Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 3**

Разветвляющаяся программа  
Вариант № 12

Преподаватель \_\_\_\_\_\_\_\_ Пушкарев К. В.

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_ 01.10.2018 Котов С.А.

подпись

Красноярск 2018

**1 Задание**

1. Разработать графическую схему алгоритма и программу, которая для двух целых чисел, введенных с клавиатуры, вычисляет остаток от целочисленного деления, частное от целочисленного деления первого числа на второе, а также частное от вещественного деления.

2. Провести трассировку программы с помощью встроенного в среду программирования отладчика, анализируя значения переменных после каждого оператора присваивания.

3. Выполнить несколько запусков программы для заранее подготовленных тестовых наборов данных. Сделать вывод результатов с применением потокового ввода-вывода, используя следующие методы потоков – width(), precision() и fill() с различными параметрами (не менее 3 для каждого метода), а также флаги left, right, устанавливаемые с помощью метода setf(). Проанализировать полученные результаты.

4. Выполнить несколько запусков программы для заранее подготовленных тестовых примеров для проверки работоспособности всех ветвей программы. Для тестовых примеров рекомендуется взять как произвольные значения исходных данных, так и критические значения (например, ноль). Полученные результаты проанализировать.

**2 Цели работы**

1. Закрепить навыки работы в среде С++.
2. Познакомиться с потоковым вводом-выводом (сin, cout).
3. Изучить операцию условия.
4. Изучить условный оператор.
5. Изучить оператор множественного выбора.

**3 Подход к решению задач**

Согласно условию задачи, требуется разработать графическую схему алгоритма и программу, которая для двух целых чисел, введенных с клавиатуры, вычисляет остаток от целочисленного деления, частное от целочисленного деления первого числа на второе, а также частное от вещественного деления.

Создадим 2 переменные num1 и num2 соответственно, значения которых будет введено с клавиатуры.

Так же следует создать переменную mod, в которой будет записан остаток от целочисленного деления, вычисленного по формуле mod = num1 % num2;  
Переменную div\_i в которой будет записано частное от целочисленного деления, вычисленного по формуле div\_i = num1 / num2;  
Переменную div\_f в которой будет записано частное от вещественного деления, вычисленного по формуле div\_f = float(num1) / num2;  
После всех вычислений выведем переменные mod, div\_i, div\_f на экран.

**4 Графическая схема алгоритма**

  
Рисунок 1 – Блок-схема основного упражнения.

**5 Код программы**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 28 29 30 31 32 33 34 35 36 37 38 39 40  41  42  43  44  45  46 47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84 | #include "pch.h"  #include <iostream>  #include <clocale>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  int num1, num2, mod, div\_i;  float div\_f;  cout << "Введите первое число: " << endl;  cin >> num1;  cout << "Введите второе число: " << endl;  cin >> num2;  if (num2 != 0) {  mod = num1 % num2;  div\_i = num1 / num2;  div\_f = float(num1) / num2;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(40);  cout.precision(2);  cout.fill('\*');  cout << "Остаток от целочисленного деления = " << mod << endl;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(50);  cout.precision(7);  cout.fill('#');  cout << "Остаток от целочисленного деления = " << mod << endl;  cout.setf(ios::fixed);  cout.setf(ios::right);  cout.width(75);  cout.precision(5);  cout.fill('$');  cout << "Остаток от целочисленного деления = " << mod << endl << endl;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(40);  cout.precision(2);  cout.fill('\*');  cout << "Частное от целочисленного деления = " << div\_i << endl;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(50);  cout.precision(7);  cout.fill('#');  cout << "Частное от целочисленного деления = " << div\_i << endl;  cout.setf(ios::fixed);  cout.setf(ios::right);  cout.width(75);  cout.precision(5);  cout.fill('$');  cout << "Частное от целочисленного деления = " << div\_i << endl << endl;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(40);  cout.precision(2);  cout.fill('\*');  cout << "Частное от вещественного деления = " << div\_f << endl;  cout.setf(ios::fixed);  cout.setf(ios::left);  cout.width(50);  cout.precision(7);  cout.fill('#');  cout << "Частное от вещественного деления = " << div\_f << endl;  cout.setf(ios::fixed);  cout.setf(ios::right);  cout.width(75);  cout.precision(5);  cout.fill('$');  cout << "Частное от вещественного деления = " << div\_f << endl;  }  else {  cout << "Произошла ошибка в вычислениях." << endl;  }  return 0;  } |

**6 Результат выполнения экспериментальной части работы.**

Результаты запуска программы с различными входными значениями приведены в таблице 1.

