Практическая работа №15  
Матрицы в Си

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

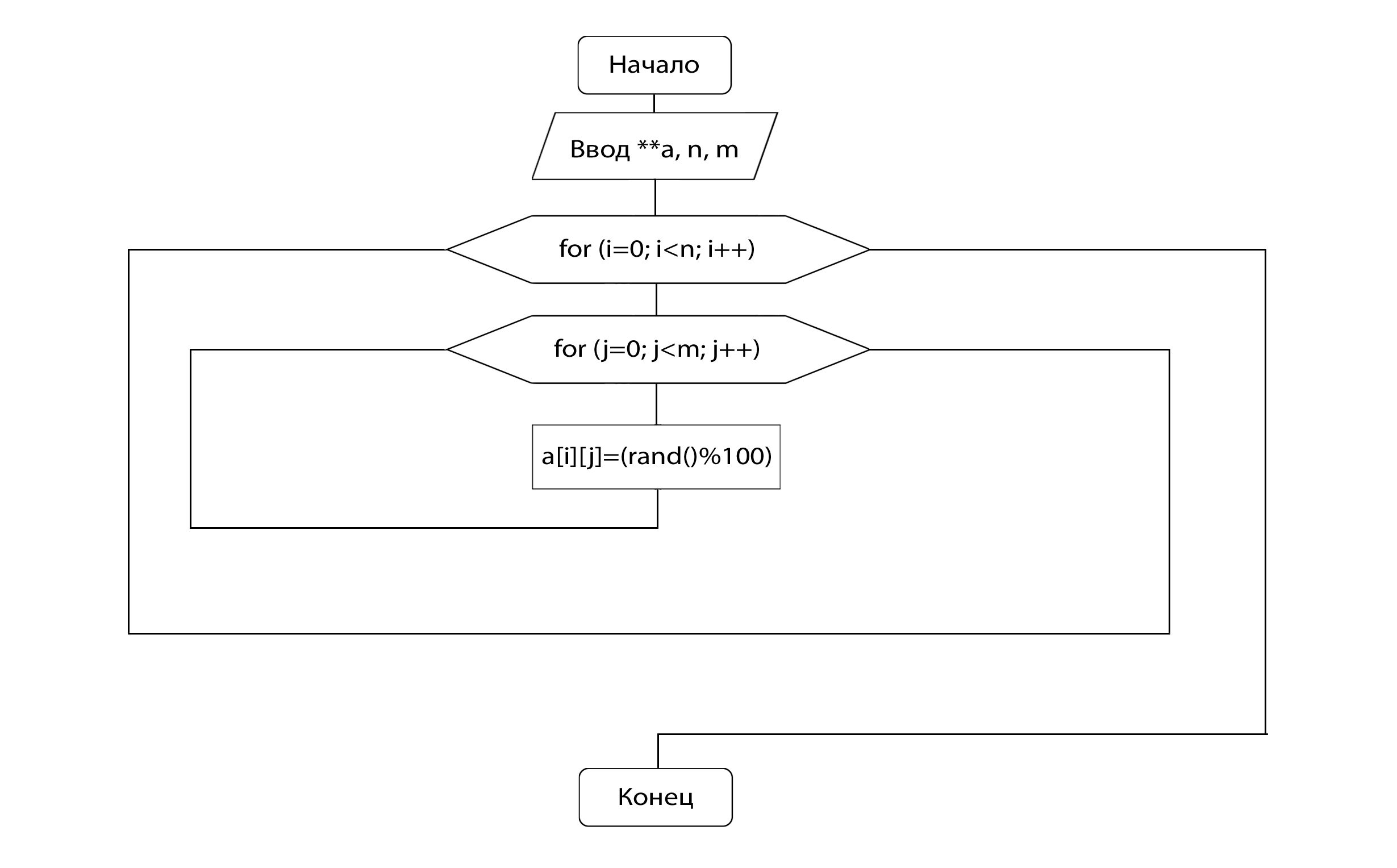
Программа №1

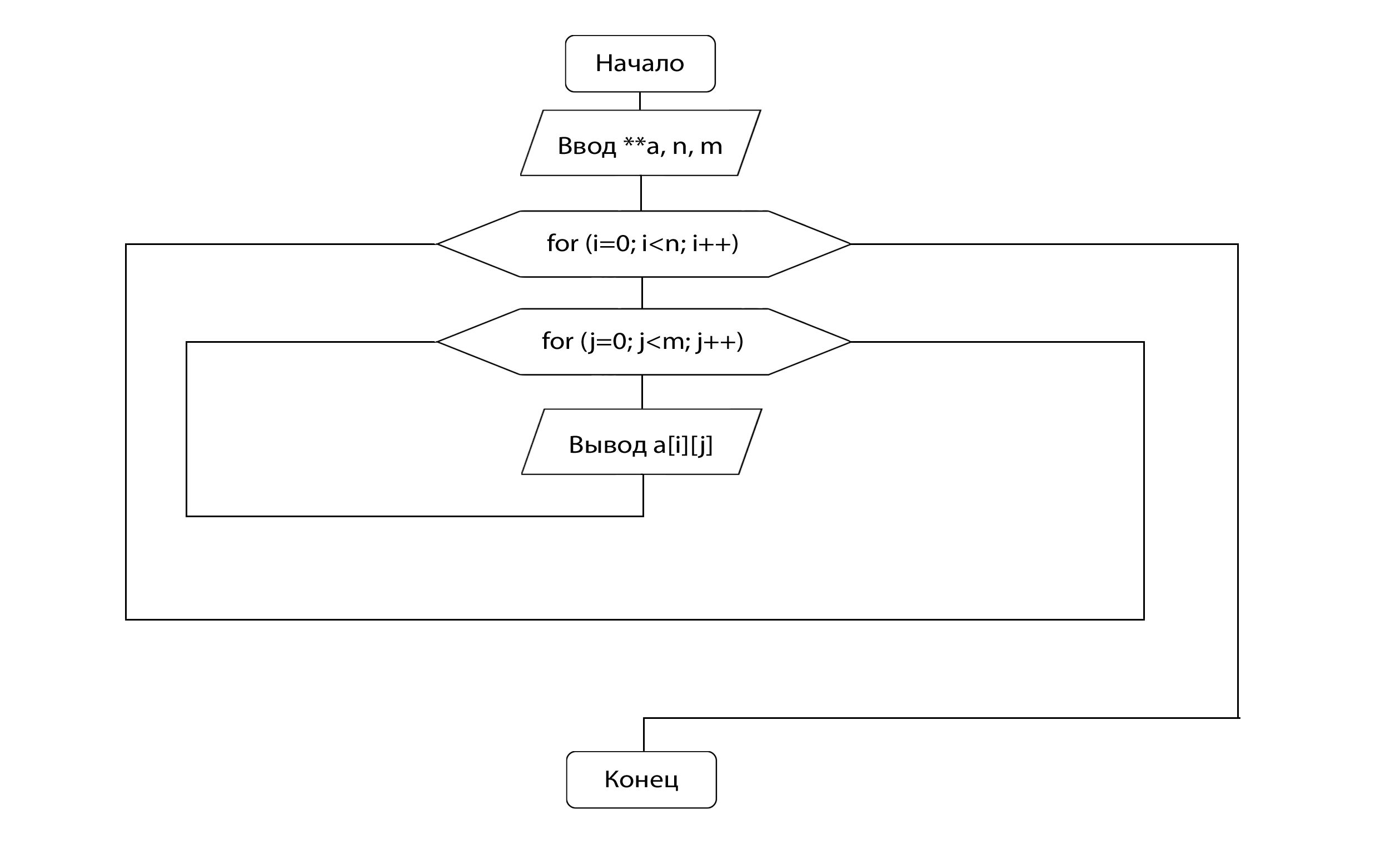
Условие:

