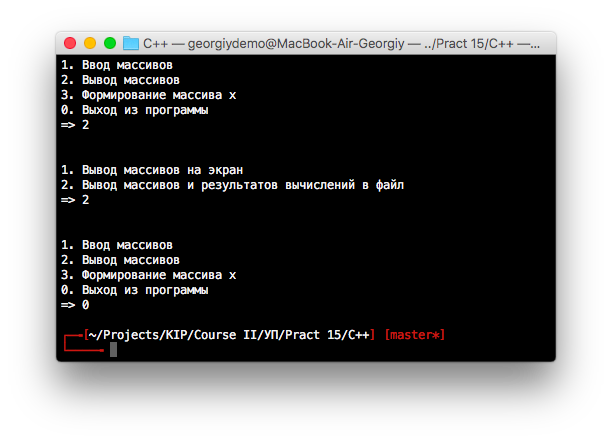
в) Формирование массива x, согласно условию, при b = 2



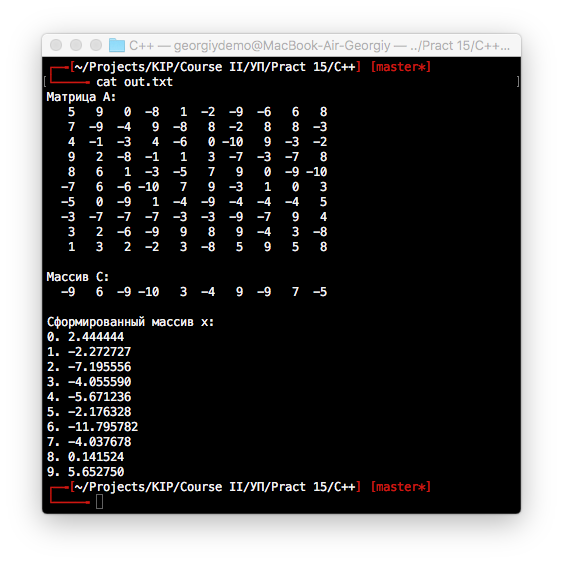
г) Формирование массива x, согласно условию, при b = 5

д) Вывод массивов, введенных вручную

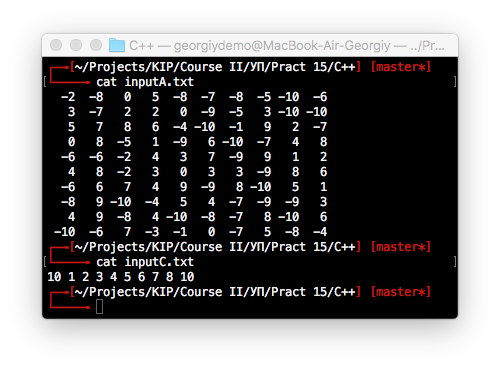


е) Вывод результатов всех операций в файл

ё) Исходный файл на выходе



ж) Исходные файлы с массивами



з) Чтение массивов из файлов

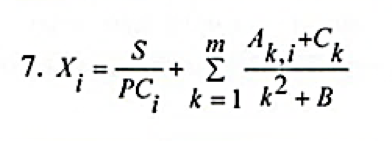


Задание №1 (Pascal)

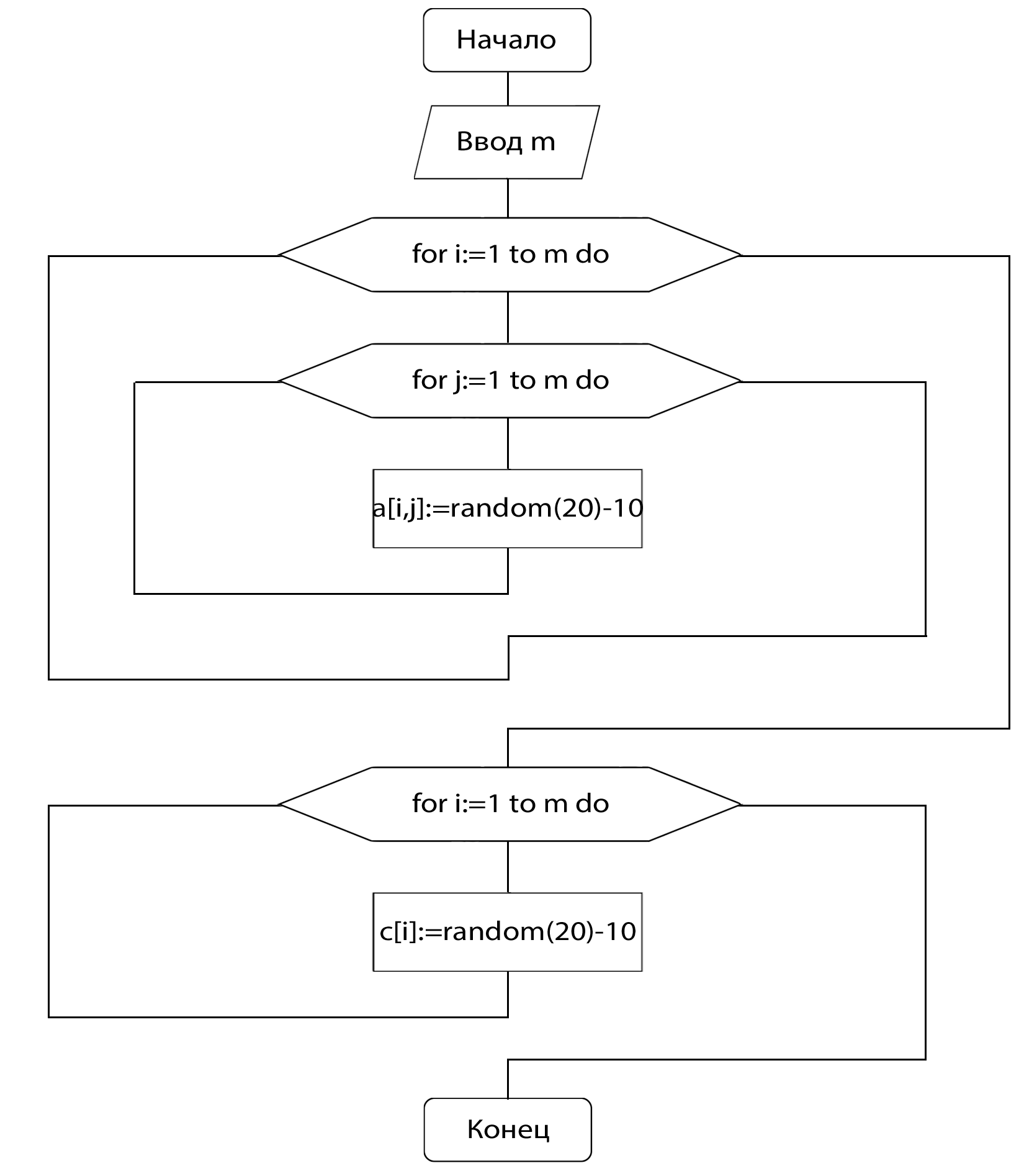
Условие:

Дана переменная В и матрица А, состоящая из m строк и m столбцов. Элементы С образуют одномерный массив из m элементов.

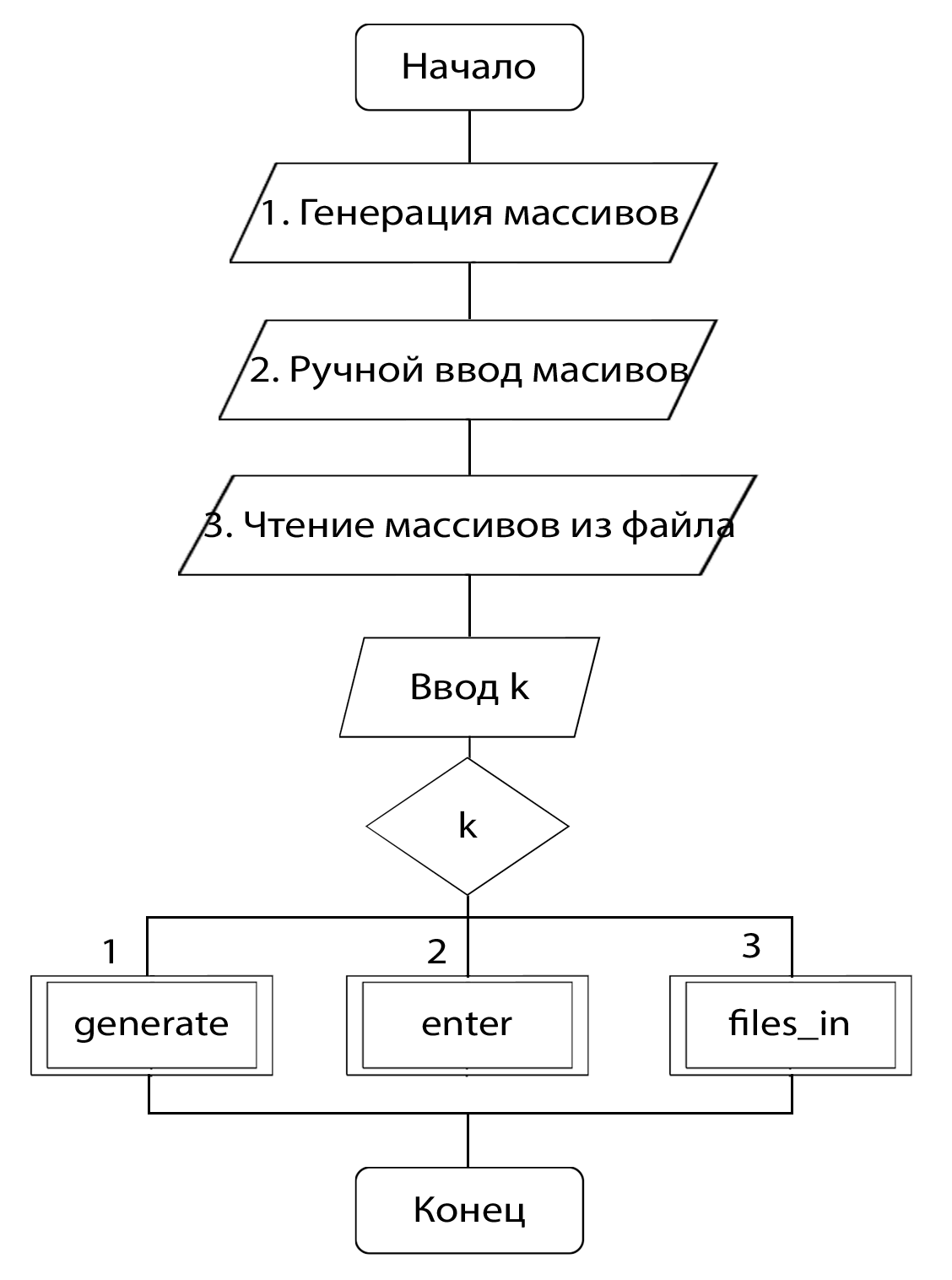
Требуется составить алгоритм и программу вычисления значений элементов одномерного массива Х1,Х2,..., X m (когда вычисляется Хi, Р равно Х i -1, для первого элемента массива Р=1).

S — сумма элементов главной диагонали матрицы А, вычисляемая в алгоритме. 

