11 Решение задач

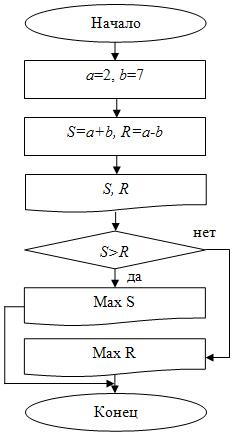
**Программная реализация разветвляющегося алгоритма**

При рассмотрении разветвляющегося алгоритма будет разобран набор задач и примеров, которые являются усложненными версиями уже разобранных выше задач.

Будут приведены примеры вычисления функций, заданных на разных промежутках кусочно. Также приведены принцип и примеры расчетов для кусочных функций, заданных на двух промежутках и на трех промежутках.

Пример 5. Даны числа a=2, b=7. Вычислить сумму S и разность R чисел a и b. Сравнить полученные значения S и R и указать большее из них.

Блок-схема представлена на [рис. 4.5](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.5).



**Рис. 4.5.**Блок-схема для примера 5

Код программы (Visual Studio):

// proga14.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

double a, b, S, R;

a=2;

b=7;

S=a+b;

R=a-b;

cout<<"S="<<S<<" R="<<R<<endl;

if(S>R) cout<<"Max S"<<endl;

else cout<<"Max R"<<endl;

return 0;

}

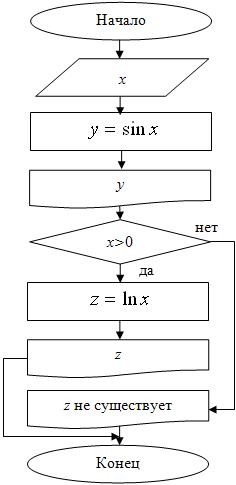
Результат выполнения программы (Visual Studio):



Пример 6. Составить блок-схему и программу для вычисления значений функций y=\sin x и z=\ln x при различных значениях аргумента. Учесть область определения функций.

Решение. Рассмотрим области определения функций y и z. Синус имеет неограниченную область определения, т.е. считается при любом действительном х. Поэтому на расчет функции y не будет никаких условий. Логарифм же существует при положительном значении аргумента, т.е. считается при x>0. Поэтому, прежде чем считать функцию z, необходимо проверить условие "x>0". Если условие верно, то можно считать z. Если ложно, то z считать нельзя, и в этом случае мы напишем в ответ, что z не существует фразой "z net".

Блок-схема представлена на [рис. 4.6](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.6).



**Рис. 4.6.**Блок-схема для примера 6

Код программы (Visual Studio):

// proga15.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

double x, y, z;

cout<<"vvedi x=";

cin>>x;

y=sin(x);

cout<<"y="<<y;

if(x>0){

z=log(x);

cout<<" z="<<z<<endl;

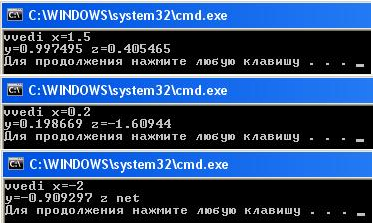
}

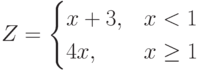
else cout<<" z net"<<endl;

return 0;

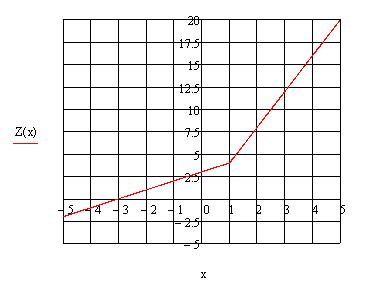
}

Результат выполнения программы:



Пример 7. Вычислить значение функции  при различных значениях аргумента х (аргумент х считать с клавиатуры).

Решение: Функция Z задана как кусочная функция, т.е. при разных значениях х она считается по-разному. Следует отметить, что при одном значении аргумента функция будет иметь лишь одно значение, т.е. функция однозначная. График функции Z представлен на [рис. 4.7](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.7).



