**МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**ЯЗЫКИ ПРОГРАММИРОВАНИЯ**

Отчет по выполнению лабораторной работы № 6

Вариант №21

Выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр.230711 Павлова Виктория Сергеевна

Проверила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. каф. ИБ Басалова Галина Валерьевна

Тула 2022

# Лабораторная работа №6. Управляющие операторы в языке C++. Часть 1

## Цель работы:

Изучить возможности использования простейших управляющих конструкций в программах на языке С++.

## Задание на работу:

Разработать самостоятельно приложения для решения трех задач по своему варианту. Для задач по своему варианту – текст задачи, текст кода программы, схема алгоритма программы, результаты выполнения разработанной программы для разных наборов исходных данных; расчеты, выполненные вручную для тех же наборов исходных данных.

## Ход работы:

**Пример №1.** Программа из примера 1 работает корректно.

|  |  |  |
| --- | --- | --- |
| **Тест программы №1.**  Входные данные:a=3, b=4  Ожидаемый результат: b ≥ a Результат работы программы из примера: | **Тест программы №2.**  Входные данные: a=7, b=2  Ожидаемый результат: a > b Результат работы программы из примера: | **Тест программы №3.**  Входные данные: a=0, b=0  Ожидаемый результат: b ≥ a Результат работы программы из примера: |

**Пример №2.** Программа изменена таким образом, что в ней при равенстве значений переменных a и b выводится соответствующее сообщение.

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int a, b;

cout << "Enter first number: "; cin >> a;

cout << "Enter second number: "; cin >> b;

if (a > b)

{

cout << "a > b \n";

cout << "Ih raznost' ravna " << a - b;

}

else

{

if (b == a) cout << "a = b \n";

else cout << "b < a \n";

cout << "Ih raznost' ravna " << b - a;

}

return 0;

}

|  |  |  |
| --- | --- | --- |
| **Тест программы №1.**  Входные данные:a=3, b=5  Ожидаемый результат:  b > a, b – a = 5 – 3 = 2.  Результат работы программы из примера: | **Тест программы №2.**  Входные данные:a=3, b=3  Ожидаемый результат:  b = a, b – a = 0.  Результат работы программы из примера: | **Тест программы №3.**  Входные данные:a=100, b=5  Ожидаемый результат:  b < a, b – a = 100 – 5 = 95.  Результат работы программы из примера: |

**Задача №1.** Даны три вещественных числа. Верно ли, что они все разные?

|  |  |  |
| --- | --- | --- |
| **Пример №1.** Входные данные: вещественные числа a, b, c.  a = 1.000000001, b =1.0, c =2.0.  Ожидаемый результат: числа *a* и *b* одинаковые.  Необходимые расчёты (с точностью до 8-го знака после запятой):  | a – b | = |1+ – 1.0 | <, значит a = b;  | a – c | = | 1+– 2.0 | >, значит a ≠ c, b ≠ c.  Результат работы программы: | **Пример №2.** Входные данные: вещественные числа a, b, c.  a = 200.1, b = 0.0, c = 0.001.  Ожидаемый результат: все числа разные.  Необходимые расчёты (с точностью до 8-го знака после запятой):  | a – b | = | 200.1 – 0.0 | >, значит a ≠ b;  | a – c | = | 200.1 – 0.001| >, значит a ≠ c;  | b – c | = | 0.0 – 0.001| >,  значит b ≠ c.  Результат работы программы: | **Пример №3.** Входные данные: вещественные числа a, b, c.  a = -1.0000000009, b=4.4, c=-1.0.  Ожидаемый результат: числа *a* и *b* одинаковые.  Необходимые расчёты (с точностью до 8-го знака после запятой):  | a – b | = | -1-9\* – 4.4 | >, значит a ≠ b;  | a – c | = | -1-9\*| <, значит a = c;  | b – c | = | 4.4 + 1.0 | >,  значит b ≠ c.  Результат работы программы: |

Код программы:

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

const float e = 10e-8;

float a, b, c;

cin >> a >> b >> c;

if ((abs(a - b) <= e ) & (abs(b - c) <= e)) cout << "Все числа одинаковые \n";

else {

if (abs(a - b) <= e) cout << "Числа a и b одинаковые \n";

if (abs(b - c) <= e) cout << "Числа b и c одинаковые \n";

if (abs(a - c) <= e) cout << "Числа a и c одинаковые \n";

if (((abs(a - b) >= e) & (abs(b - c) >= e))&(abs(a - c) >= e) ) cout << "Все числа разные \n";

return 0;

