

МИНИСТЕРСТВО ПО ОБРАЗОВАНИЮ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Р.Е.Алексеева»

Кафедра «Прикладная математика»

РЕШЕНИЕ ТИПОВЫХ ЗАДАЧ ПО КУРСУ
«ИНФОРМАТИКА»
(язык программирования C/C++)
Методическая разработка
для студентов всех форм обучения
и всех специальностей

Нижний Новгород 2012

Составители: Т.В. Моругина, С.П. Никитенкова, О.И. Чайкина

УДК 651.3.06

Решение типовых задач по курсу «Информатика» (язык программирования С/С++): методическая разработка для студентов всех форм обучения и всех специальностей / НГТУ; сост.: Т.В. Моругина, С.П. Никитенкова, О.И. Чайкина. Н.Новгород, 2012- 27 с.

Изложены примеры решения типовых задач к лабораторным работам по курсу «Информатика» (язык программирования С/С++).

Научный редактор С.Н. Митяков
Редактор Э.Б. Абросимова

Подп. к печ. 14.09.12 Формат 60x84 $\frac{1}{16}$.- Бумага- газетная.- Печать офсетная.Печ.л.1,5 .Уч.-изд. л. 1 . Тираж 100 экз. Заказ

Нижегородский государственный технический университет им. Р.Е.Алексеева.

Типография НГТУ. 603950,Н.Новгород, ул.Минина,24.

© Нижегородский государственный
технический университет им. Р.Е.Алексеева,2012

Порядок выполнения работы в программе Microsoft Visual Studio

Создание проекта и добавление исходного файла

1. Создайте проект (последовательно выбрав в меню **File** пункты **New** и **Project**).
2. В области типов проектов **Visual C++** выберите группу **Win32** и щелкните элемент **Win32 Console Application**.
3. Введите имя проекта.

По умолчанию имя решения, содержащего проект, совпадает с именем проекта. Можно указать другое расположение для проекта.

Нажмите кнопку **ОК**, чтобы создать проект.

4. В мастере приложений **Win32** нажмите кнопку **Next**, выберите вариант **Console application** и нажмите кнопку **Finish**.
5. В файле, открытом в редакторе, введите программный код на языке C.
6. Сохраните файл.
7. Произвести компиляцию нажав **CTR+F5**(или выбери команду меню **Debug** затем из раскрытого меню **Start Without Debugging**)

В окне **Output** выводятся сведения о ходе компиляции.

Если есть ошибки, необходимо исправить и повторить компиляцию.

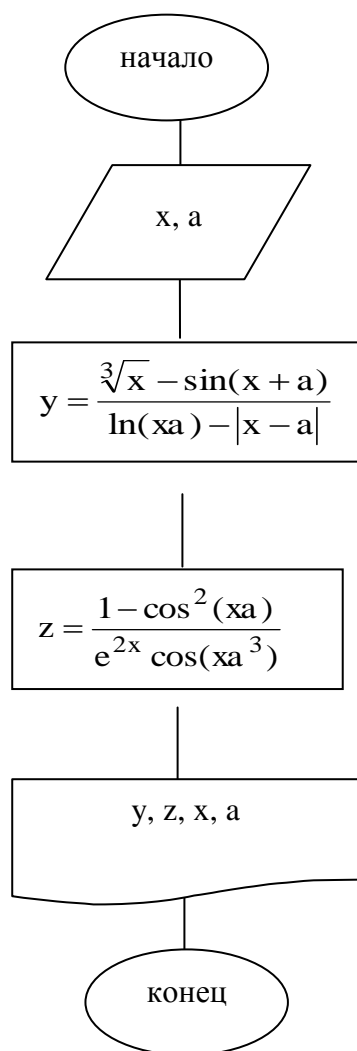
Если ошибок нет, система перейдет в окно выполнения программы.

Линейный алгоритм

1. Вычислить значения функций при $x = 0,5$ и $a = 0,7$

$$y = \frac{\sqrt[3]{x} - \sin(x + a)}{\ln(xa) - |x - a|}, \quad z = \frac{1 - \cos^2(xa)}{e^{2x} \cos(xa^3)}$$

Блок – схема



Программа на языке C

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;

int main()
{
    // переменные x,a,y,z вещественного типа
    double x, a,y,z;
    //ввод данных
    cout << "Введите x,a: " << endl;
    cin >> x >> a ;
    //вычисление функции y и z
    y=(pow(x,1./3)-sin(x+a))/(log(x*a)-fabs(x-a));
    z=(1-pow(cos(x*a),2))/
        (exp(2*x)*cos(x*pow(a,3)));

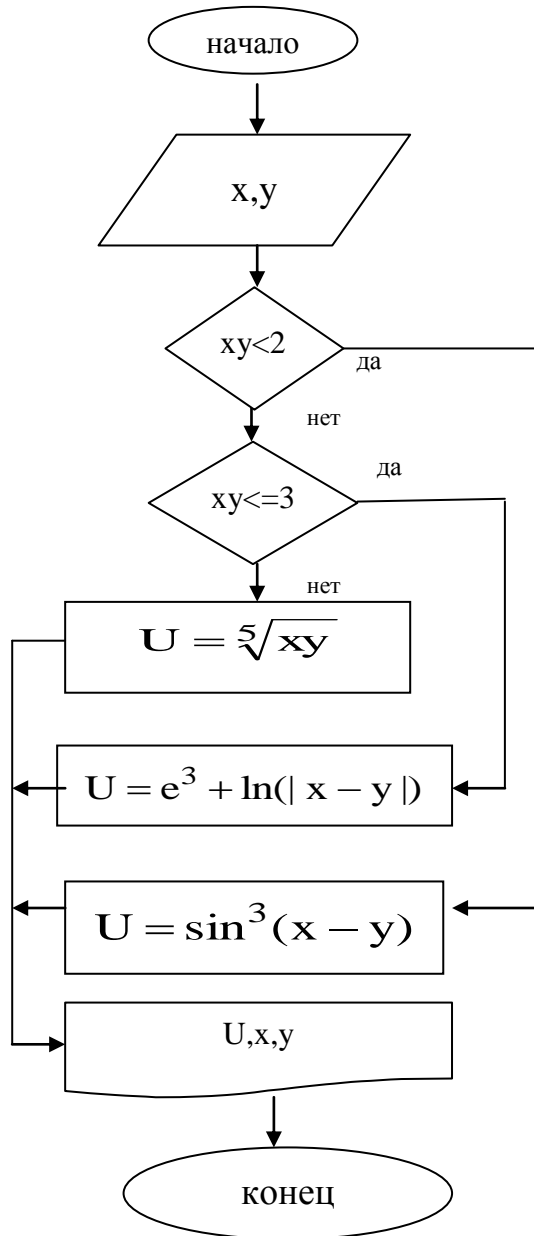
    //вывод результатов и исходных данных
    cout<<"Исходные данные"<<endl;
    cout<<" x= " <<x<<" a= " <<a<<endl;
    cout<<"результат \n";
    cout<<" y = " <<y<<" z= " <<z<<endl;
    return 0;
}
```

Разветвляющийся алгоритм

Вычислить значение при заданных $x=0,2$ и $y=0,1$

$$U = \begin{cases} \sin^3(x-y), & \text{если } xy < 2 \\ e^3 + \ln(|x-y|), & \text{если } 2 \leq xy \leq 3 \\ \sqrt[5]{xy}, & \text{если } xy > 3 \end{cases}$$

Блок – схема



Программа на языке C

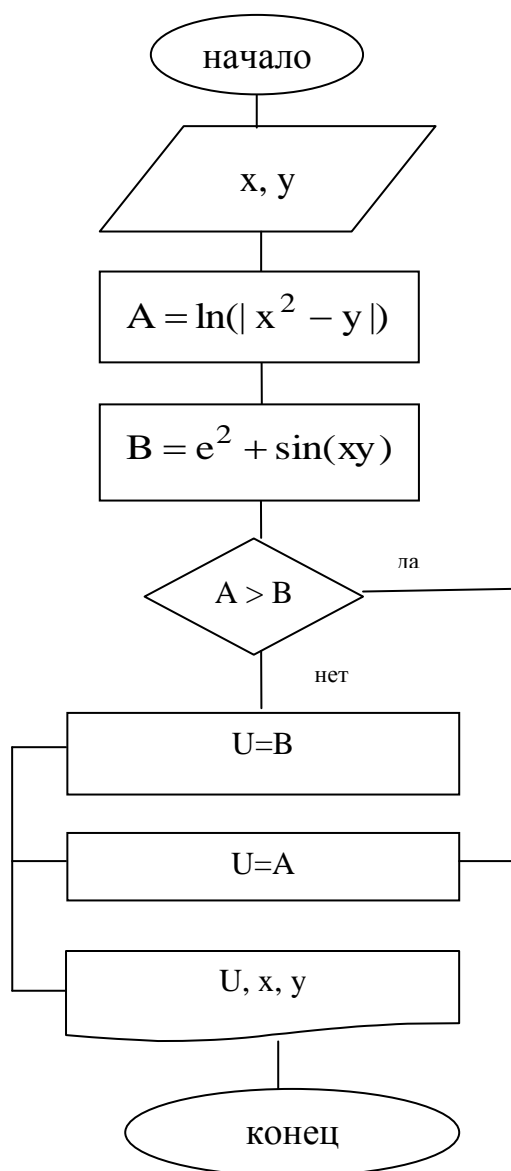
```

