Министерство образования и науки РФ
ФГБОУ ВПО «ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра «Информационная безопасность систем и технологий»

Отчёт
о лабораторной работе №1
Основы работы в среде CodeLite

Дисциплина: Языки программирования
Группа: 18ПИ1
Выполнил: М.М. Колясов
Количество баллов:
Дата сдачи:
Принял: к.т.н., доцент М.Ю. Лупанов

2019

1 Цель работы

Освоить процесс создания приложений с использованием среды разработки CodeLite.

2 Задание

2.1 Создать новое рабочее пространство. Создать в нем новый проект для консольного приложения с шаблоном Simple executable (g++). Поместить в проект текст программы из приложения А. Отформатировать код программы.

2.2 Выполнить сборку проекта. Проанализировать обнаруженные ошибки. Исправить ошибки. Отформатировать код программы. Выполнить повторную сборку проекта. Запустить программу и проанализировать результат ее работы.

2.3 Переключить проект в конфигурацию Release. Выполнить сборку проекта. Установить в конфигурации проекта следующие параметры командной строки: «фамилия» «имя» «отчество» «номер зачетки» (4 параметра, каждый указывает личные данные). Запустить программу и проанализировать результат ее работы.

2.4 Создать новый консольный проект в уже имеющимся рабочем пространстве. Поместить в проект текст программы из приложения Б. Отформатировать код программы. Выполнить сборку проекта. Выполнить пошаговую отладку программы, наблюдая за переменными iRes и fRes на каждом шаге. Результаты наблюдения свести в таблицу.

2.5 Создать новый консольный проект в уже имеющимся рабочем пространстве. Поместить в проект текст программы из приложения В. Отформатировать код программы. Выполнить сборку проекта. Запустить программу и проанализировать результат ее работы.

2.6 . Поставить в программу точку останова на функцию toUpperCase и настроить игнорирование первого срабатывания. Выполнить программу в режиме отладки наблюдая за значениями переменных i и s. Результаты

наблюдения свести в таблицу. Проанализировать результаты наблюдения и идентифицировать ошибку в программе.

2.7 Исправить программу или предложить решение по ее исправлению.

3 Результаты работы

3.1 Было создано новое рабочее пространство, используя пункт меню «Workspace::New Workspace...», далее открывается окно выбора типа проекта, приведенное на рисунке 1. Здесь был выбран пункт C++. Сразу после этого открывается окно создания рабочего пространства, приведенное на рисунке 2. В этом окне было заполнено имя рабочего пространства, а также указан путь к папке, где будет создано наше рабочее пространство. После корректного заполнения полей и нажатия кнопки «Ок» было создано рабочее пространство, в которое можно добавлять новые проекты.