**Рис. 4.7.**График кусочной функции

Посчитаем значение функции Z при х=0. Прежде всего необходимо проверить, какое условие верно: "х<1" или "х\ge 1". Условие для первой ветки функции верно, значит считаем Z=x+3=0+3=3.

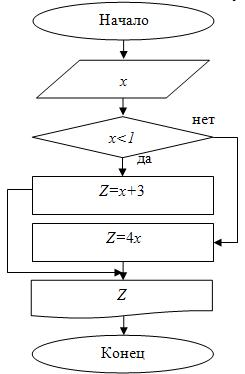
Посчитаем значение функции Z при х=4. Прежде всего необходимо проверить, какое условие верно: "х<1" или "х\ge 1". Условие для второй ветки функции верно, значит считаем Z=4x=4\cdot 4=16.

При расчете кусочной функции так или иначе сначала проводится проверка условия, поэтому вычисление таких функций реализуется через разветвляющийся алгоритм. В данном примере будет один блок разветвления (соответственно один оператор if в программе). Условием будет служить проверка "х<1". Если условие верно, то функция считается по первой ветке, если ложно, то по второй ветке. Второе условие "х\ge 1" будет лишним, т.к. оно соответствует стрелке "нет" от первого условия "х<1".

Примечание. Функция Z – это функция, которая имеет один аргумент. Не следует путать запись кусочной функции с системой уравнений или системой неравенств, уравнения или неравенства которых также объединены фигурной скобкой. При расчете кусочной функции не решается никакая система уравнений, система неравенств, а просто подставляется конкретное значение аргумента и вычисляется значение функции.

Примечание. При расчете функции Z не следует сразу рассчитывать обе ветки функции, т.к. в ответ нам нужен только один результат. Корректно составленный алгоритм не должен быть избыточен. Не следует повторять одно и то же по сути условие.

Блок-схема для примера 7 представлена на [рис. 4.8](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.8).



**Рис. 4.8.**Блок-схема для примера 7

Код программы (Visual Studio):

// proga16.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main(){

double x, Z;

cout<<"x=";

cin>>x;

if(x<1){

Z=x+3;

}

else {

Z=4\*x;

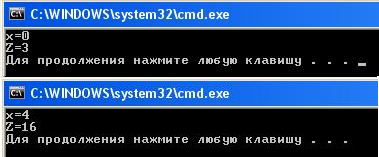
}

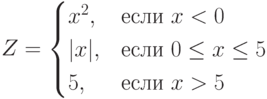
cout<<"Z="<<Z<<endl;

return 0;

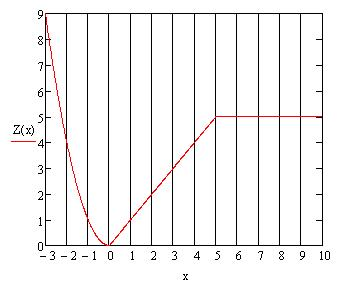
}

Результат выполнения программы:



Пример 8. Вычислить значение функции  при различных значениях аргумента х (аргумент х считать с клавиатуры).

Решение: Функция Z задана как кусочная функция, определенная на трех промежутках. График функции Z представлен на [рис. 4.9](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.9).



**Рис. 4.9.**График кусочной функции

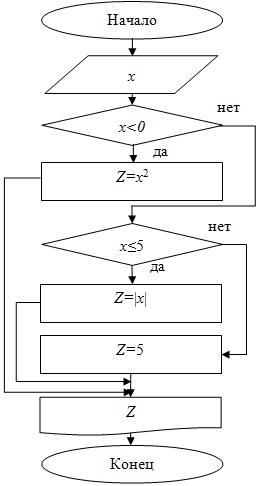
В данной задаче будем использовать два условия для определения нужной ветви функции. Первое условие "x<0" определяет, следует ли рассчитывать по первой ветви функции. Второе условие "x\le 5" определяет, следует ли рассчитывать по второй ветви. Если и первое, и второе условия оказались ложными (не выполнились), то следует рассчитывать по третьей ветви. Следовательно, в блок-схеме будем использовать два блока разветвления, а в программе два оператора условия if.