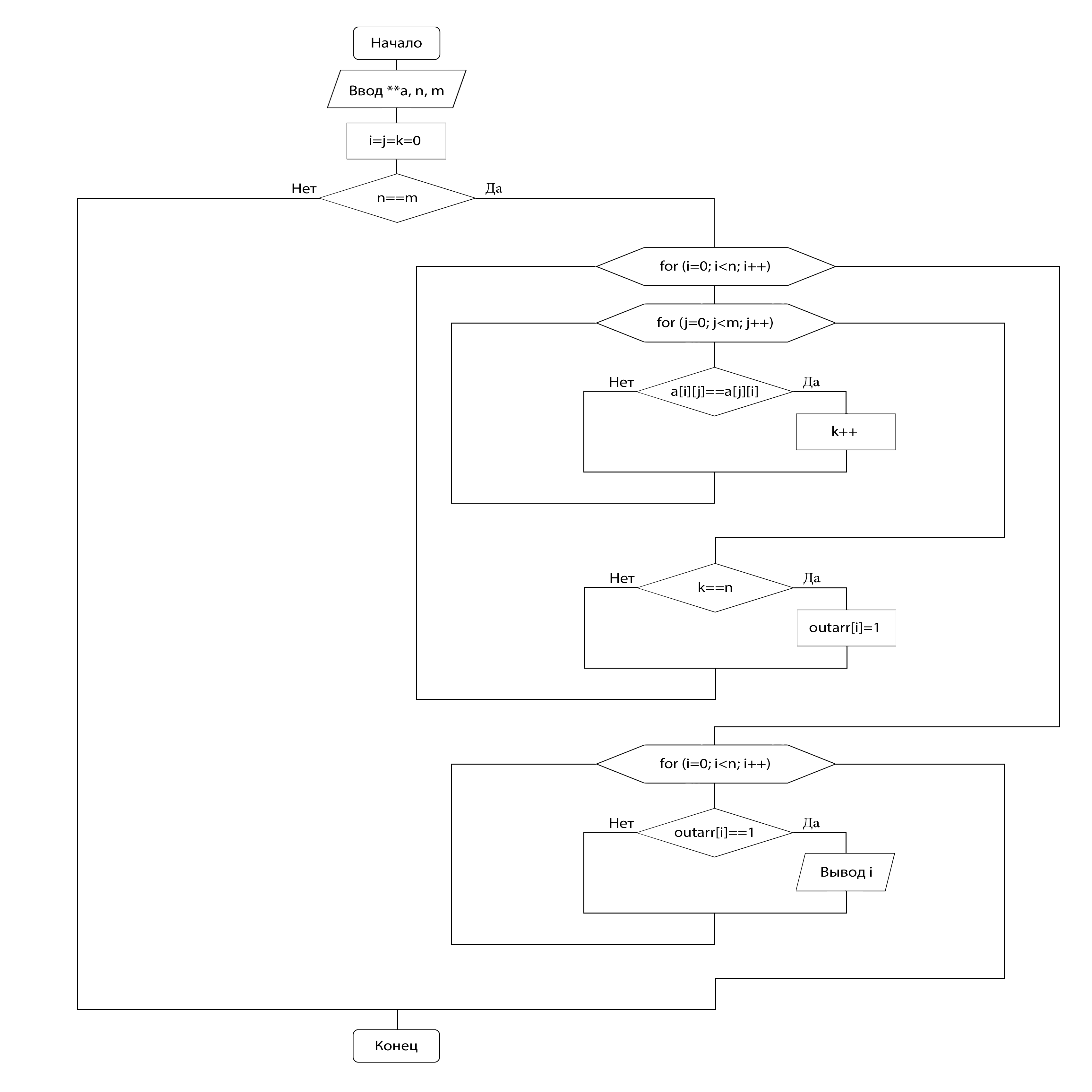
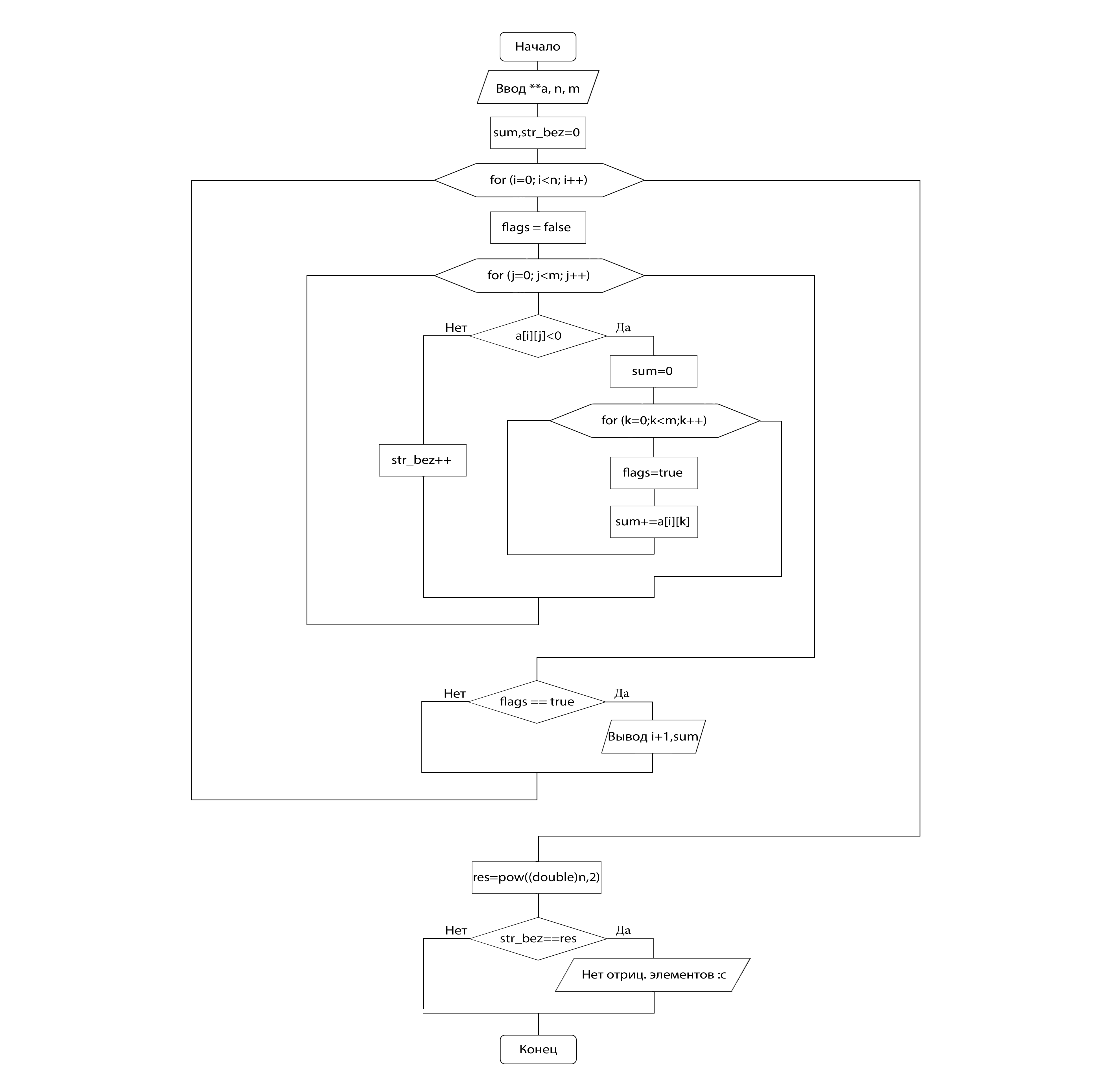