}

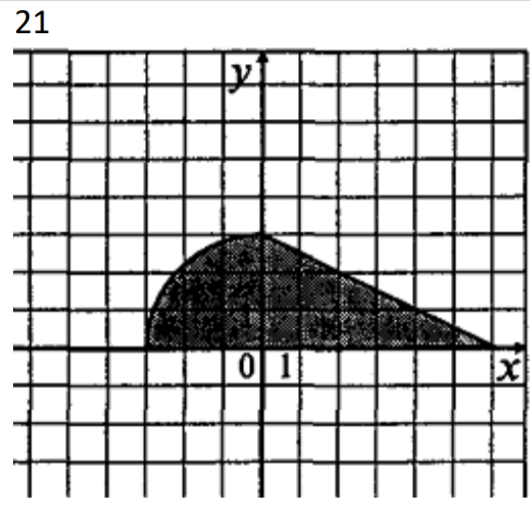
}



**Задача №2.** На свой день рождения Петя купил торт, который имел идеально круглую форму. Петя не знал, сколько гостей придет на его день рождения, поэтому вынужден был разработать алгоритм, согласно которому он сможет быстро разрезать торт на N равных частей. Следует учесть, что разрезы торта можно производить как по радиусу, так и по диаметру. Помогите Пете решить эту задачу, определив наименьшее число разрезов торта по заданному числу гостей. Вводится натуральное число N – число гостей, включая Петю (N <= 1000). Выведите минимально возможное число разрезов торта.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Пример №1.** Входные данные: N – число гостей, включая самого Петю, 1000 ≥ N ≥ 2.  N = 2.  Ожидаемый результат: k = 1 – число разрезов торта.  Необходимые расчёты: если число гостей чётно, то можно резать по диаметру, если же нечётно, то – по радиусу. N чётно, значит k = N/2 = 1.    Результат работы программы: | | **Пример №2.** Входные данные: N – число гостей, включая самого Петю, 1000 ≥ N ≥ 2.  N = 3.  Ожидаемый результат: k = 3 – число разрезов торта.  Необходимые расчёты: если число гостей чётно, то можно резать по диаметру, если же нечётно, то – по радиусу. N чётно, значит k = N= 3.    Результат работы программы: | | **Пример №3.** Входные данные: N – число гостей, включая самого Петю, 1000 ≥ N ≥ 2.  N = 8.  Ожидаемый результат: k = 4 – число разрезов торта.  Необходимые расчёты: если число гостей чётно, то можно резать по диаметру, если же нечётно, то – по радиусу. N чётно, значит k = N/2 = 4.    Результат работы программы: | |
| Код программы:  #include <iostream>  using namespace std;  int main()  {  int n, k;  cin >> n;  if (n == 1) cout << 0;  else {  if (n % 2 == 0) {  k = n / 2; //n - чёт  }  else { k = n; //n – нечёт }  cout << k;  } | | Схема программы: | |

**Задача №3.** Дана точка с действительными координатами x, y. Определить принадлежность точки закрашенной области:



С

А

В

|  |  |  |
| --- | --- | --- |
| **Пример №1.** Входные данные: вещественные координаты точки (x,y) с точностью до 8 знака после запятой.  Точка A (0;0).  Ожидаемый результат: in area  Необходимые расчёты: точка попадает в область при x [-3,6] и y[0,3]. Если точка попала в «левую» часть этой области, т.е. левее Оy, то её ордината должна находиться в пределах кривой , т.е. , иначе ее ордината ограничена сверху прямой y = 3 – 0,5x. Координаты данной точки удовлетворяют обоим случаям и попадают в закрашенную область.  Результат работы программы: | **Пример №2.** Входные данные: вещественные координаты точки (x,y) с точностью до 8 знака после запятой.  Точка B (5;5).  Ожидаемый результат: not in area  Необходимые расчёты: точка попадает в область x [-3,6], но не попадает в y[0,3]. Дальнейшие условия проверять не нужно, она уже не попадает в закрашенную область.    Результат работы программы: | **Пример №3.** Входные данные: вещественные координаты точки (x,y) с точностью до 8 знака после запятой.  Точка C (-1;2).  Ожидаемый результат: in area  Необходимые расчёты: точка попадает в область при x [-3,6] и y [0,3]. Точка попала в «левую» часть этой области, т.е. лежит левее Оy, значит её ордината должна находиться в пределах кривой , т.е. Координаты данной точки удовлетворяют этому условию: 2 , а значит точка попадает в закрашенную область.  Результат работы программы: |

Схема программы:



Код программы:

#include <iostream>

using namespace std;

int main()

{

float x, y;

cin >> x >> y;

if (((x > 6.0) || (x < -3.0)) || ((y < 0.0) || (y > 3.0))) cout << "not in area \n";

else {

if ((x <= 0.0) & (x >= -3.0)) {

if (y <= sqrt(9.0 - x \* x)) cout << "in area \n";

else cout << "not in area \n";

}

if ((x <= 6.0) & (x > 0.0)) {

if (y <= 3.0 - 0.5 \* x) cout << "in area \n";

else cout << "not in area \n";

}

} }