Например, нужно вычислить Z при х=-2. Первое условие "x<0" выполняется верно, т.к. -2<0. Следовательно, считать следует по первой ветви, а именно: Z=x^2=(-2)^2=4.

Например, нужно вычислить Z при х=2,5. Первое условие "x<0" выполняется неверно, т.к. 2,5>0. Проверяем второе условие "x\le5": 2,5≤5 – верно. Следовательно, считать следует по второй ветви, а именно: Z=|x|=|2,5|=2,5.

Например, нужно вычислить Z при х=10. Первое условие "x<0" выполняется неверно, т.к. 10>0. Проверяем второе условие "x\le 5": 10≤5 – неверно. Следовательно, ни первое, ни второе условия не выполнились, и считать следует по третьей ветви, а именно: Z=5.

Блок-схема представлена на [рис. 4.10](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.10).



**Рис. 4.10.**Блок-схема для примера 8

Код программы (Visual Studio):

// proga17.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <math.h>

using namespace std;

int main(){

double x,Z;

cout<<"x=";

cin>>x;

if(x<0){ Z=pow(x, 2.0);

}

else {if(x<=5){

Z=fabs(x);}

else {

Z=5.0;}

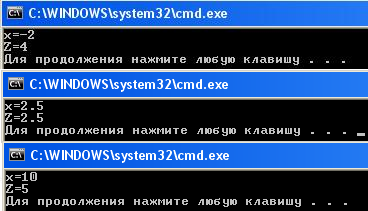
}

cout<<"Z="<<Z<<endl;

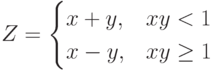
return 0;

}

Результат выполнения программы (Visual Studio):

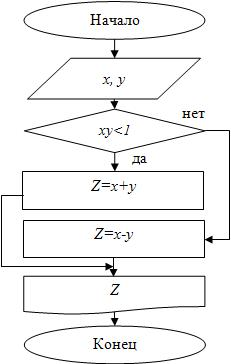


Примечание. В качестве второго условия можно, конечно, взять "0\le x\le 5". Но при проверке первого условия мы уже знаем, будет ли x<0. Поэтому можно пропустить повторную проверку на положительность переменной x. Также в качестве второго условия можно взять условие "x>5". Если оно выполнится верно, то считать следует по третьей ветви, если неверно – то по второй.

Пример 9. Вычислить значение функции  при различных значениях аргументов х и у (аргументы х и у считать с клавиатуры).

Решение: Данный Пример аналогичен Примеру 7. Отличие состоит в том, что сейчас функция Z зависит от двух переменных x и y. Для реализации алгоритма разницы нет, поэтому за основу возьмем блок-схему и программу из Примера 7.

Блок-схема представлена на [рис. 4.11](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.11).



**Рис. 4.11.**Блок-схема для примера 9

Код программы (Visual Studio):

// proga18.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main(){

double x, y, Z;

cout<<"x=";

cin>>x;

cout<<"y=";

cin>>y;

if(x\*y<1){ Z=x+y;

}

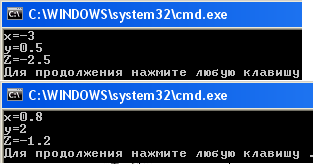
else {Z=x-y;

}

cout<<"Z="<<Z<<endl;

return 0;}

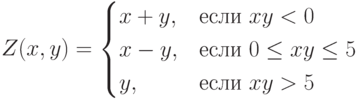
Результат выполнения программы (Visual Studio):



Ручной счет.