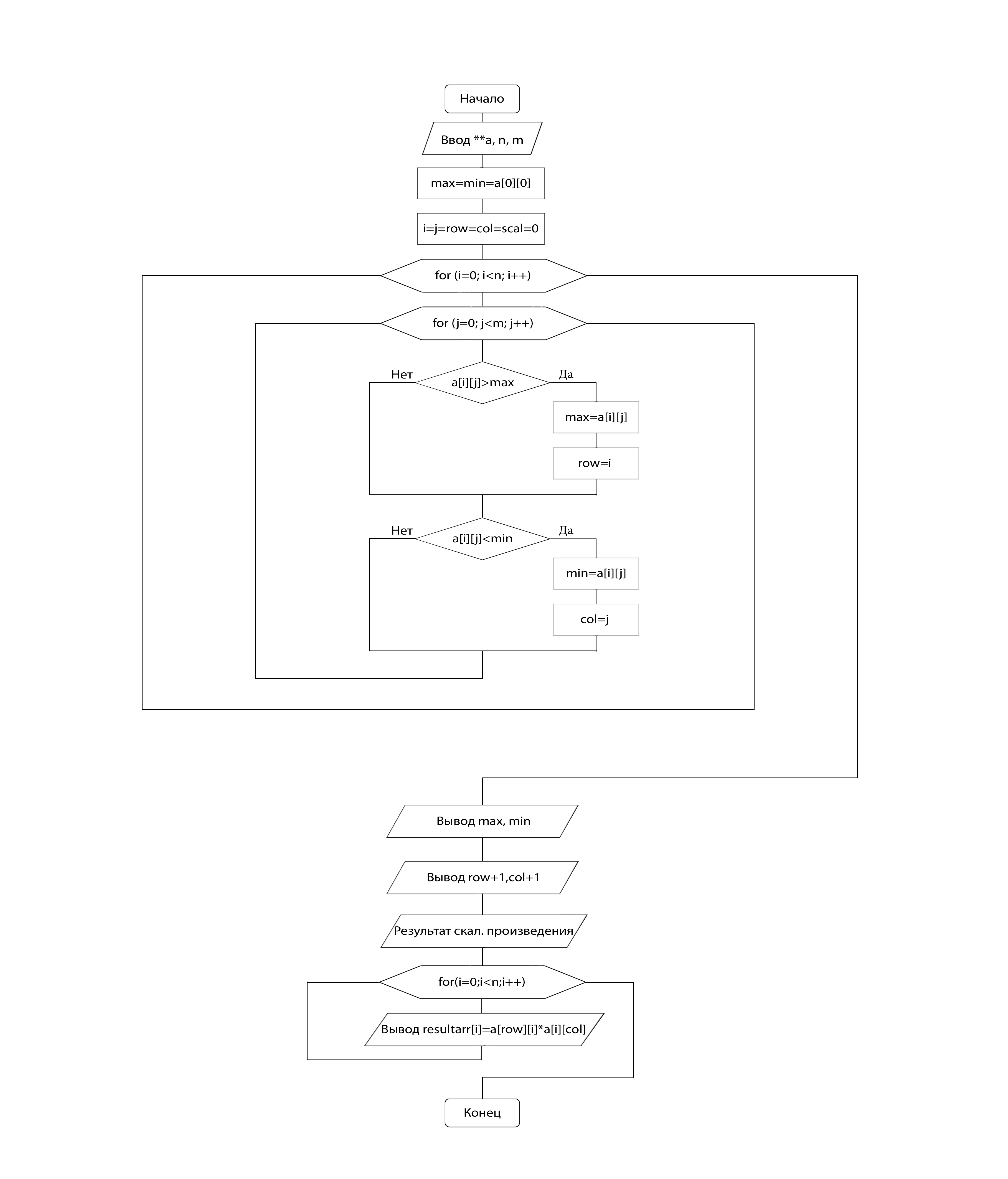
1) Для заданной квадратной матрицы найти такие k, что k-я строка матрицы совпадает с k-м столбцом.

2) Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

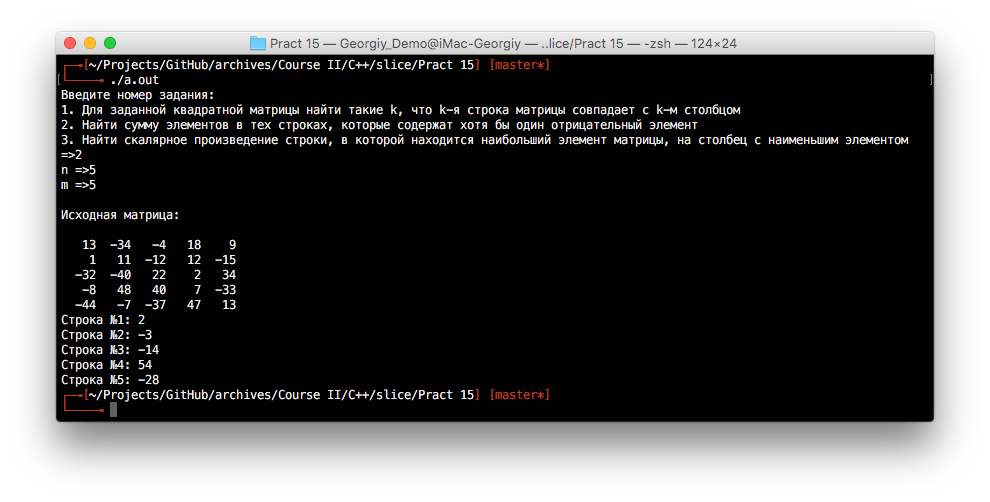
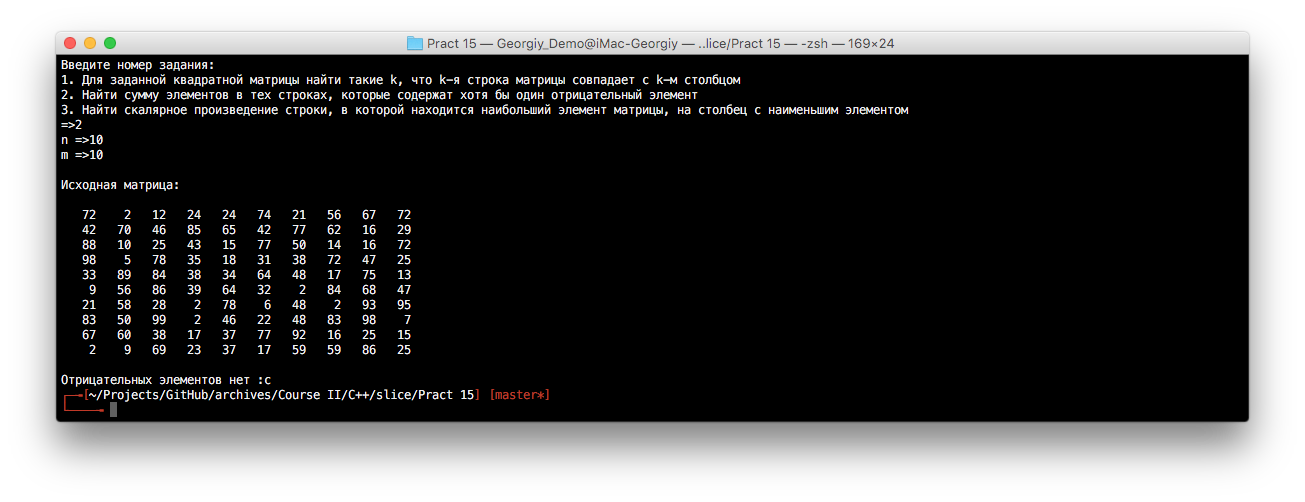
3) Найти скалярное произведение строки, в которой находится наибольший элемент матрицы, на столбец с наименьшим элементом.