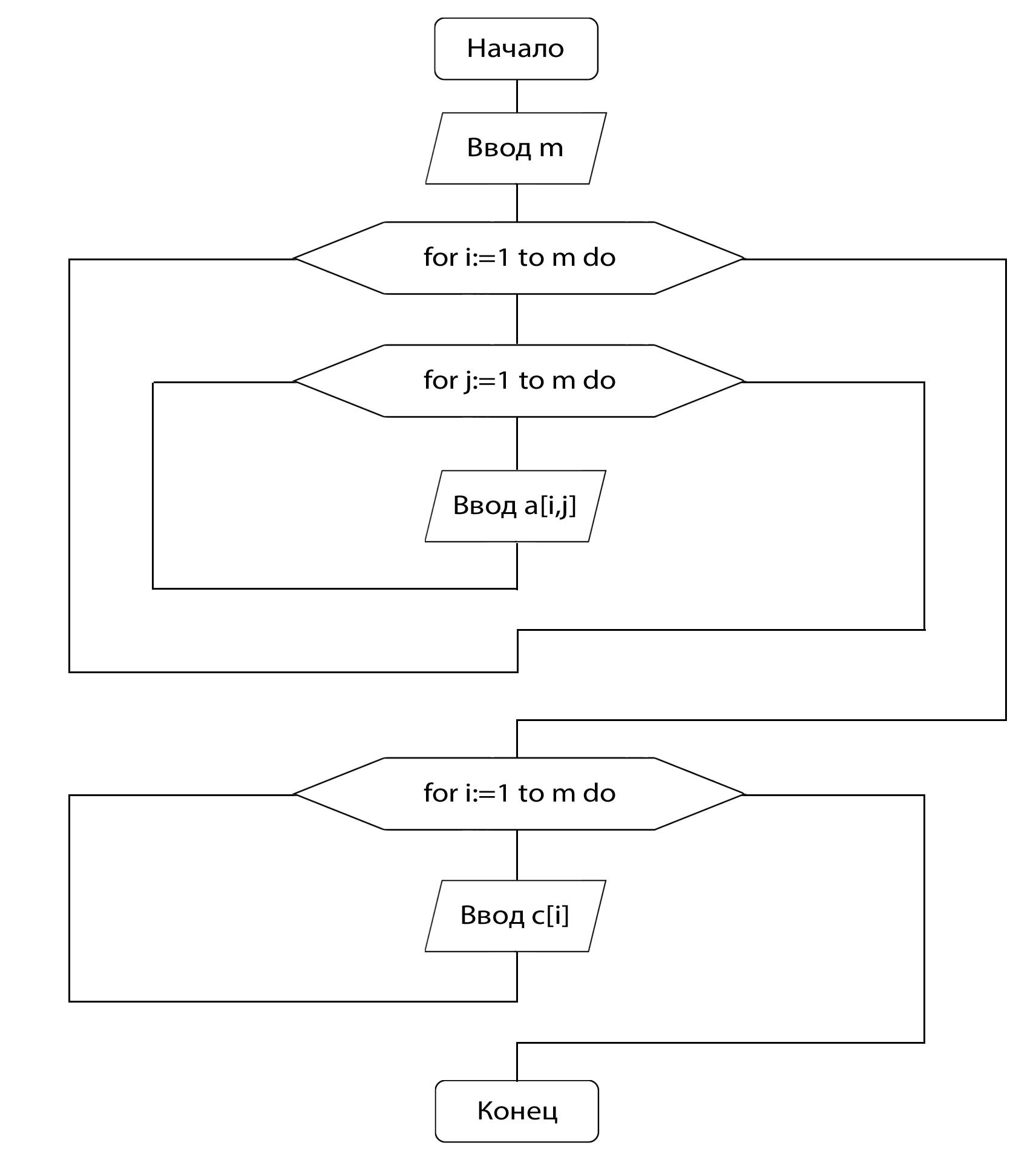
Алгоритм процедуры generate:



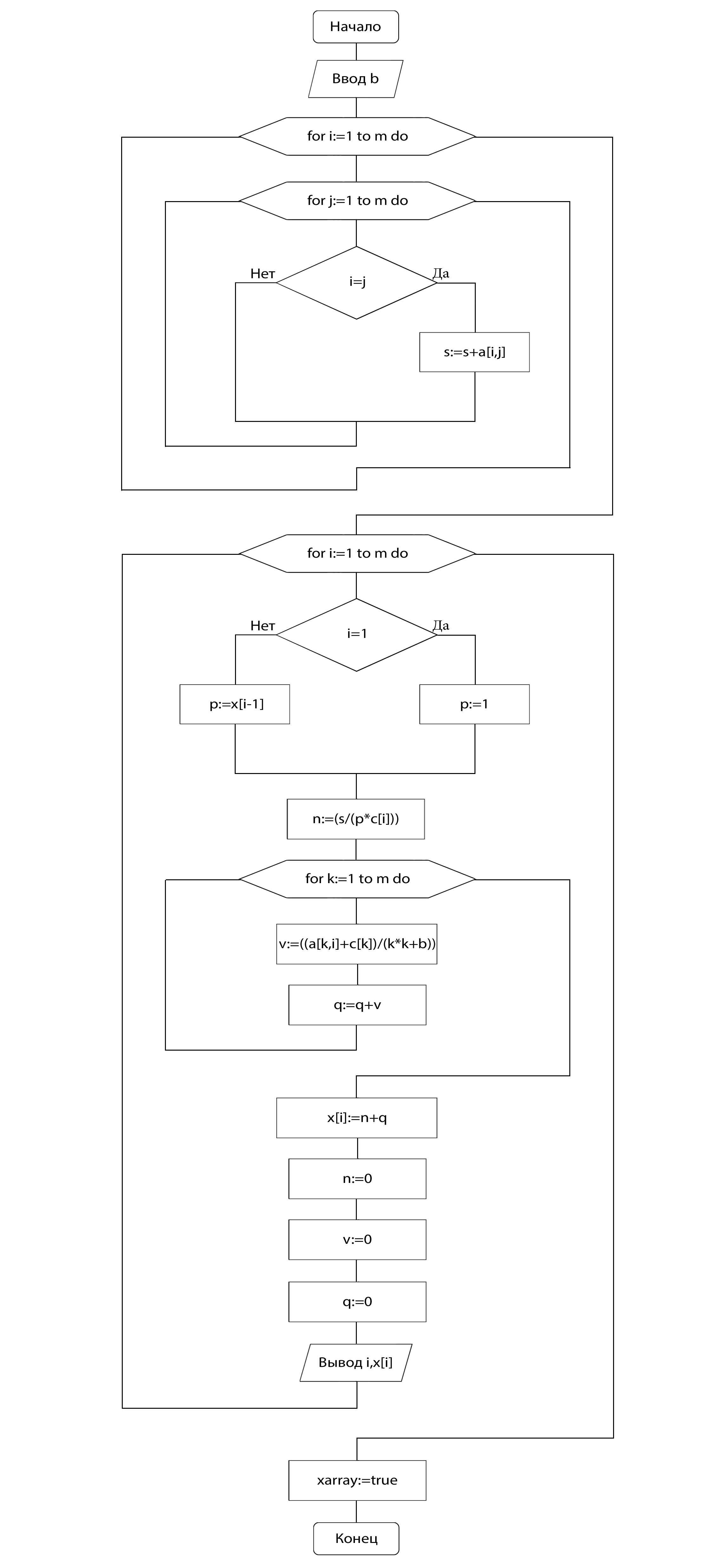
Алгоритм процедуры initarray:

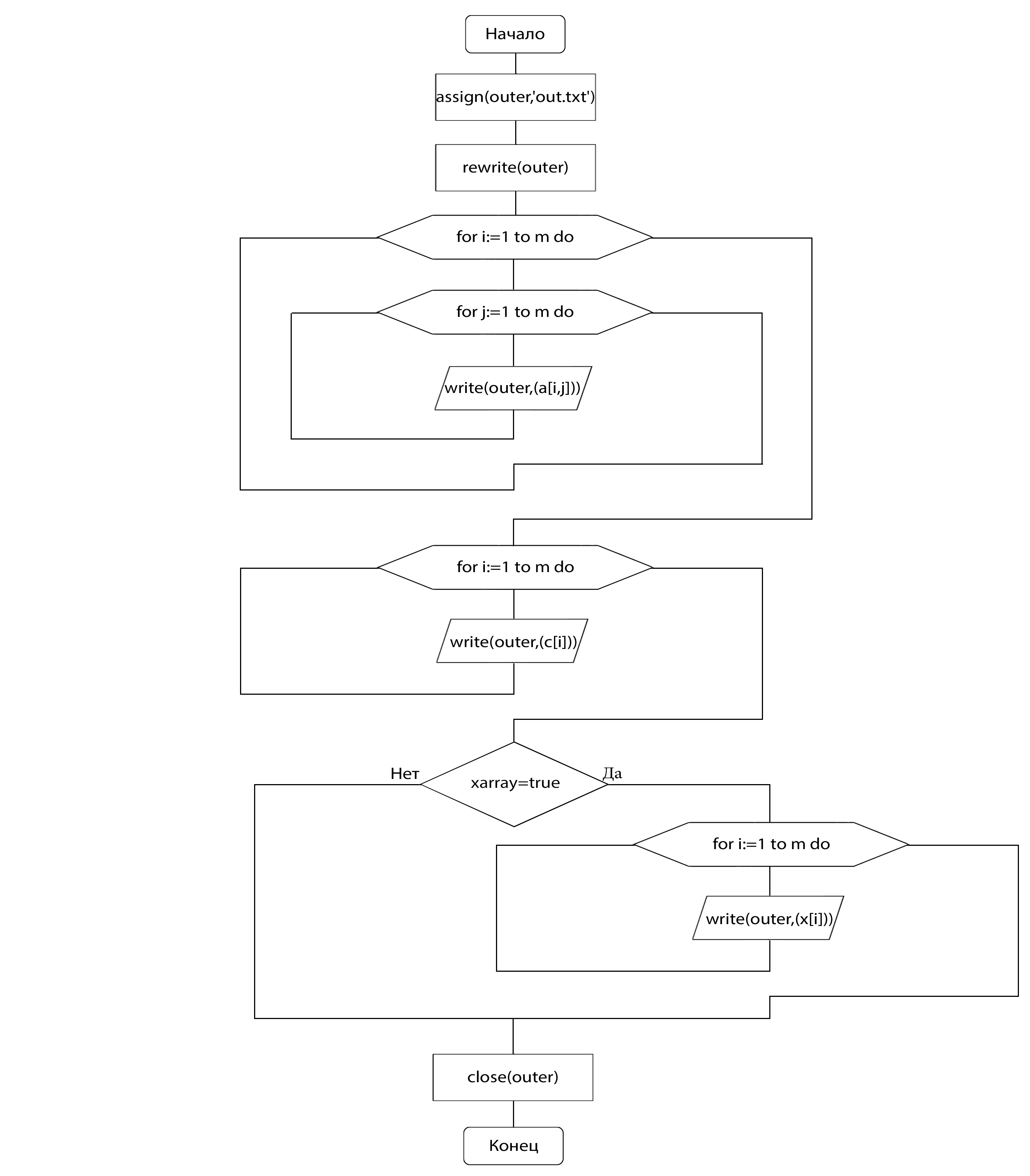


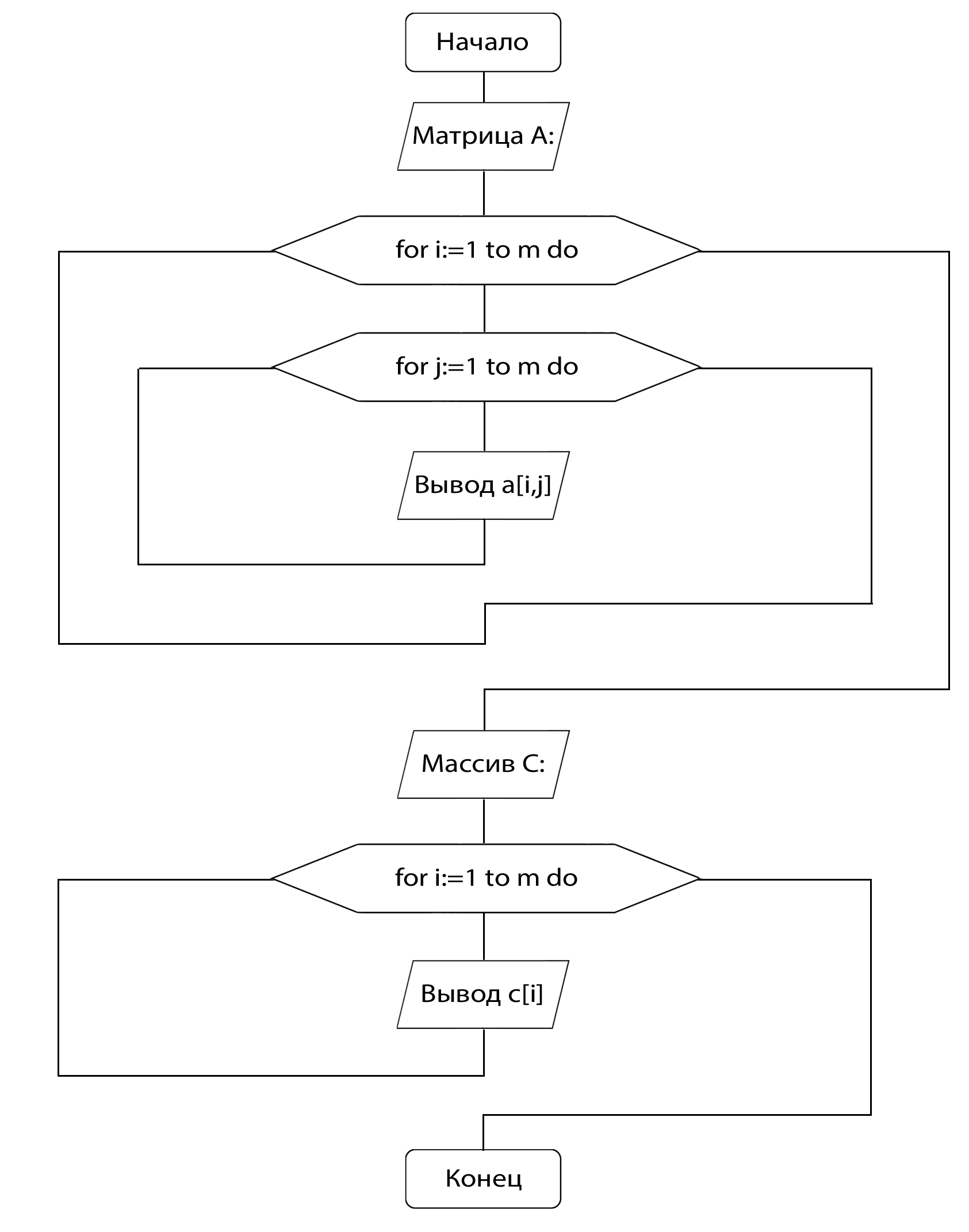
Алгоритм процедуры enter:



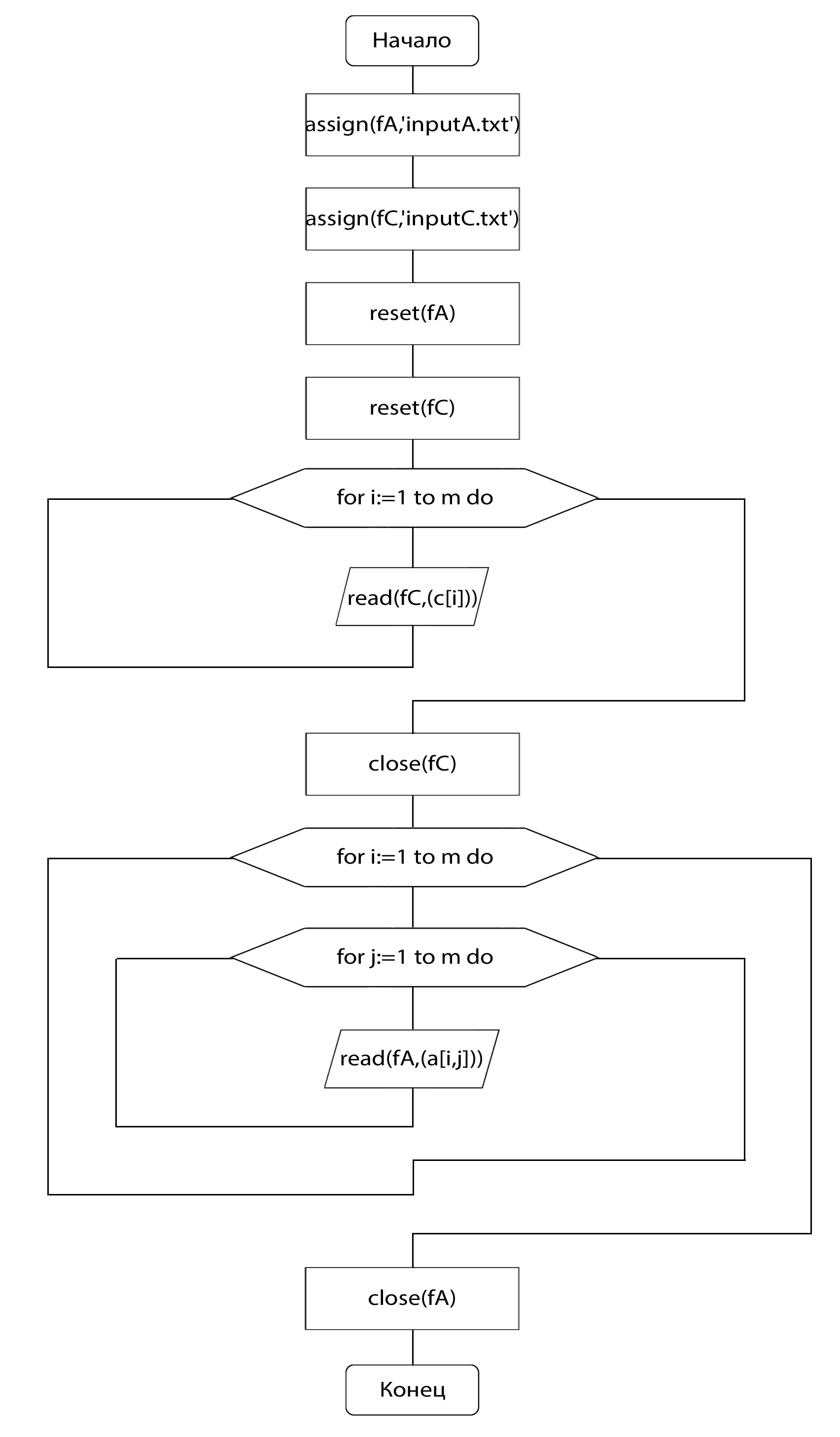
Алгоритм процедуры counter:



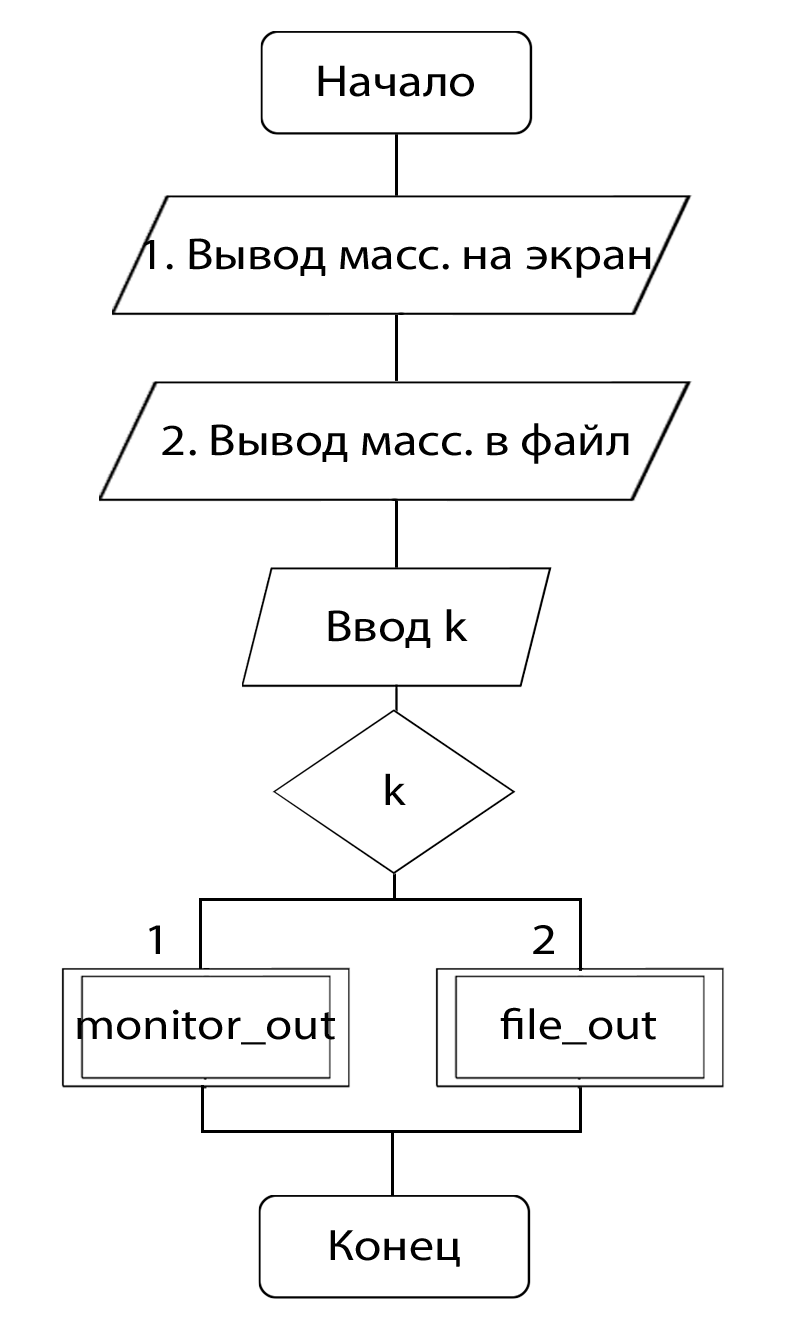
Алгоритм процедуры file\_out:

Алгоритм процедуры monitor\_out:

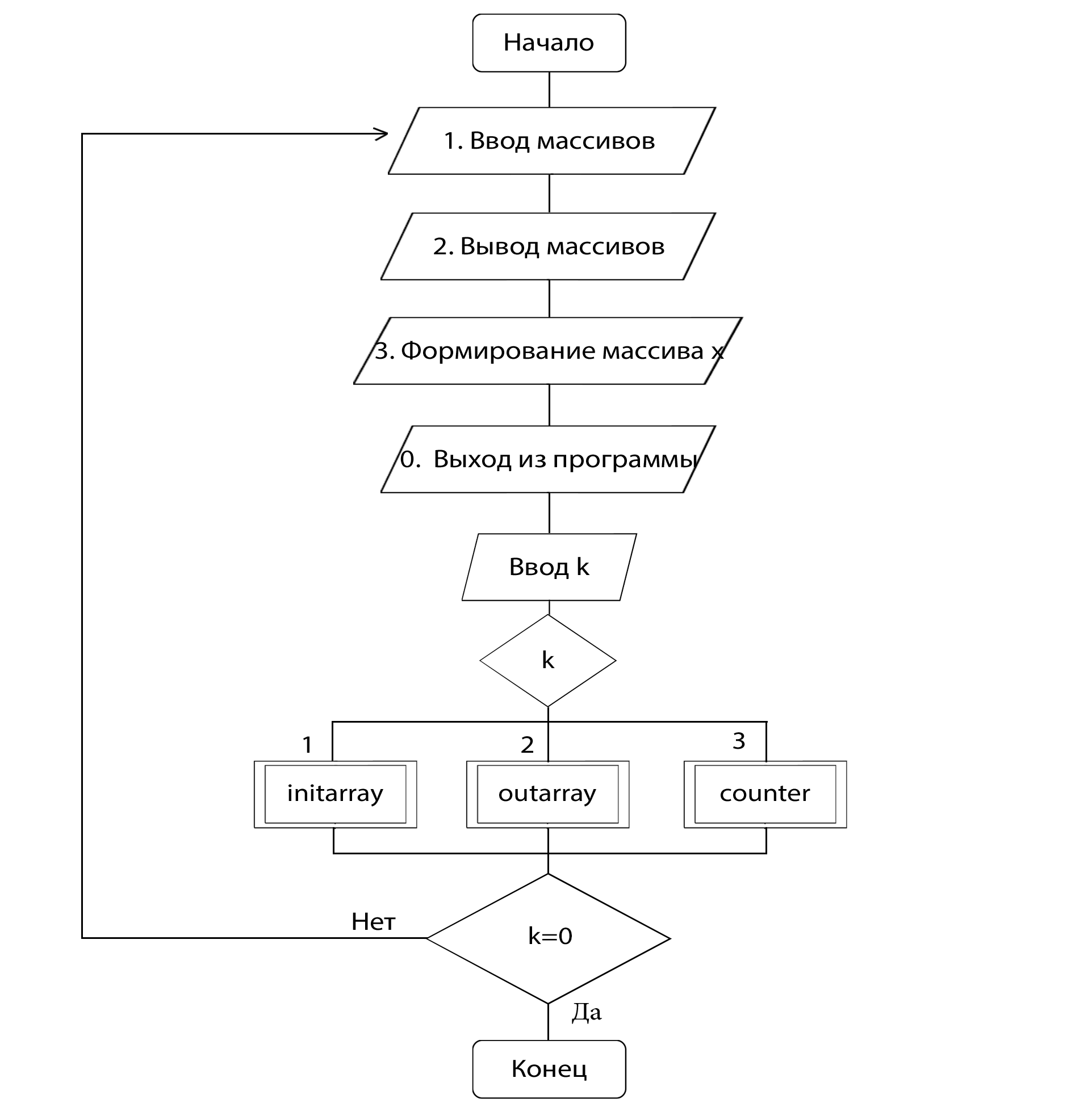
Алгоритм процедуры files\_in:



Алгоритм процедуры outarray:



Алгоритм основной программы:



Исходный код программы:

**program** meow1;

**uses** crt;

**var** a : **array** [1..100,1..100] **of** integer;

    c : **array** [1..100] **of** integer;

    x : **array**[1..100] **of** real;

    k, i, j, s, m, b : integer;

    p, q, n, v : real;

    xarray : boolean;

**procedure** initarray();

**procedure** generate();

**var** i,j:integer;

**begin**

write('Введите размерность массивов => '); read(m);

    randomize;

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

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

a[i,j]:=random(20)-10;

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

c[i]:=random(20)-10;

**end**;

**procedure** enter();

**var** i,j:integer;

**begin**

write('Введите размерность массивов => '); read(m);

    writeln('<Заполняем матрицу А>');

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

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

**begin**

write('Введите элемент [',i,',',j,']: ');

        read(a[i,j]);

**end**;

    writeln('<Заполняем массив C>');

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

**begin**

write('Введите элемент [',i,']: ');

        read(c[i]);

**end**;

**end**;

**procedure** files\_in();

**var** i,j:integer; fA,fC:**file of** integer;

**begin**

assign(fA, 'inputA.txt');

    assign(fC, 'inputC.txt');

    reset(fA); reset(fC);

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

read(fC,(c[i]));

    close(fC);

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

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

read(fA,(a[i,j]));

    close(fA);

**end**;

**begin**

writeln('1. Генерация массивов');

  writeln('2. Ручной ввод масива');

  writeln('3. Чтение массивов из файла');

  write('=> '); readln(k);

**case** k **of**

1 : generate();

       2 : enter();

       3 : files\_in();