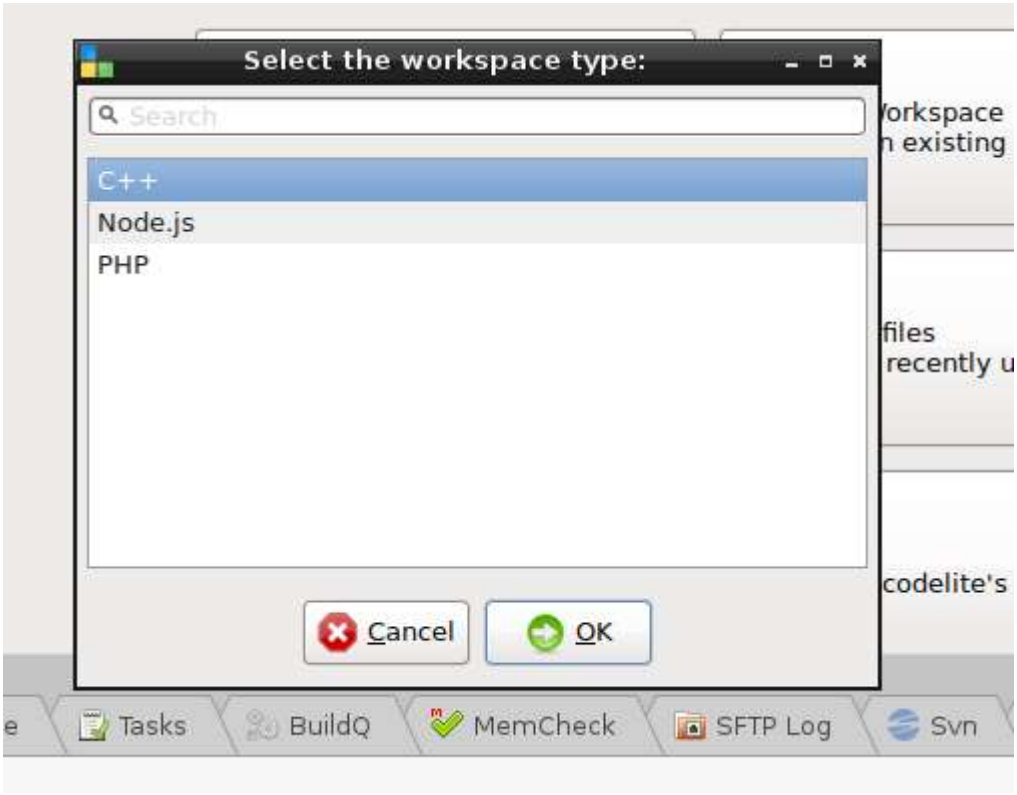

Рисунок 1 - Окно выбора типа проекта

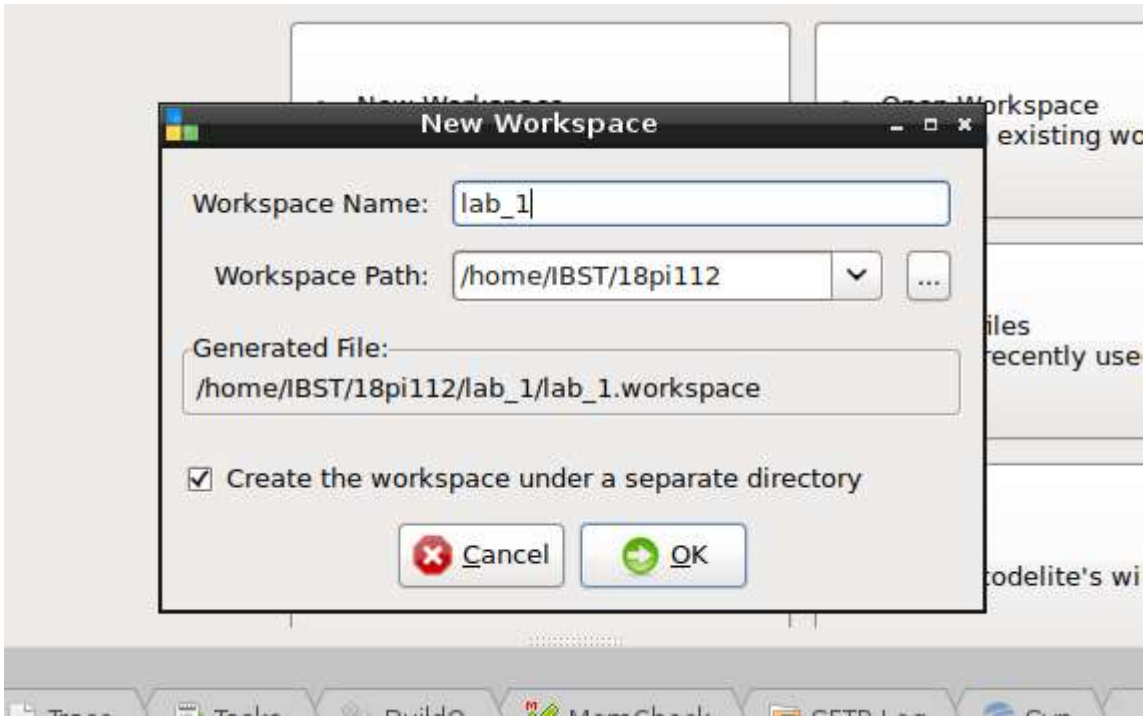

Рисунок 2 - Окно создания рабочего пространства

Для добавления нового проекты был использован пункт меню «Workspace::New Project». Для программирования на C++ в Linux с использованием компилятора GCC необходимо выбрать шаблон «Simple executable (g++)» (рисунок 3). После выбора шаблона необходимо перейти ко второму шагу, нажав кнопку «Next». Окно мастера на втором шаге приведено на рисунке 4. На этом шаге было выбрано название проекта. Путь к проекту будет сформирован автоматически для нашего рабочего пространства, хотя при необходимости его можно изменить. Для перехода к следующему шагу необходимо нажать кнопку «Next». Третий шаг мастера предназначен для опытных пользователей, поэтому его можно завершить, нажав кнопку «Finish»