#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;
int main()
{ // переменные x,y,U вещественного
  типа
    double x, y,U;
  //ввод данных
  cout << "Введите x,y: " << endl;
  cin >> x >> y ;
  //вычисление функции U
  if (x*y < 2) U=pow(sin(x-y),3);
  else if(x*y <= 3)
  U=exp(3)+log(fabs(x-y));
  else U=pow(x*y,1./5);
  //вывод исходных данных и
  результата
  cout << "Исходные данные \n";
  cout << "x= " << x << "y= " << y << endl;
  cout << "Результат U= " << U << endl;
  return 0; }
    
```

Вычислить значение функции при $x=0,6$ и $y=0,4$.

$$U = \max \left\{ \ln(|x^2 - y|); e^2 + \sin(xy) \right\}$$

Блок -схема



Программа на языке C

```
#include "stdafx.h"

#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;

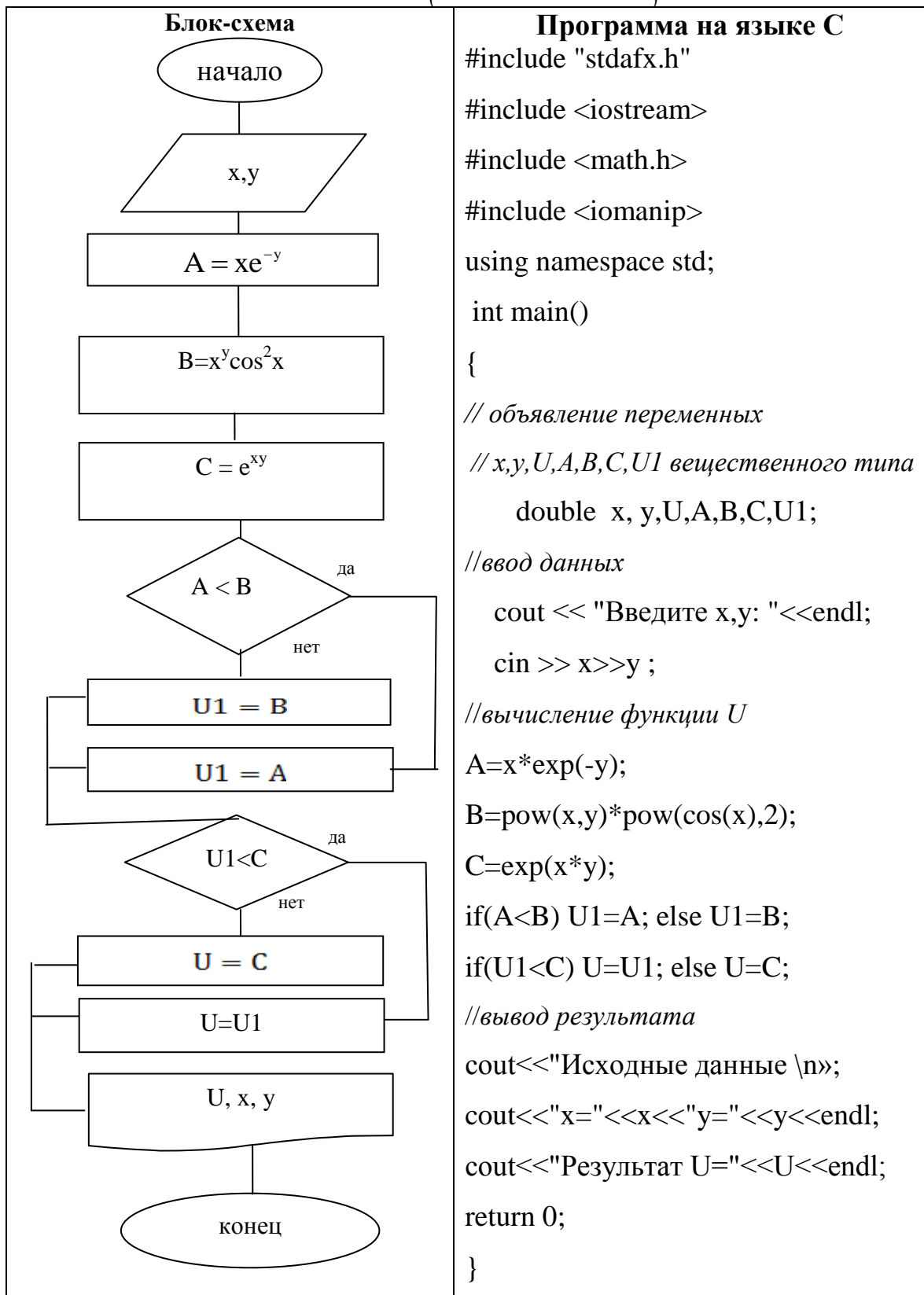
int main()
{
    // объявление переменных x,y,U
    // вещественного типа
    double x, y,U,A,B;
    //ввод данных
    cout << "Введите x,y: " << endl;
    cin >> x >> y ;
    //вычисление функции U
    A=log(fabs(pow(x,2)-y));
    B=exp(2.0)+sin(x*y);