**end**;

**end**;

**procedure** outarray();

**procedure** monitor\_out;

**var** i,j:integer;

**begin**

writeln('Матрица А:');

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

**begin**

writeln;

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

write(a[i,j]:4);

**end**;

    writeln;

    writeln('Массив C:');

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

write(c[i]:4);

    readkey();

**end**;

**procedure** file\_out;

**var** outer: **file of** real;

      i,j:integer;

**begin**

assign(outer,'out.txt');

    rewrite(outer);

    //Матрица А

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

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

write(outer, (a[i,j]));

    //Массив С

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

write(outer, (c[i]));

    //Массив x, если он сформирован

**if** (xarray=true) **then**

**begin**

writeln(outer, 'Сформированный массив x:');

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

write(outer, (x[i]));

**end**;

     close(outer);

**end**;

**begin**

writeln('1. Вывод массивов на экран');

    writeln('2. Вывод массивов в файл');

    write('=> '); readln(k);

**case** k **of**

1 : monitor\_out();

       2 : file\_out();

**end**;

**end**;

**procedure** counter();

**var** i,j,k:integer;

**begin**

write('Введите переменную b =>');

  read(b);

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

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

**if** (i=j) **then**

s:=s+a[i,j];

  writeln('Массив x на выходе:');

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

**begin**

**if** (i=1) **then**

p:=1

**else**

p:=x[i-1];

    n:=(s/(p\*c[i]));

**for** k:=1 **to** m **do**

**begin**

v:=((a[k,i]+c[k])/(k\*k+b));

      q:=q+v;

**end**;

    x[i]:=n+q;

    n:=0;

    v:=0;

    q:=0;

    writeln(i,'. ',x[i]);

**end**;

  xarray:=true;

  readkey();

**end**;

**begin**

xarray:=false;

**repeat**

clrscr;

    writeln('1. Ввод массивов');

    writeln('2. Вывод массивов');

    writeln('3. Формирование массива x');

    writeln('0. Выход из программы');

    write('=> '); readln(k);

**case** k **of**

1 : initarray();

       2 : outarray();

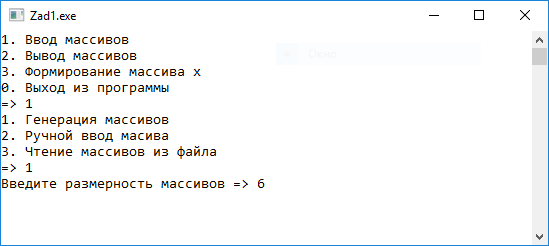
       3 : counter();

**end**;

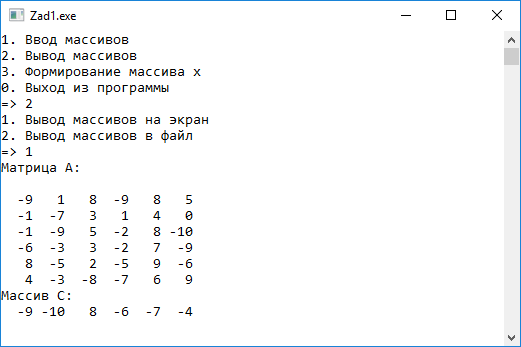
**until** (k=0);

**end**.

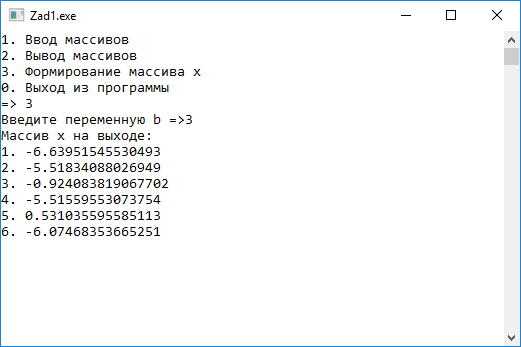
Скриншоты программы:

а) Ввод исходной матрицы через генератор случайных чисел

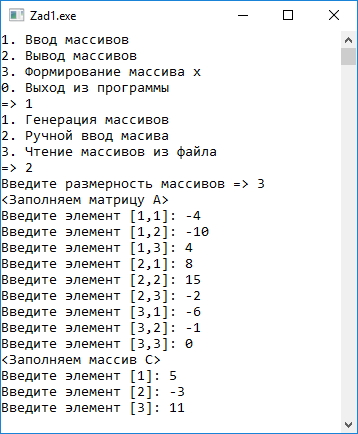
б) Вывод массивов на экран



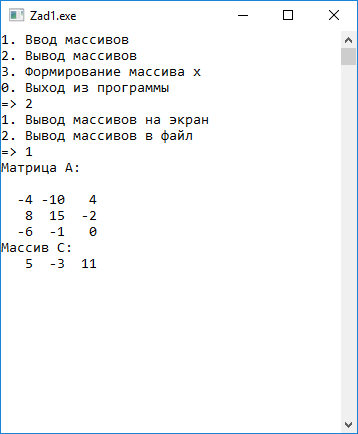
в) Формирование массива x, согласно условию, при b = 3



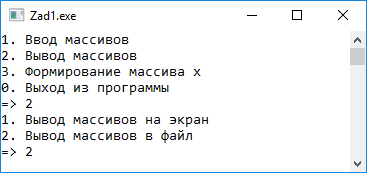
г) Ввод элементов массивов вручную



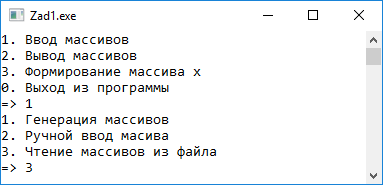
д) Вывод массивов, введенных вручную



е) Вывод результатов всех операций в файл



ё) Чтение массивов из файлов



ж) Вывод массивов, прочитанных из файлов