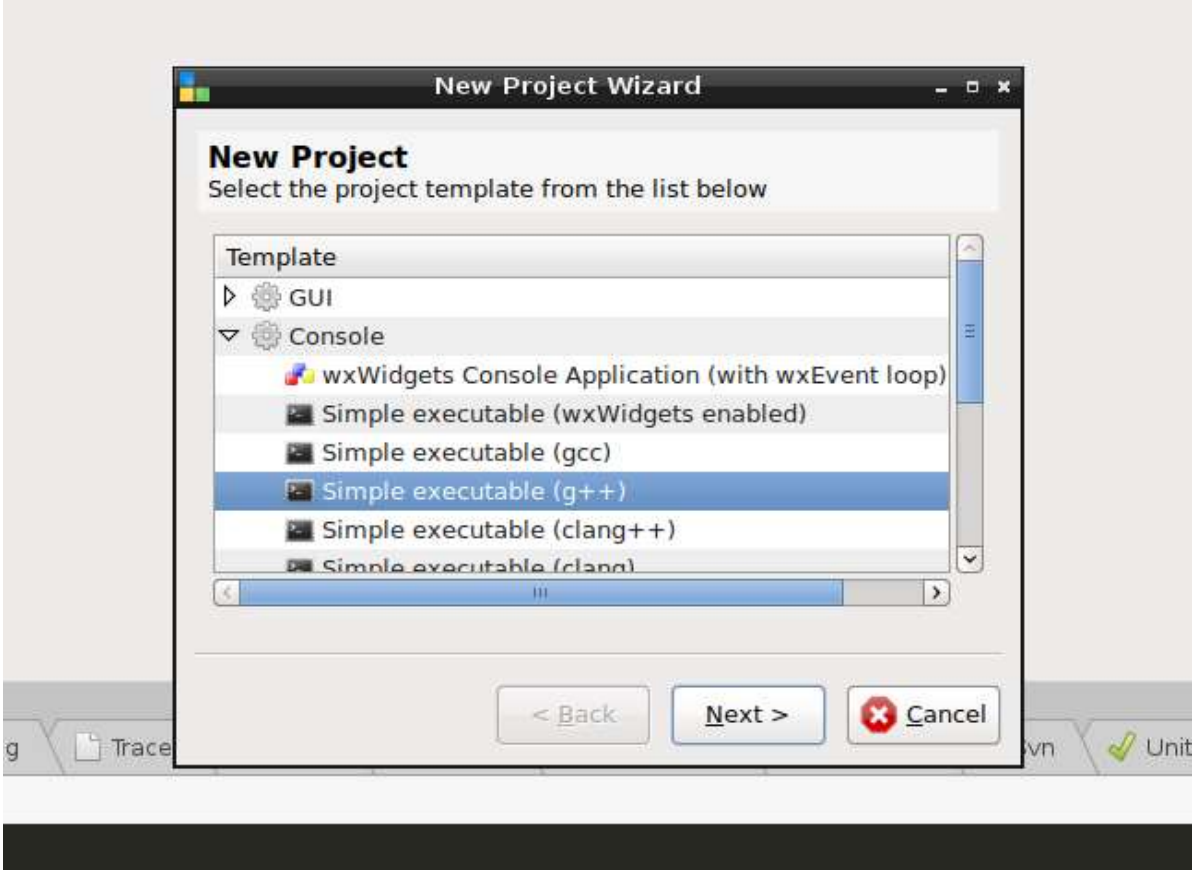

Рисунок 3 - Мастер создания проектов. Шаг 1

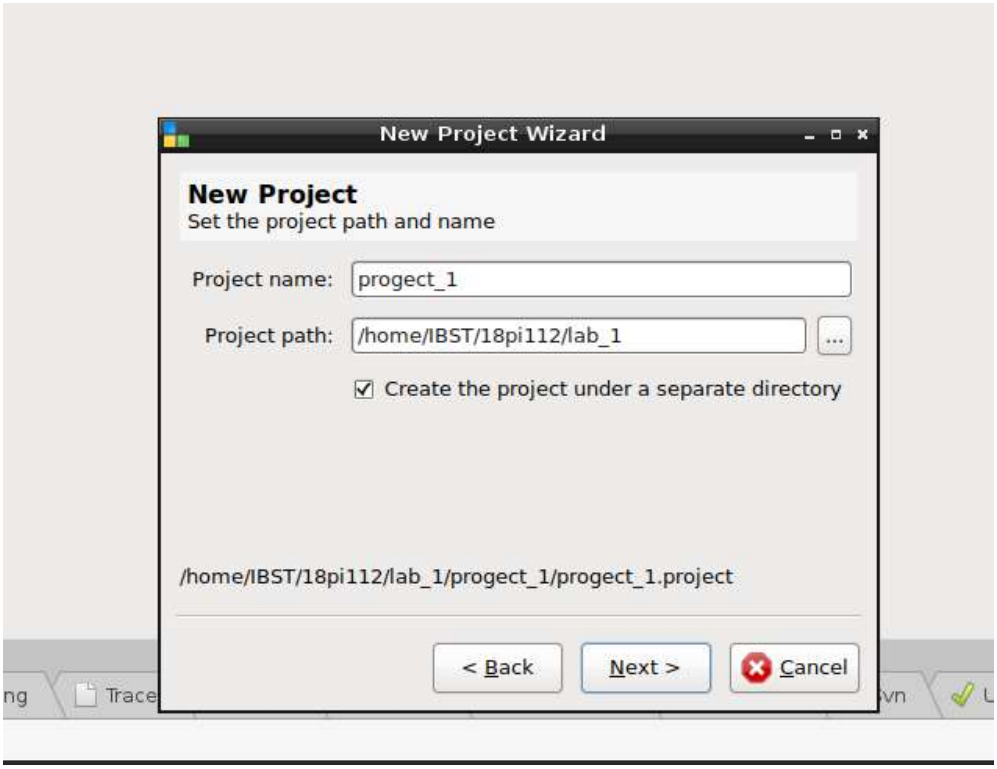

Рисунок 4 - Мастер создания проектов. Шаг 2

В проект был помещен текст программы и отформатирован с помощью комбинации клавиш «Ctrl+I».

```
#include <climits>
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char** argv)
{
   char* path = new char[PATH_MAX];
   const char* id[] = { "Фамилия\t\t", "Имя\t\t", "Отчество\t", "Группа\t\t" };
   realpath(argv[0], path);
   cout << "Путь к исполняемому файлу: ";
   cout << path << endl;
   delete[] path;
   for(int i = 1; i < argc; ++i) {
      if(i - 1 < 4) cout << id[i - 1];
      cout << string(agv[i]) << endl;
      return 0;
   }
```

3.2 Сборка проекта в CodeLite была выполнена через меню «Build::Build Project» (или нажатием клавиши «F7»). Текст программы имел несколько ошибок рисунок 5.