    if(A>B) U=A;
    else U=B;
    //вывод исходных данных и
    //результата

    cout<<"Исходные данные \n";
    cout<<"x="<<x<<"y="<<y<<endl;
    cout<<"Результат U="<<U<<endl;
    return 0;
}
```

Вычислить значение функции при $x=0,1$ и $y=0,2$.

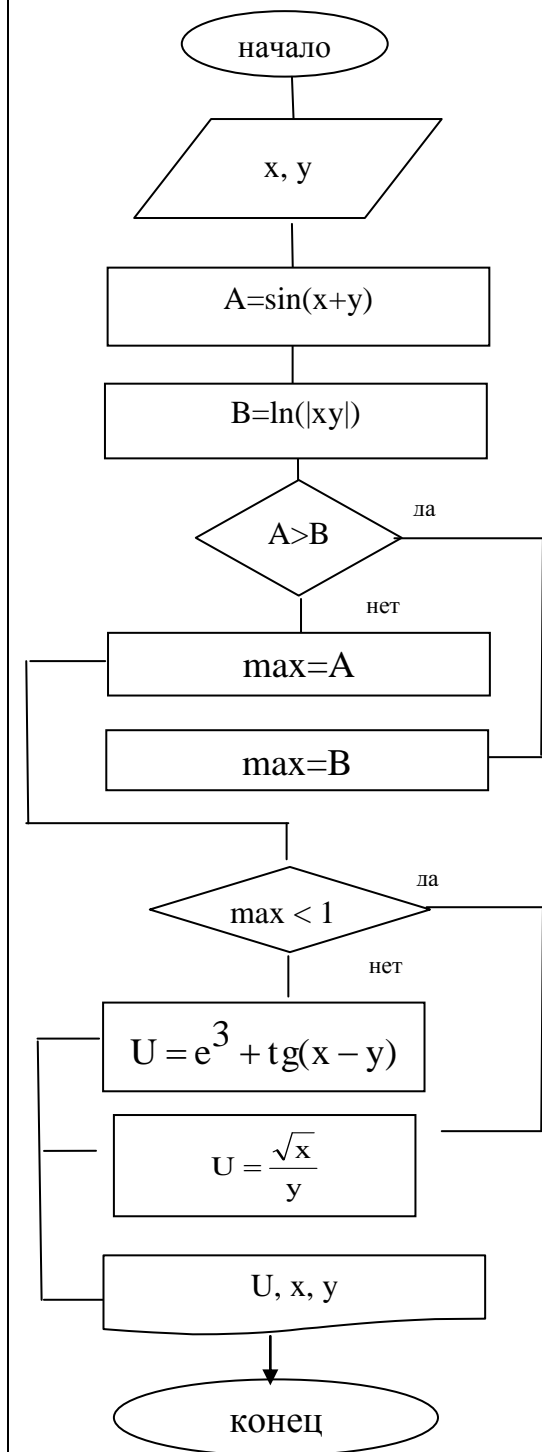
$$U = \min \{ x e^{-y}; x^y \cos^2 x; e^{xy} \}$$



Вычислить значение функции при $x=0,7$ и $y=0,2$.

$$U = \begin{cases} \frac{\sqrt{x}}{y} & ; \text{если } \max \{ \sin(x+y); \ln(|xy|) \} < 1 \\ (e^3 + \operatorname{tg}(x-y)) & ; \text{если } \max \{ \sin(x+y); \ln(|xy|) \} \geq 1 \end{cases}$$

Блок – схема

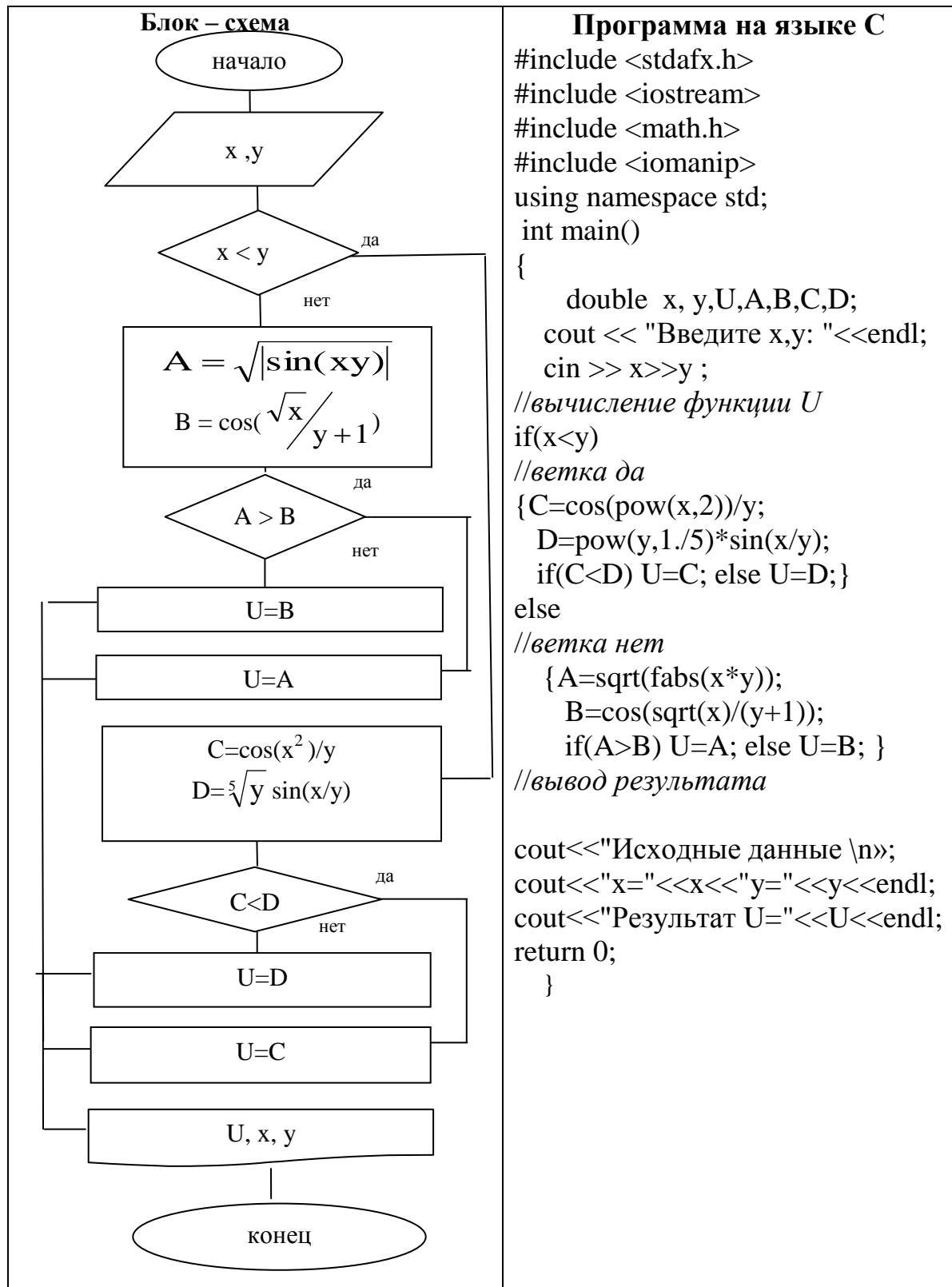


Программа на языке C

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;
int main()
{
    double x, y,U,A,B,max;
    //ввод данных
    cout << "Введите x,y: " << endl;
    cin >> x >> y ;
    //вычисление функции U
    A=x*exp(-y);
    B=pow(x,y)*pow(cos(x),2);
    if(A>B) max=A; else max=B;
    if(max<1) U=sqrt(x)/y;
    else U=exp(3.0)+tan(x-y);
    //вывод исходных данных и
    //результата
    cout<<"Исходные данные \n";
    cout<<"x="<<x<<"y="<<y<<endl;
    cout<<"Результат U="<<U<<endl;
    return 0;
}
```


Вычислить значение функции при $x=0,7$ и $y=0,2$.

$$U = \begin{cases} \min \left\{ \frac{\cos x^2}{y}; \sqrt[5]{y} \sin \frac{x}{y} \right\}, & \text{если } x < y, \\ \max \left\{ \sqrt{|\sin(xy)|}; \cos \frac{\sqrt{x}}{y+1} \right\}, & \text{если } x \geq y. \end{cases}$$



Форматный вывод printf()

Оператор вызова функции printf() имеет следующую структуру:

printf("форматная строка", список аргументов);

Форматная строка включает произвольный текст, управляющие символы, спецификаторы формата. Список аргументов — это перечень выводимых переменных. Количество спецификаторов в «форматной строке» должно соответствовать количеству выводимых переменных.

Управляющие символы:

| | |
|----|--------------------------|
| \n | Переход на новую строку |
| \t | Горизонтальная табуляция |
| \v | Вертикальная табуляции |

Спецификатор формата определяет форму внешнего представления выводимой величины.

| | |
|----|---|
| %c | символ |
| %s | строка |
| %d | целое десятичное число |
| %i | целое десятичное число без знака |
| %f | вещественное число в форме с фиксированной точкой |
| %e | вещественное число в форме с плавающей точкой |

К спецификатору формата могут быть добавлены числовые параметры: ширина поля и точность. Ширина — это число позиций, отводимых на экране для вывода, а точность — число позиций под дробную часть (после точки). Параметры записываются между значком % и символом формата отделяются друг от друга точкой.

Если в пределы указанной ширины поля выводимое значение не помещается, то этот параметр игнорируется, и величина будет выводиться полностью.

Например:

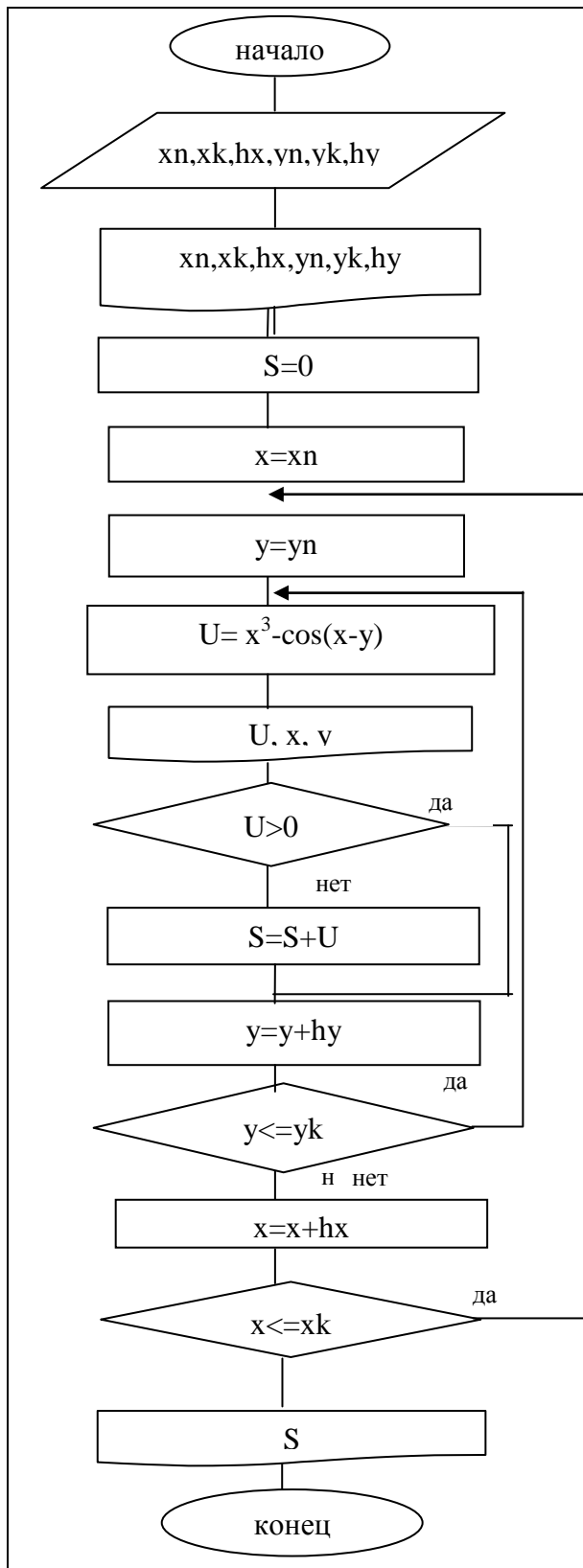
```
printf("\n Я люблю информатику \n ");
```

```
printf("\n %5.2f \t %4.1f\n",x,y);
```

```
printf("\n x=%5.2f \t y=%4.1f\n",x,y);
```

Циклический алгоритм

Вычислить таблицу значений функции $U = x^3 - \cos(x - y)$ при изменении $0 \leq x \leq 1$ с шагом 0,3 и $1 \leq y \leq 2,3$ с шагом 0,6. Вычислить сумму всех положительных значений функции.