Таблица 1 – Работа программы на тестовых наборах данных

| Входные данные | Результат |
| --- | --- |
| num1 = 5, num2 = 2 | Остаток от целочисленного деления = \*\*\*\*1  Остаток от целочисленного деления = ##############1  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Остаток от целочисленного деления = 1  \*\*\*\*Частное от целочисленного деления = 2  ##############Частное от целочисленного деления = 2  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Частное от целочисленного деления = 2  \*\*\*\*\*Частное от вещественного деления = 2.50  ###############Частное от вещественного деления = 2.5000000  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Частное от вещественного деления = 2.50000 |
| num1 = 48, num2 = 15 | Остаток от целочисленного деления = \*\*\*\*3  Остаток от целочисленного деления = ##############3  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Остаток от целочисленного деления = 3  \*\*\*\*Частное от целочисленного деления = 3  ##############Частное от целочисленного деления = 3  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Частное от целочисленного деления = 3  \*\*\*\*\*Частное от вещественного деления = 3.20  ###############Частное от вещественного деления = 3.2000000  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Частное от вещественного деления = 3.20000 |

**7 Дополнительное упражнение № 1**

2. Грузовой автомобиль выехал из одного города в другой со скоростью v1 км/ч. Через t ч в этом же направлении выехал легковой автомобиль со скоростью v2 км/ч. Составить программу, определяющую, догонит ли легковой автомобиль грузовой через t1 ч после своего выезда. Значения v1, v2, t, t1 вводятся с клавиатуры.

**8 Подход к решению задачи дополнительного упражнения № 1**

Согласно условию задачи, требуется разработать программу, которая будет определять, догонит ли легковой автомобиль грузовой через t1 часов после своего выезда. Создаем переменные типа float: v1, v2, t и t1. Запрашиваем ввод значений с клавиатуры в эти переменные. Создаем дополнительные вспомогательные переменные типа float: s1 и s2. Они нам нужны, для вычисление пути, который прошли автомобили. s1 будет высчитываться по форуме s1 = v1 \* (t + t1), а s2 соответственно по формуле s2 = v2 \* t1. Сравним полученные значения s1 и s2 и определим, сможет ли легковой автомобиль догнать грузовой.

**9 Код программы дополнительного упражнения № 1**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | #include "pch.h"  #include <iostream>  #include <clocale>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  float v1, v2, t, t1;  cout << "Введите скорость грузового автомобиля: ";  cin >> v1;  cout << "Введите скорость легкового автомобиля: ";  cin >> v2;  cout << "Введите время выезда легкового автомобиля: ";  cin >> t;  cout << "Введите время легкового автомобиля в пути: ";  cin >> t1;  float s1, s2;  s1 = v1 \* (t + t1);  s2 = v2 \* t1;  if (s1 <= s2) {  cout << "\nЛегковой автомобиль догонит грузовой за это время." << endl;  }  else {  cout << "\nЛегковой автомобиль не сможет догнать грузовой за это время." << endl;  }  return 0;  } |

**10 Результат выполнения экспериментальной части работы дополнительного упражнения № 1**

Результаты запуска программы с различными входными значениями приведены в таблице 1.

Таблица 1 – Работа программы на тестовых наборах данных

| Входные данные | Результат |
| --- | --- |
| v1 = 50 v2 = 90  t = 1  t1 = 3 | Легковой автомобиль догонит грузовой за это время. |
| v1 = 75 v2 = 80  t = 2  t1 = 6 | Легковой автомобиль не сможет догнать грузовой за это время. |