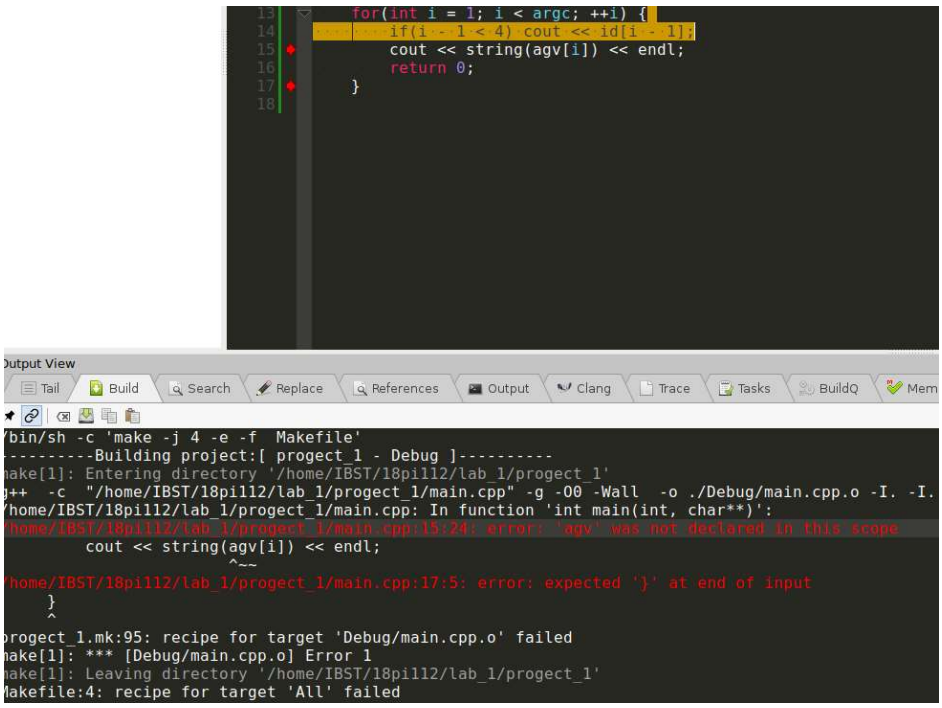

Рисунок 5 - Сборка проекта

Проанализировав программу, ошибки были исправлены и текст программы приобрел следующий вид:

```
#include <climits>
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char** argv)
{
   char* path = new char[PATH_MAX];
   const char* id[] = { "Фамилия\t\t", "Имя\t\t", "Отчество\t", "Группа\t\t" };
   realpath(argv[0], path);
   cout << "Путь к исполняемому файлу: ";
   cout << path << endl;
   delete[] path;
   for(int i = 1; i < argc; ++i) {
      if(i - 1 < 4) cout << id[i - 1];
      cout << string(argv[i]) << endl;
   }
   return 0;
}
```

Была выполнена повторная сборка проекта, на выходе ошибок не обнаружено и программа выдала следующий результат (рисунок 6).