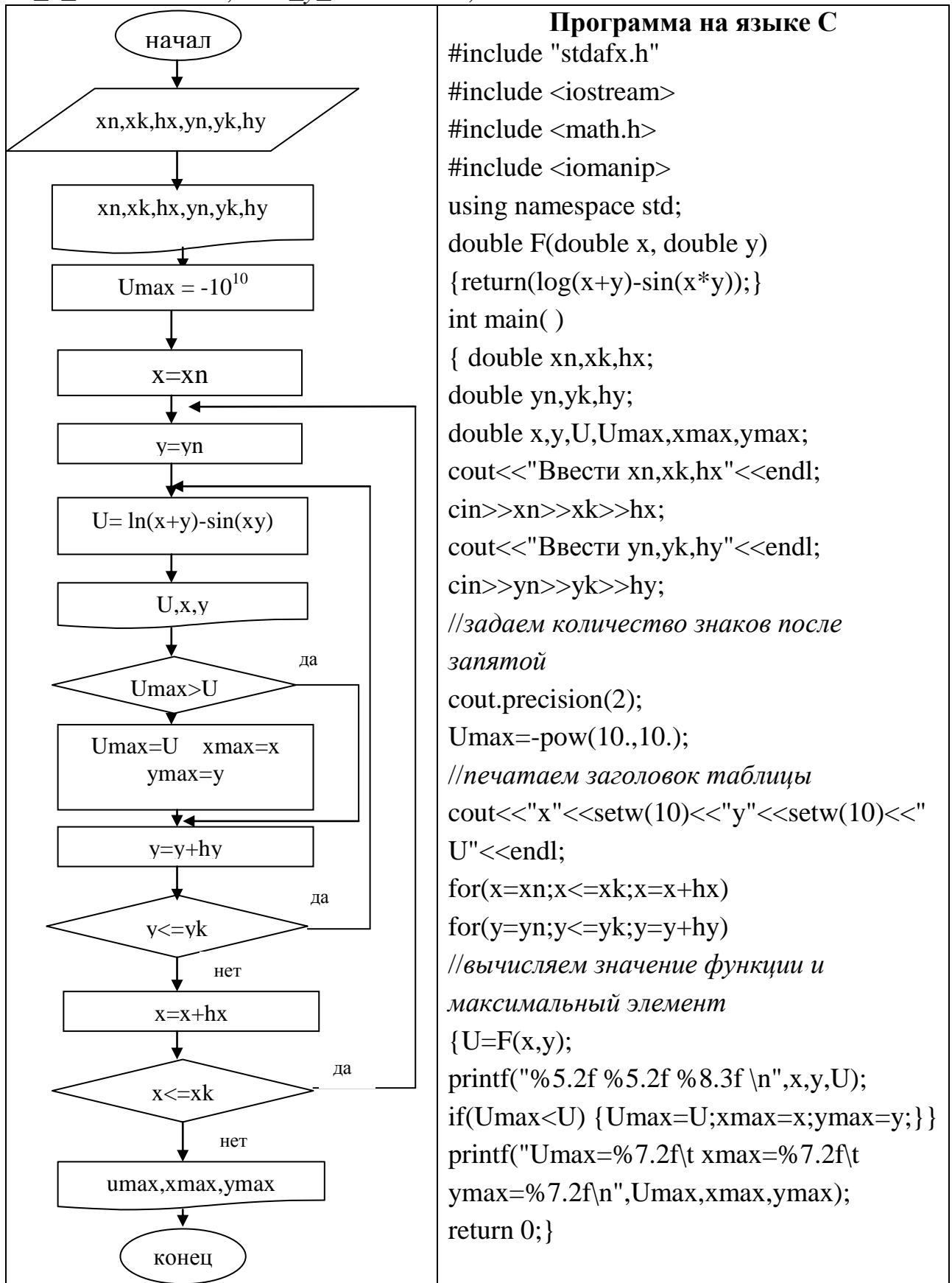


Программа на языке C

```

#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;
int main( )
{ double xn,xk,hx;
double yn,yk,hy;
double x,y,U,S=0;
cout<<"Ввести xn,xk,hx"<<endl;
cin>>xn>>xk>>hx;
cout<<"Ввести yn,yk,hy"<<endl;
cin>>yn>>yk>>hy;
//задаем количество знаков после
//запятой
cout.precision(2);
//печатаем заголовок таблицы, пропуская
// между буквами по 10 пробелов
cout<<"x"<<setw(10)<<"y"<<setw(10)
<<"U"<<endl;
for(x=xn;x<=xk;x=x+hx)
for(y=yn;y<=yk;y=y+hy)
{U=pow(x,3)-cos(x-y);
if(U>0) S=S+U;
//печатаем значения x,y,U
printf("%5.2f %5.2f %8.3f \n",x,y,U);
}
cout<<"Сумма значений функции
"<<S<<endl;
return 0;}
  
```

Вычислить максимальное значение функции $U = \ln(x+y) - \sin(xy)$ при изменении $1 \leq x \leq 2$ с шагом 0,3 и $0 \leq y \leq 1$ с шагом 0,2.



МАССИВЫ

Одномерный массив

Массив – это упорядоченный набор элементов одного типа. Например, последовательность - 0 1 1 2 3 5 8 13 21 представляет собой 9 элементов массива "A". Для того чтобы объявить массив и проинициализировать его данными элементами, нужно написать следующую инструкцию C++:

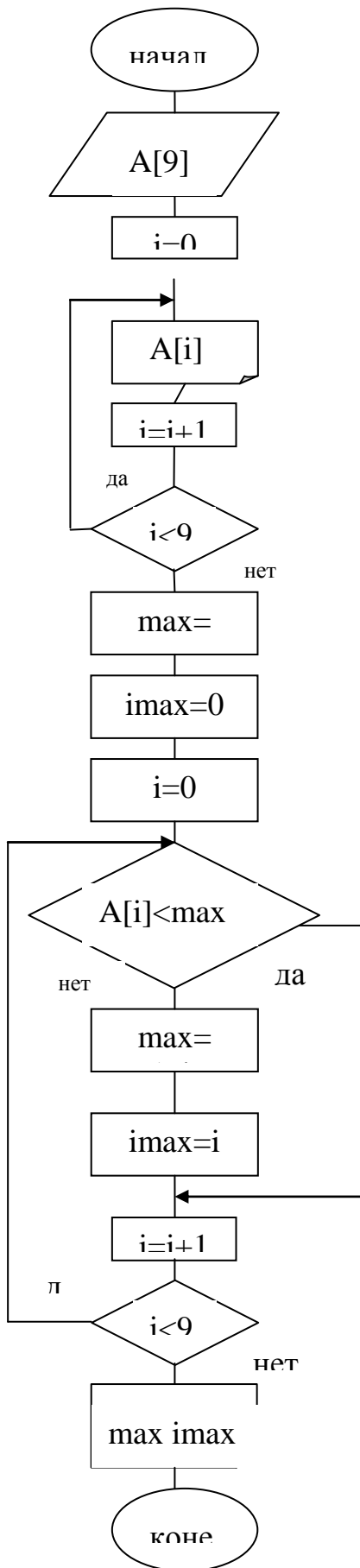
```
int A[9] = { 0, 1, 1, 2, 3, 5, 8, 13, 21 };
```

Здесь "A" – это имя массива. Элементы массива имеют тип `int`, *размер* (длина) массива равна 9. Значение первого элемента – 0, последнего – 21. Для работы с массивом мы *индексируем* (нумеруем) его элементы, а доступ к ним осуществляется с помощью операции *взятия индекса*. Индексация массивов начинается с 0, поэтому индекс первого элемента равен 0. Таким образом, чтобы обратиться к последнему элементу массива, мы должны вычесть единицу из размера массива.