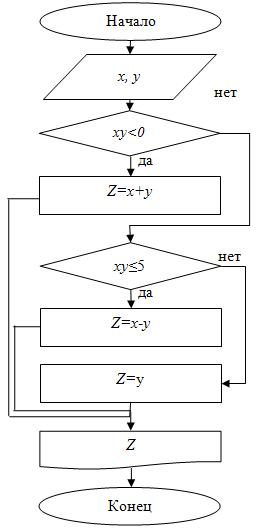
при х=-3; у=0,5 ху=-3·0,5=-1,5 <1 Z=x+y=-3+0,5=-2,5.

при x=0,8; y=2 xy=0,8·2=1,6≥1 Z=x-y=0,8-2=-1,2.

Пример 10. Вычислить значение функции  при различных значениях аргументов х и у (аргументы х и у считать с клавиатуры).

Решение. В данном примере функция Z определена на трех промежутках, как в Примере 8. Отличие состоит в том, что Z зависит от двух переменных x и y. Для реализации алгоритма разницы нет, поэтому за основу возьмем блок-схему и программу из Примера 8.

Блок-схема представлена на [рис. 4.12](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.12).



**Рис. 4.12.**Блок-схема для примера 10

Код программы (Visual Studio):

// proga19.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main(){

double x, y, Z;

cout<<"x=";

cin>>x;

cout<<"y=";

cin>>y;

if(x\*y<0){ Z=x+y;}

else {

if(x\*y<=5){

Z=x-y;

}

else {

Z=y;

}

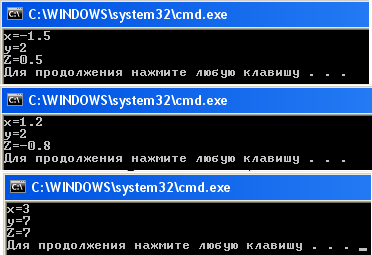
}

cout<<"Z="<<Z<<endl;

return 0;

}

Результат выполнения программы (Visual Studio):



Ручной счет:

при х=-1,5; у=2 ху=-1,5·2=-3 <0 Z={по первой ветви}=x+y=-1,5+2=0,5.

при x=1,2; y=2 xy=1,2·2=2,4≤5 Z={по второй ветви}=x-y=1,2-2=-0,8.

при x=3; y=7 xy=3·7=21 >5 Z={по третьей ветви}=y=7.

Пример 11. Найти наибольшее (максимальное) из трех чисел.

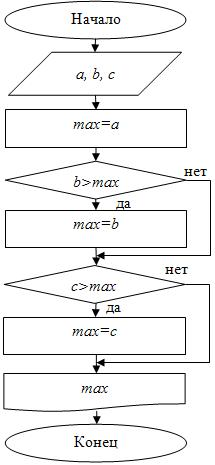
Решение. Обозначим три числа буквами a, b, c. Числа a, b, c будем считывать с клавиатуры. Для нахождения наибольшего из трех чисел введем еще одну переменную max, которая будет хранить наибольшее значение. Принцип нахождения наибольшего значения заключается в следующем. Сначала принимаем max как первое число а. Потом поочередно сравниваем с оставшимися числами b и c. Если окажется, что b больше уже запомненного max, то следует перезаписать max, изменив его значение на значение b. Если же b<max, то max изменять не нужно, поэтому ветвь "нет" будет пустой. Аналогично сравниваем с и max. После двух сравнений в переменной max будет лежать наибольшее из трех чисел.

Примечание. Если чисел будет больше трех, например, пять чисел, то следует сделать четыре сравнения.

Примечание. Если нужно определить наименьшее из трех чисел, то вводят переменную min, приравнивают к первому числу. Затем поочередно сравнивают оставшиеся числа: "b<min", "c<min".

Если нужно определить наименьшее из большего числа чисел, например, из пяти чисел, то следует сделать четыре сравнения.

Блок-схема представлена на [рис. 4.13](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.13).



**Рис. 4.13.**Блок-схема для примера 11

Выполнение блок-схемы (при вводимых значениях чисел 23,7; -0,4; 34,28)

Начало

ввод данных a=23,7; b=-0,4; c=34,28

|  |
| --- |
| Экран |
| a=23,7  b=-0,4  c=34,28 |

max=a=23,7

b>max -0,4>23,7 нет

c>max 34,28>23,7 да

max=c=34,28

вывод max

|  |
| --- |
| Экран |
| a=23,7  b=-0,4  c=34,28  max=34,28 |

Конец.