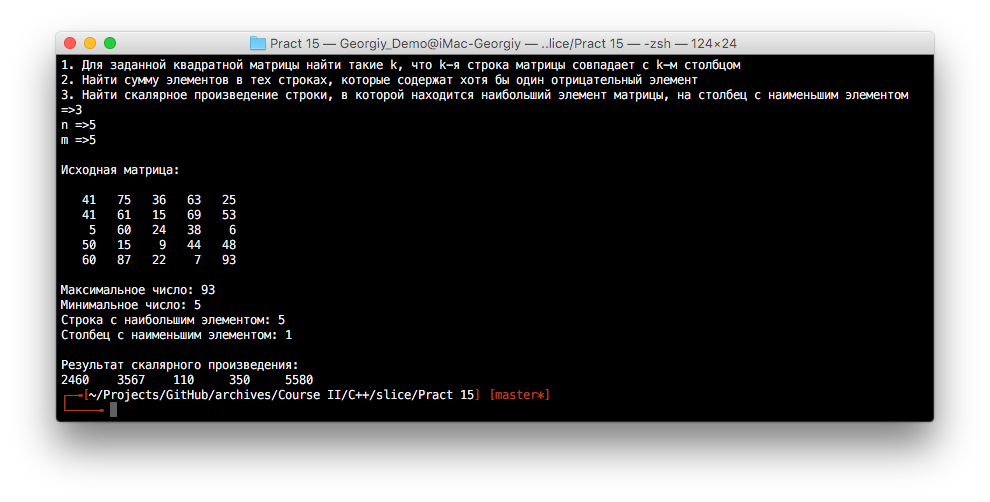
Алгоритм функции initarray (ввод массива):  


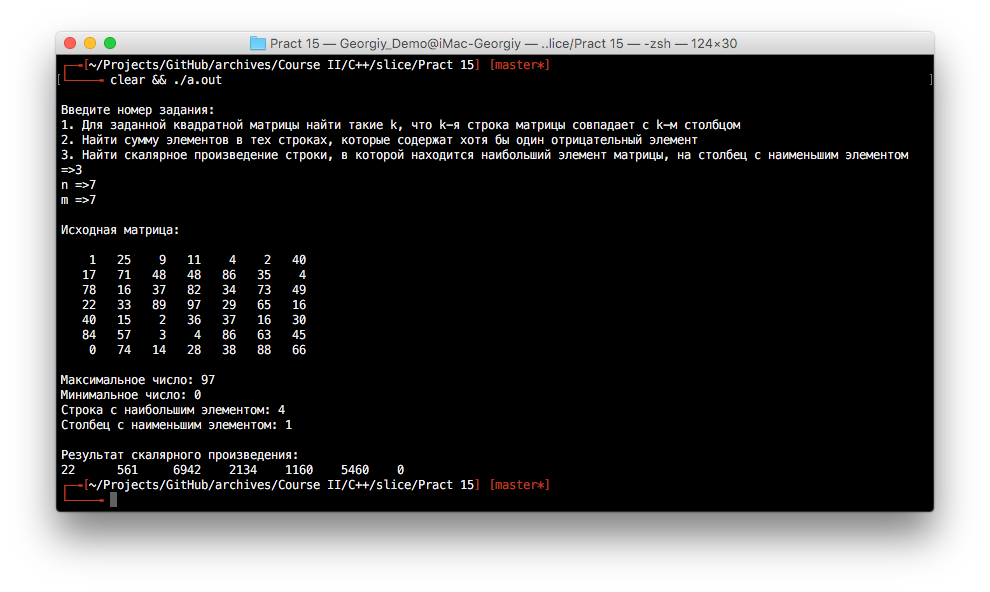
Алгоритм функции outarray (вывод массива):  


Алгоритм функции nomer1 (задание №1):  
  
  
  
  
  
  
  
  
  
  
  
  
  
Алгоритм функции nomer2 (задание №2):  
  
  
  
  
  
  
  
  
  
  
  
  
  
Алгоритм функции nomer3 (задание №3):  
  
  
  
  
  
  
  
  
Исходный код программы:

#include **<iostream>**#include **<string>**#include **<locale>**#include **<iomanip>**#include **<math.h>**#include **<time.h>  
  
using namespace** std;  
  
**int** i,j,n,m,\*\*a;  
  
**void** initarray(**int** \*\*a, **int** n, **int** m){  
 srand((**unsigned**)time(0));   
 **for**(i=0;i<n;i++)  
 **for**(j=0;j<m;j++)  
 a[i][j] = (rand()%100);  
}  
  