Можно ввести элементы массива в цикле.

| Инструкция: | Блок – схема |
|---|--|
| <pre>#include "stdafx.h" #include "iostream" using namespace std; int main() { int A[9]; for(int i=0; i<9; i++) cin>> A [i]; return 0; }</pre> | <pre>graph TD Start([i=0]) --> Input[/A./] Input --> Process[i=i+1] Process --> Decision{i < 9} Decision --> Input Decision --> End([конец])</pre> |

Задача 1. Найти максимальный элемент массива $A[9]=\{2, 30, 4, 5, 6, -7, 8, 9, -1\}$



```

#include "stdafx.h"
#include <iostream>
#include <iomanip> //заголовочный файл,нужен
                  // для использования функции setw()

using namespace std; //пространство имен

int main()
{
// задание количества элементов массива
const int M=9;

//объявление массива и инициализация его элементов
int A[M]={2, 30, 4, 5, 6,-7, 8, 9,-1};

// вывод массива на экран
for(int i=0;i<M;i++)
cout<<setw(4)<<A[i]; //функция setw(4)
// обеспечивает вывод значения A[i] в 4 позиции
cout<<endl; //перевод строки

// поиск максимального элемента
int max=A[0]; // начальное значение
int imax=0; // начальное местоположение
for(int i=0;i<M;i++)
if(A[i]>max) {max=A[i];imax=i;}

// вывод max и imax
cout<<"\n
max="<<max<<setw(7)<<"imax="<<imax<<"\n";

return 0;
}
  
```

Двумерные массивы – матрицы

Матрица — математический объект, записываемый в виде прямоугольной таблицы чисел. Здесь изображены две матрицы: матрица А и матрица В.

$$A = \begin{vmatrix} 1 & 0 & 2 \\ 3 & -1 & 4 \end{vmatrix} \quad B = \begin{vmatrix} 3 & 5 & -1 \\ 2 & 7 & 0 \\ 4 & 3 & 3 \end{vmatrix}$$

Размерность 2x3

Размерность 3x3

Матрица А - это матрица 2x3 (то есть у нее две строки и три столбца), тогда как матрица В - это матрица 3x3. Можно получить доступ к элементу матрицы А, используя запись $A[m][n]$, где m - это строка, а n - столбец. Элемент в верхнем углу матрицы А будет обозначаться $A[0,0]$, и он равен единице.

Создание матриц (двумерных массивов)

Создание возможно в двух вариантах: с инициализацией и без инициализации.

1. Для создания и инициализации матрицы нужно указать её размерности и значения элементов:

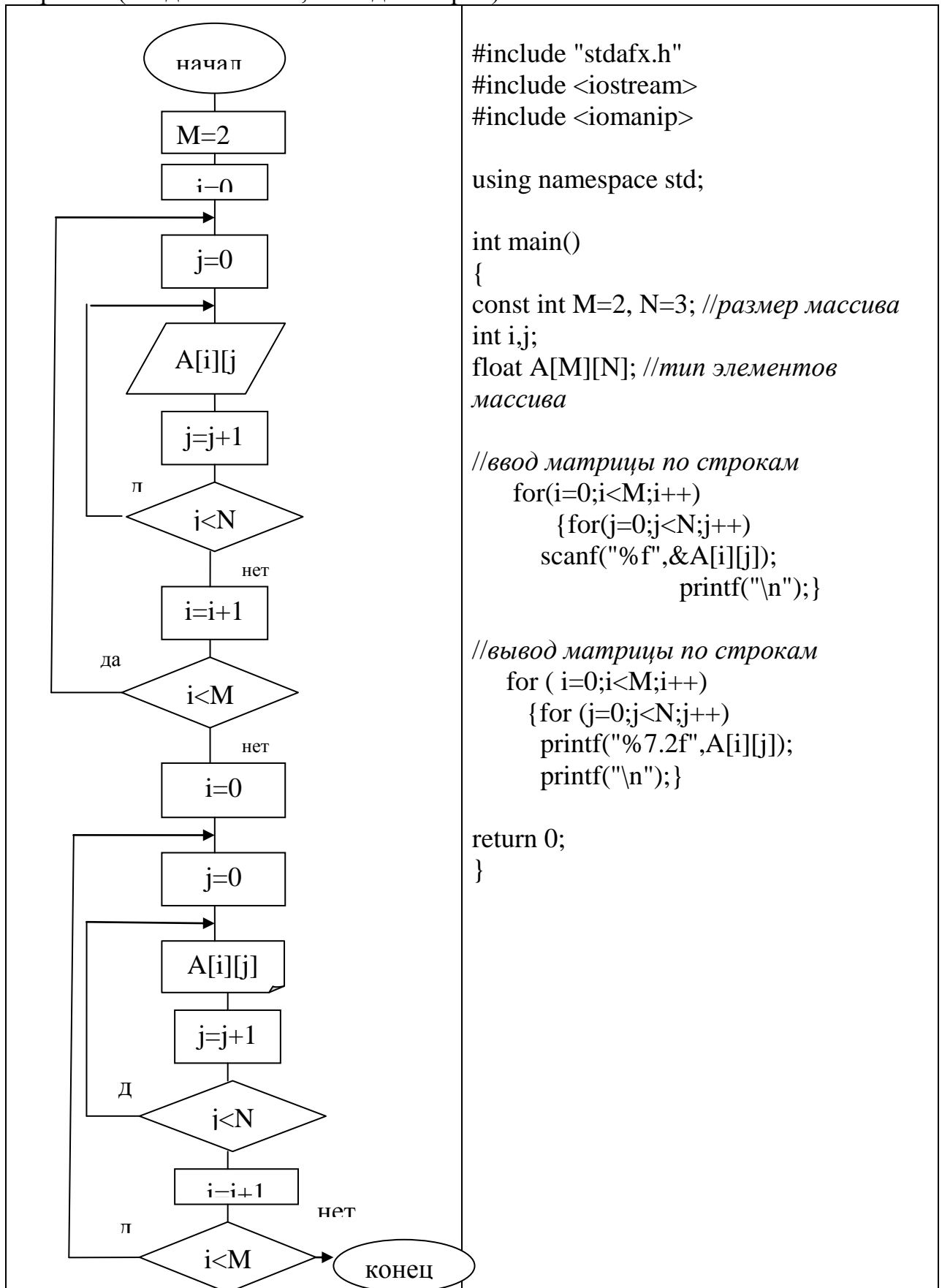
```
int A[2][3]; // объявление двумерного массива
```

```
A[2][3]={ {1,0,2}, {3,-1,4} }; // заданы значения каждого элемента
```

2. Можно создать матрицу с помощью генератора случайных чисел `rand()`:

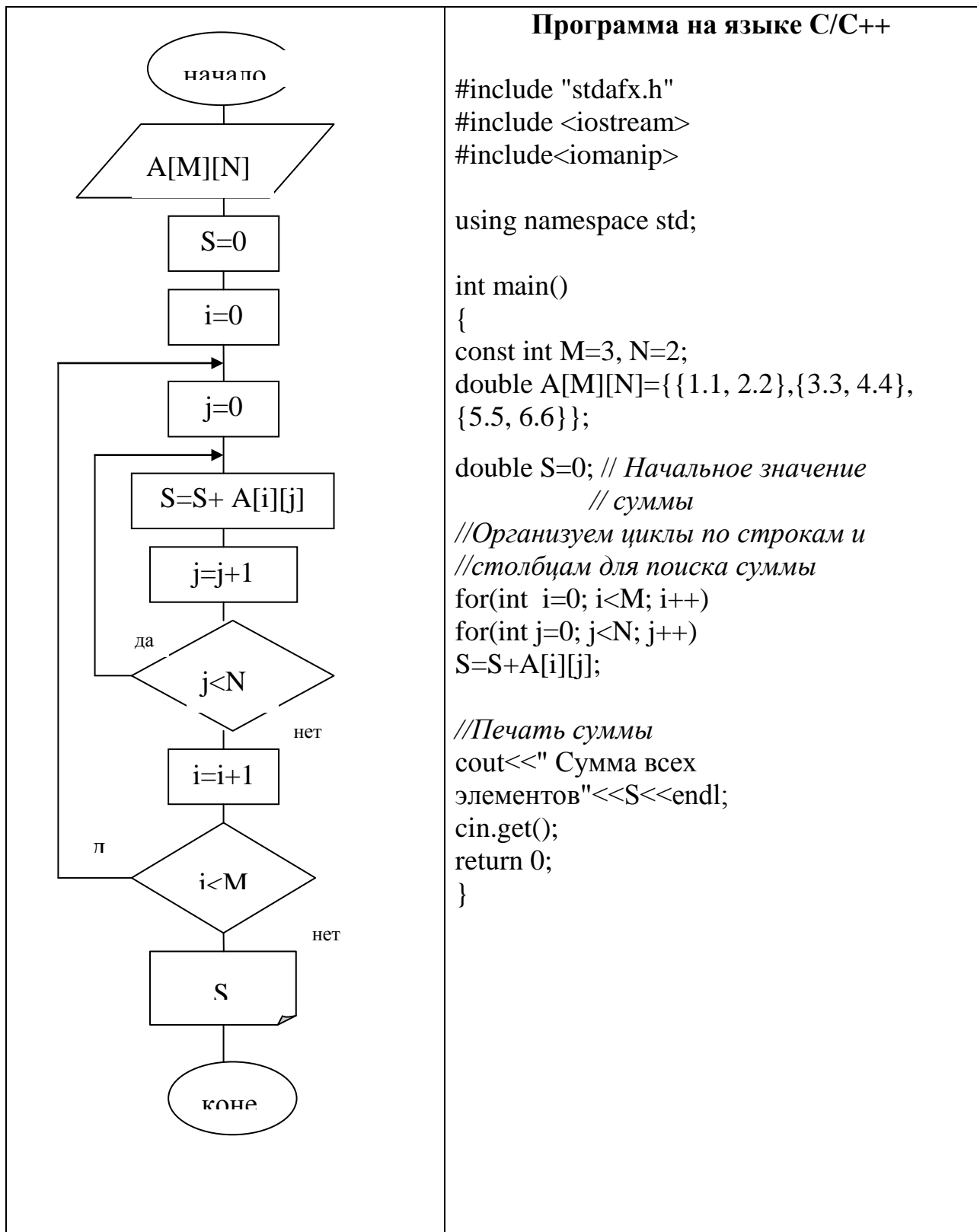
```
#include "stdafx.h"
#include <iomanip>
#include <time.h>
using namespace std;
void main()
{
    int A[2][3];
    srand(time(NULL));
    //Заполнение массива А
    for( int i=0;i<2;i++)
        for(int j=0; j<3; j++)
            A[i][j] = rand()%20; // Заполнение массива А числами от 0 до 20
    cout<<"\n \n";
}
```