Код программы (Visual Studio):

// proga20.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

double a, b, c, max;

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

cout<<"c=";

cin>>c;

max=a;

if(b>max){

max=b;

}

if(c>max){

max=c;

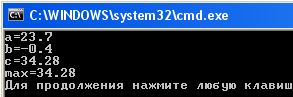
}

cout<<"max="<<max<<endl;

return 0;

}

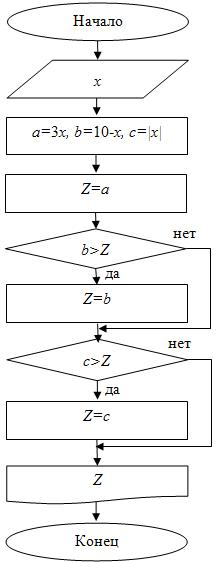
Результат выполнения программы (Visual Studio):



Пример 12. Найти значение функции Z(x)=\max\{3x; 10-x; |x|\} при различных значениях аргумента х (аргумент х считать с клавиатуры).

Решение. Функция Z считается как наибольшее из трех выражений. Задача схожа с Примером 11. Введем обозначения: a=3x, b=10-x, c=|x|. Тогда наша задача максимально будет похожа на Пример 11.

Блок-схема представлена на [рис. 4.14](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.14).



**Рис. 4.14.**Блок-схема для примера 12

Выполнение блок-схемы (при вводимом x=1)

Начало

ввод x=1

|  |
| --- |
| Экран |
| x=1 |

a=3x=3·1=3

b=10-x=10-1=9

c=|x|=|1|=1

Z=a=3

b>Z 9>3 да

Z=b=9

c>Z 1>9 нет

вывод Z

|  |
| --- |
| Экран |
| x=1  Z=9 |

Конец.

Код программы (Visual Studio):

// proga21.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <math.h>

using namespace std;

int main(){

double x, a, b, c, Z;

cout<<"x=";

cin>>x;

a=3.0\*x;

b=10.0-x;

c=fabs(x);

Z=a;

if(b>Z){Z=b;}

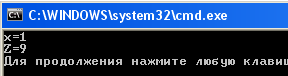
if(c>Z){Z=c;}

cout<<"Z="<<Z<<endl;

return 0;

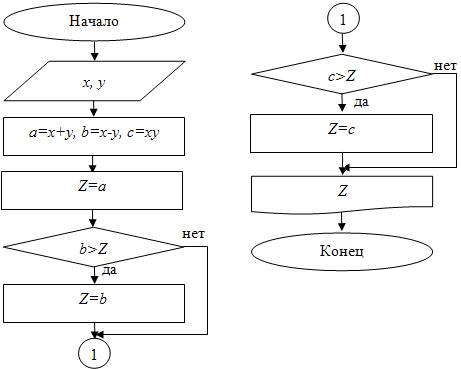
}

Результат выполнения программы (Visual Studio):



Пример 13. Найти значение функции, заданной по формуле Z(x,y)=\max\{x+y, x-y, xy\} при различных значениях аргументов х и у (аргументы х и у считать с клавиатуры).

Блок-схема представлена на [рис. 4.15](https://intuit.ru/studies/courses/16740/1301/lecture/25627?page=2#image.4.15).



**Рис. 4.15.**Блок-схема для примера 13

Код программы (Visual Studio):

// proga22.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

double x, y, a, b, c, Z;

cout<<"x=";

cin>>x;

cout<<"y=";

cin>>y;

a=x+y;

b=x-y;

c=x\*y;

Z=a;

if(b>Z){

Z=b;

}

if(c>Z){

Z=c;

}

cout<<"Z="<<Z<<endl;

return 0;

}

Результат выполнения программы (Visual Studio):