**void** outarray(**int** \*\*a, **int** n, **int** m){  
 cout<<**"\nИсходная матрица:\n"**;  
 **for**(i=0;i<n;i++){  
 cout<<**"\n"**;  
 **for**(j=0;j<m;j++)  
 cout<<setw(5)<<a[i][j];  
 }  
}  
  
**void** nomer1(**int** \*\*a, **int** n, **int** m){  
 **int** k, \*outarr;  
 outarr = **new int**[n];  
 i=j=k=0;  
 **if** (n==m){  
 **for**(i=0;i<n;i++){  
 **for**(j=0;j<m;j++){  
 **if**(a[i][j]==a[j][i])  
 k++;  
 }  
 **if**(k==n)  
 outarr[i]=1;  
 printf(**"\n"**);  
 }  
 **for**(**int** i=0;i<n;i++){  
 **if**(outarr[i]==1)  
 printf(**"Совпадают строка и столбец под №%d\n"**,i);  
 }  
 }  
}  
  
**void** nomer2(**int** \*\*a, **int** n, **int** m){  
 **int** sum, str\_bez = 0;  
 **bool** flags;  
 **for**(**int** i = 0; i < n; i++){  
 flags = **false**;  
 **for**(**int** j = 0; j < m; j++){  
 **if**(a[i][j] < 0){  
 sum = 0;  
 **for**(**int** k = 0; k < m; k++){  
 flags = **true**;  
 sum += a[i][k];  
 }  
 }  
 **else** str\_bez++;  
 }  
 **if**(flags == **true**)  
 cout << **"\nСтрока №"** << i+1 <<**": "**<< sum;  
 }  
 **int** res = pow((**double**)n, 2);  
 **if**(str\_bez == res)  
 cout << **"\n\nОтрицательных элементов нет :c"**;  
 cout << **"\n"**;   
}  
  
**void** nomer3(**int** \*\*a, **int** n, **int** m){  
**int** resultarr[n+m];  
**int** i,j,min,max,scal,row,col;  
max=min=a[0][0];  
i=j=row=col=scal=0;  
**for**(**int** i=0; i<n; i++)  
 **for**(**int** j=0; j<m; j++){  
 **if** (a[i][j]>max){  
 max=a[i][j];  
 row=i;  
 }  
 **if** (a[i][j]<min){  
 min=a[i][j];  
 col=j;  
 }  
 }  
 cout<<**"\n\nМаксимальное число: "**<<max<<**"\nМинимальное число: "**<<min<<**'\n'**;  
 cout<<**"Строка с наибольшим элементом: "**<<row+1<<**"\nСтолбец с наименьшим элементом: "**<<col+1<<**'\n'**;  
 cout<<**"\nРезультат cкалярного произведения:\n"**;  
 **for**(i=0; i<n; i++){  
 cout<<(resultarr[i]=a[row][i]\*a[i][col])<<**'\t'**;  
 }  
 cout<<**"\n"**;  
}  
  
**int** main(){  
 setlocale(LC\_ALL, **"Russian"**);  
 **int** k;  
  
 cout<<**"Введите номер задания:\n1. Для заданной квадратной матрицы найти такие k, что k-я строка матрицы совпадает с k-м столбцом\n2. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент\n3. Найти скалярное произведение строки, в которой находится наибольший элемент матрицы, на столбец с наименьшим элементом"**;  
 cout<<**"\n=>"**;  
 cin>>k;  
 cout<<**"n =>"**; cin>>n;  
 cout<<**"m =>"**; cin>>m;  
  
 a = **new int** \*[n];  
 **for**(i=0;i<n;i++)  
 a[i]=**new int**[n];  
   
 initarray(a,n,m);  
 outarray(a,n,m);  
   
 **switch** (k)   
 {   
 **case** 1:{  
 nomer1(a,n,m);  
 **break**;  
 }  
 **case** 2:{  
 nomer2(a,n,m);  
 **break**;  
 }  
 **case** 3:   
 nomer3(a,n,m);  
 **break**;  
 **default**:   
 cout<<**"Нет такого задания :с"**;   
 }   
}

Скриншоты программы:  
  






-----------------------------------------------------------------------------------------------------------------------------------------------------------------