3. Форматный ввод - вывод элементов двумерного массива (матрицы) по строкам (ввод с консоли, вывод на экран)



Вычисление суммы элементов матрицы

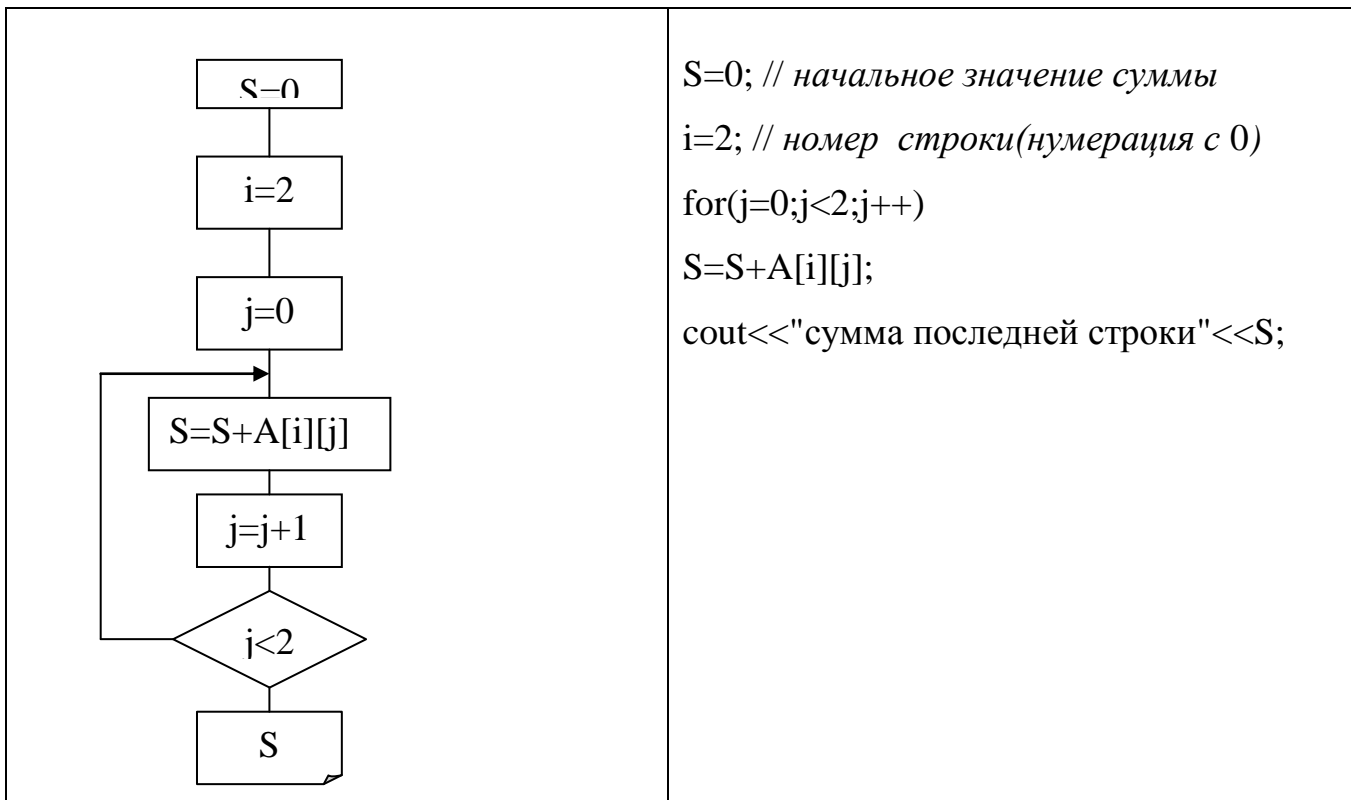
Задача 1. Найти сумму всех элементов матрицы размером 3x2



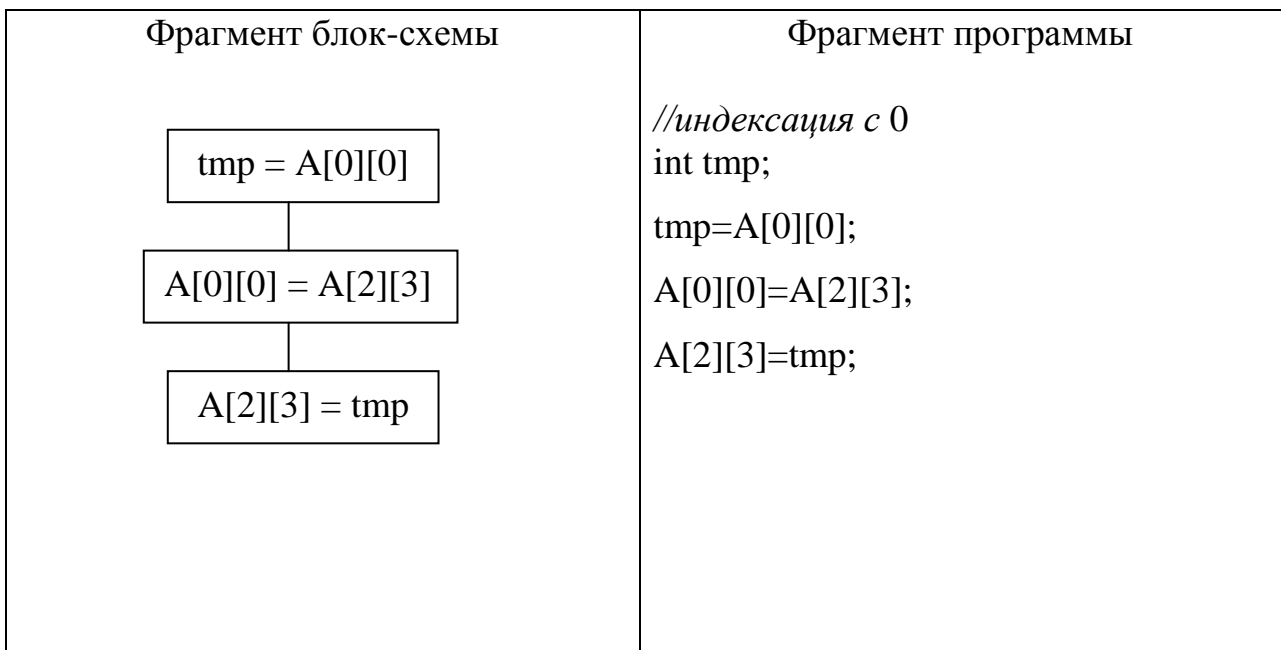
Задача 2. Найти сумму элементов последней (третьей) строки матрицы размером 3x2.

Фрагмент блок схемы

Фрагмент программы



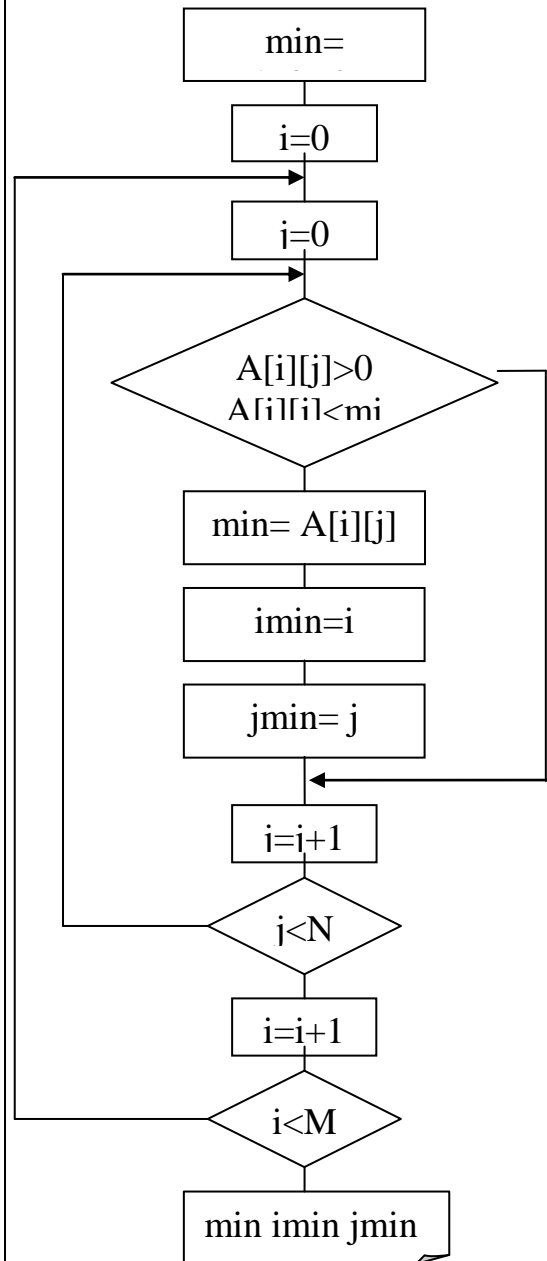
Задача 3. В массиве размером 3x4 поменять местами первый и последний элементы



Поиск максимума (минимума) двумерного массива (матрицы)

Задача 4. Найти минимальный элемент, среди положительных элементов матрицы

Фрагмент блок-схемы



Фрагмент программы

```

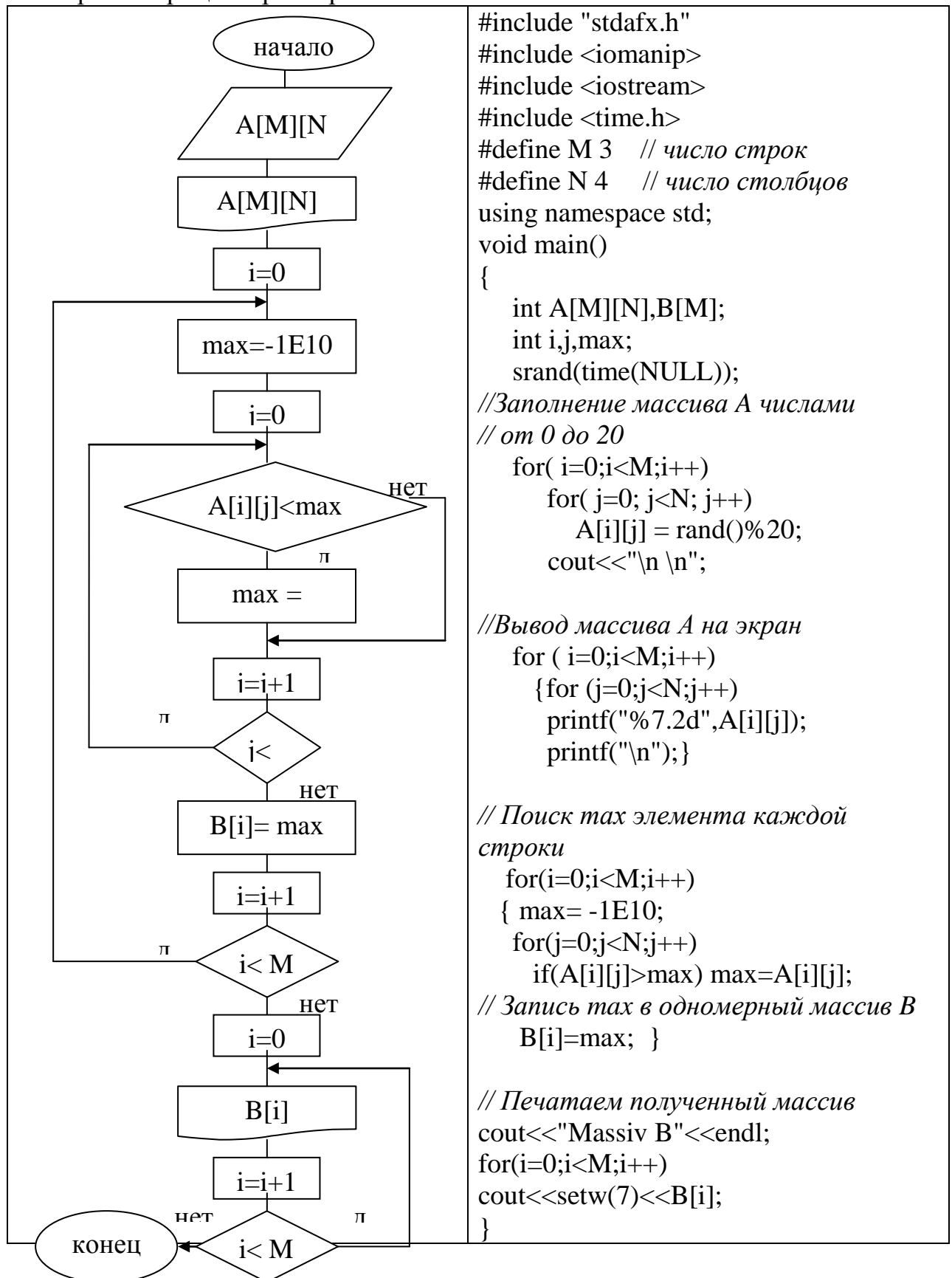
int min=A[0][0]; // начальное значение
                // min
// Поиск минимального элемента из
// положительных

for(i=0;i<M;i++)
  for(j=0;j<N;j++)
    if((A[i][j]>0)&&(A[i][j]<min))
      {min=A[i][j]; imin=i; jmin=j;}

// Вывод значения минимального элемента
cout<<" минимальный элемент ="<<min;
cout <<" imin="<<imin<<"jmin="<<jmin;
    
```

Формирование нового массива

Задача 5 Сформировать одномерный массив из максимальных элементов строк матрицы A размером 3x4



Квадратная матрица

Если количество строк матрицы равно количеству столбцов, то такая матрица называется квадратной.

Свойства элементов квадратной матрицы:

| | |
|--|------------------------------------|
| элементы лежат на главной диагонали | $i=j$ |
| элементы лежат ниже главной диагонали | $i>j$ |
| элементы лежат выше главной диагонали | $i<j$ |
| элементы лежат на побочной диагонали | $i+j=N-1$ (N – порядок матрицы) |
| элементы лежат ниже побочной диагонали | $i+j>N-1$ |
| элементы лежат выше побочной диагонали | $i+j<N-1$ |

Задача 6 Найти произведение не равных нулю элементов матрицы размером 4x4 ($M=4, N=4$), лежащих ниже главной диагонали.

| Фрагмент блок-схемы | Фрагмент программы |
|---|--|
| <pre> graph TD Start([P=1]) --> I0[i=0] I0 --> J0[i=0] J0 --> Cond1{i > j A[i][j] != 0} Cond1 -- Yes --> Calc[P = P * A[i][j]] Calc --> Iinc[i=i+1] Iinc --> Cond2{j < N} Cond2 -- Yes --> J0 Cond2 -- No --> Iinc2[i=i+1] Iinc2 --> Cond3{i < M} Cond3 -- Yes --> J0 Cond3 -- No --> End([P]) </pre> | <pre> // начальное значение произведения // элементов ниже главной диагонали double P=1; // цикл поиска произведения P for(int i=0; i<M; i++) for(int j=0; j<N; j++) if((i>j)&&(A[i][j]!=0)) P=P*A[i][j]; // печатаем произведение cout<<"P="<<P; </pre> |

Функции и массивы

Задача 2. Вычислить количество положительных элементов массива А.

| без функции | с функцией |
|---|---|
| <pre> #include "stdafx.h" #include <iostream> #include <iomanip> using namespace std; int main() { double A[3][2]={{ 1,-2},{3,4},{5,-6}}; int i,j; for(i=0;i<3;i++) { for(j=0;j<2;j++) cout<<setw(7)<<A[i][j]; cout<<endl; } int k1=0; for(i=0;i<3;i++) for(j=0;j<2;j++) if(A[i][j]>0) k1++; cout<<"количество в A=" "<<k1<<endl; char ch; cin>>ch; return 0;} </pre> | <pre> #include "stdafx.h" #include <iostream> #include <iomanip> using namespace std; //Функция поиска количества //положительных элементов int f_kol(double *X, int M, int N) { int k=0; for(int i=0;i<M;i++) for(int j=0;j<N;j++) if(*(X+N*i+j)>0) k++; return k;} //Функция вывода элементов матрицы //A по строкам void print(double *X, int M, int N) { for(int i=0;i<M;i++) { for(int j=0;j<N;j++) cout<<setw(7)<<*(X+N*i+j); cout<<endl;} } //Основная функция int main() { double A[3][2]={{ 1,-2},{3,4},{5,-6}}; //Обращение к функции вывода //матрицы A print(*A,3,2); //Обращение к функции поиска //количества положительных //элементов cout<<"количество в A="<< f_kol(*A,3,2)<<endl; cin.get(); return 0; } </pre> |

Задача 3. Вычислить среднее арифметическое значение элементов матрицы

| без функции | с функцией |
|---|---|
| <pre> #include "stdafx.h" #include <iomanip> </pre> | <pre> #include "stdafx.h" #include <iomanip> </pre> |

| | |
|--|--|
| <pre>#include <iostream> using namespace std; int main() { const int M=3, N=2; double A[M][N]={{ 1,2},{ 3,4},{ 5,6 }}; double S=0, Srednee; for(int i=0;i<M;i++) for(int j=0;j<N;j++) S+=A[i][j]; Srednee=S/(M*N); cout<<"среднее="<<Srednee; cin.get(); return 0;}</pre> | <pre>#include <iostream> using namespace std; double f_srednee(double **X, int M, int N) { double S=0, Srednee; for(int i=0;i<M;i++) for(int j=0;j<N;j++) S+=X[i][j]; Srednee=S/(M*N); return Srednee; } int main() { double A[3][2]={{ 1,2},{ 3,4},{ 5,6 }}; // вспомогательный массив указателей double *pA[3]; for(int i=0;i<3;i++) pA[i]=(double*)&A[i]; cout<<"srednearifm=" <<f_srednee(pA,3,2)<< endl; cin.get(); return 0;}</pre> |
|--|--|

Задача 4. Определить максимальный элемент каждой строки матрицы

| без функции | с функцией |
|--|---|
| <pre>#include "stdafx.h" #include <iomanip> #include <iostream> using namespace std; int main() { const int M=3, N=2; double A[M][N]={ 1,2,3,4,5,6 }; double max; int i,j,jmax; for(i=0;i<M;i++) { max=A[i][0]; jmax=0; for(j=0;j<N;j++) if(A[i][j]>max) { max=A[i][j]; jmax=j; } cout<<"max"<<i<<"строки="<<max; cout<<"в столбце"<<jmax<<endl; } return 0;}</pre> | <pre>[Заголовочные файлы] const int M=3, N=2; int max_stroka(double X[M][N], int stroka) { double max=X[stroka][0]; int jmax=0; for(int j=0;j<N;j++) if(X[stroka][j]>max) { max=X[stroka][j]; jmax=j; } return jmax; } int main() { double A[M][N]={ 1,2,3,4,5,6 }; int jmax; for(int i=0;i<M;i++) { jmax=max_stroka(A,i); cout<<"max"<<i<<"строки="<<A[i][jmax]; cout<<" в столбце"<<jmax<<endl; } return 0;}</pre> |

Ввод-вывод элементов многомерного массива в файл

Форматный ввод-вывод

Для организации в программе ввода-вывода в файл необходимо:

1. Объявить указатель на файл

- FILE *имя указателя на 1-й файл, *имя указателя на 2-й файл, ...;**
- Открыть файл для записи-чтения функцией
fopen("имя файла", "режим обработки файла");
Режимы обработки файлов:
"r" - файл открывается для чтения;
"w" - файл открывается для записи. Если он существует, его содержимое пропадает.
"a" - файл открывается для записи. Если файл не существует, он создается.
"r+" - существующий файл открывается для чтения и для записи.
"w+" - создается файл для чтения и записи;
"a+" - файл открывается для чтения и записи в конец файла.
 - Выполнить чтение из файла функцией
fscanf(имя указателя на файл, список ввода);
 - Выполнить запись в файл функцией
fprintf(имя указателя на файл, список вывода);
 - Закрыть файл функцией
fclose(имя указателя на файл);

Потоковый ввод-вывод из файла.

Файловый ввод/вывод аналогичен стандартному вводу/выводу, единственное отличие – это то, что ввод/вывод выполнятся не на экран, а в файл. Для работы с файлом нужно создать объект класса `ifstream` (если бы нужно было считать данные из файла), `ofstream` (если нужно сделать запись в файл).

ifstream имя_объекта, **ofstream** имя_объекта.

- Открыть файл для записи-чтения функцией
имя_объекта.open("имя файла", "режим открытия файла");
Режимы открытия файла
ios_base::in - открыть файл для чтения
ios_base::out - открыть файл для записи
ios_base::ate - при открытии переместить указатель в конец файла
ios_base::app - открыть файл для записи в конец файла
ios_base::trunc - удалить содержимое файла, если он существует
ios_base::binary - открытие файла в двоичном режиме
- Выполнить чтение из файла функцией
имя_объекта>> список ввода;
- Выполнить запись в файл функцией
имя_объекта <<список вывода;
- Закрыть файл функцией
имя_объекта.close();

Задача Записать массив 3x3 (3 строки, 3 столбца) в файл и прочитать его из файла.

```
#include "stdafx.h"  
#include <stdio.h>
```



```

#define n 3
int main()
{
    FILE *f1,*f2;
    int i,j;
    int a[n][n]={ { 1,3,5 },{ -2,-4,-6 },{ 0,10,8 } };
    //открываем файл для записи
    f1=fopen("massiv1.txt ","a");
    //выводим массив "a" в файл massiv1.txt
    for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
      fprintf(f1,"%5d",a[i][j]);
      fprintf(f1,"\n");}
    //закрываем файл
    fclose(f1);
    //открываем файл для чтения
    //при этом нужно проверить, что он существует
    f2=fopen("massiv1.txt ","r");
    if(f2==NULL){printf("ошибка открытия файла"); return 0;} /*выход из
    программы, если файл не найден*/
    else
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                fscanf(f2,"%d",&a[i][j]);
    fclose(f2);
    //выведем на экран то, что прочитали из файла
    for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
      printf("%5d",a[i][j]);
      printf("\n");}
    getchar();
    return 0;
}

```

Экран результата

```

c:\Users\tom\Desktop\Программ
 1      3      5
-2     -4     -6
 0     10     8

```

Задача Записать массив 3x3 (3 строки, 3 столбца) в файл и прочитать его из файла.

```
#include "stdafx.h"
#include<fstream>
#include<iostream>
#include<iomanip>
#define n 3
int main()
{ ofstream outFile;
//открываем файл для записи
outFile.open("d:\\massiv1.txt ");
    int a[n][n]={ {1,3,5},{-2,-4,-6},{0,10,8}};
    int i,j;
//выводим массив "a" в файл massiv1.txt
outFile.precision(2);
    for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        outFile <<setw(5)<<a[i][j]);
        outFile <<endl;}
//закрываем файл
    outFile.close();
//открываем файл для чтения
//при этом нужно проверить, что он существует
    ifstream inFile;
    if(!inFile.is_open())
    {cout<<"ошибка открытия файла";
return 0;} /*выход из программы, если файл не найден*/
    else
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                inFile>>a[i][j]);
    inFile.close();
//выведем на экран то, что прочитали из файла
    for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        cout<<a[i][j]);
        cout<<endl;}
    getch();
return 0;
}
```

Список литературы

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня: учебник для вузов./ Т.А. Павловская СПб.: «Питер», 2007.
2. Павловская Т.А. С/С++. Структурное программирование: практикум./ Т.А. Павловская, Ю.В. Щупак СПб.: «Питер», 2007.
3. Полубенцева М.И. С/С++. Процедурное программирование./ М.И. Полубенцева СПб.: ВНУ-Санкт-Петербург, 2008.
4. Подбельский В.В. Программирование на языке Си./ В.В.Подбельский, С.С. Фомин М.:Изд.– во «Финансы и статистика», 2007.
5. Хабибулин И.В. Программирование на языке высокого уровня С/С++./ И.В.Хабибулин, СПб.: ВНУ-Санкт-Петербург, 2006.
6. Шелест В.И. Программирование./ В.И.Шелест, СПб.: ВНУ-Санкт-Петербург, 2001.