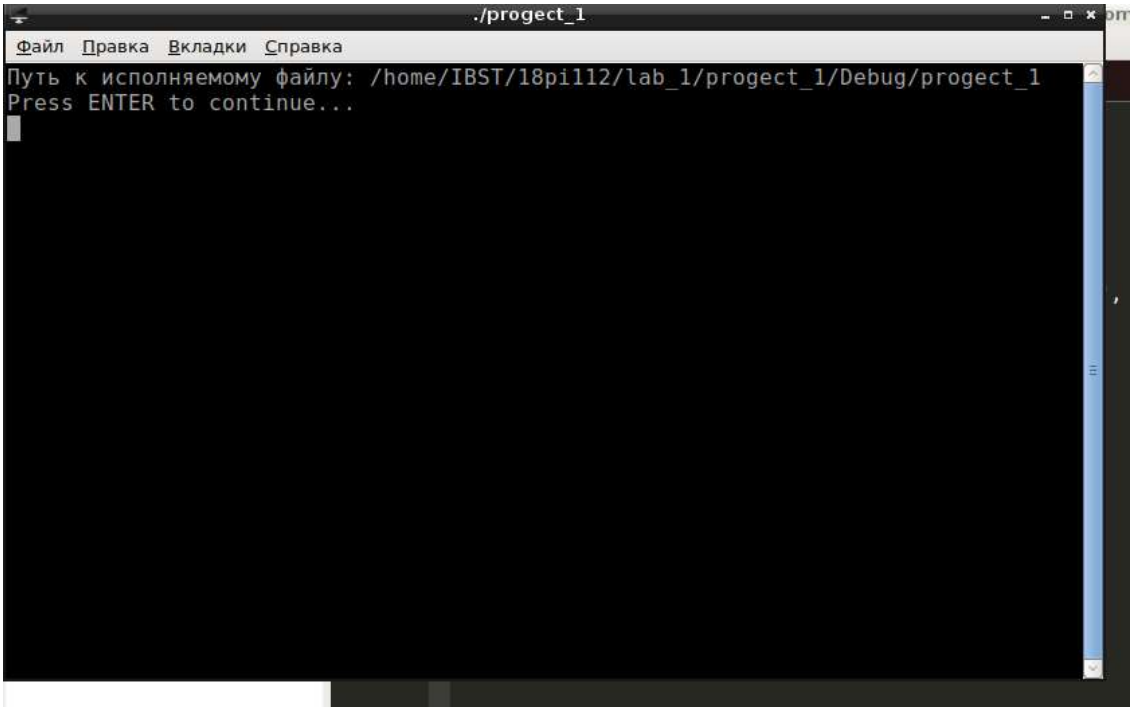

Рисунок 6 - Результат программы 1

3.3 Окно установки конфигурации проекта было вызвано через меню «Workspace::Open Active Project Settings...» (или комбинацией клавиш «Alt+F7»). В открывшемся окне был выбран Release (рисунок 7), а затем в левой части выбран пункт «General», а в левой — раскрыт узел «Execution» и выбран «Program Arguments», как показано на рисунке 8. Программе были переданы 4 параметра в соответствии с заданием.

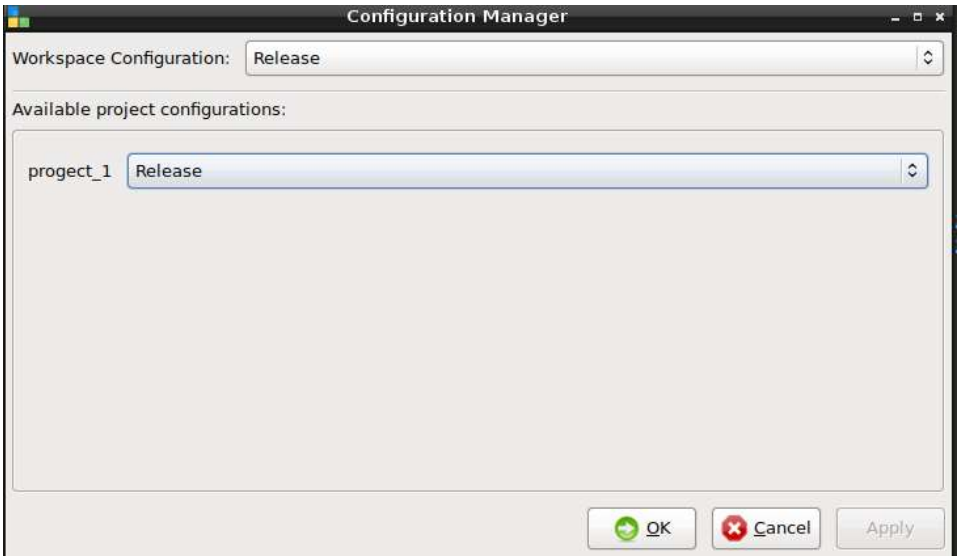

Рисунок 7 - Смена конфигурации

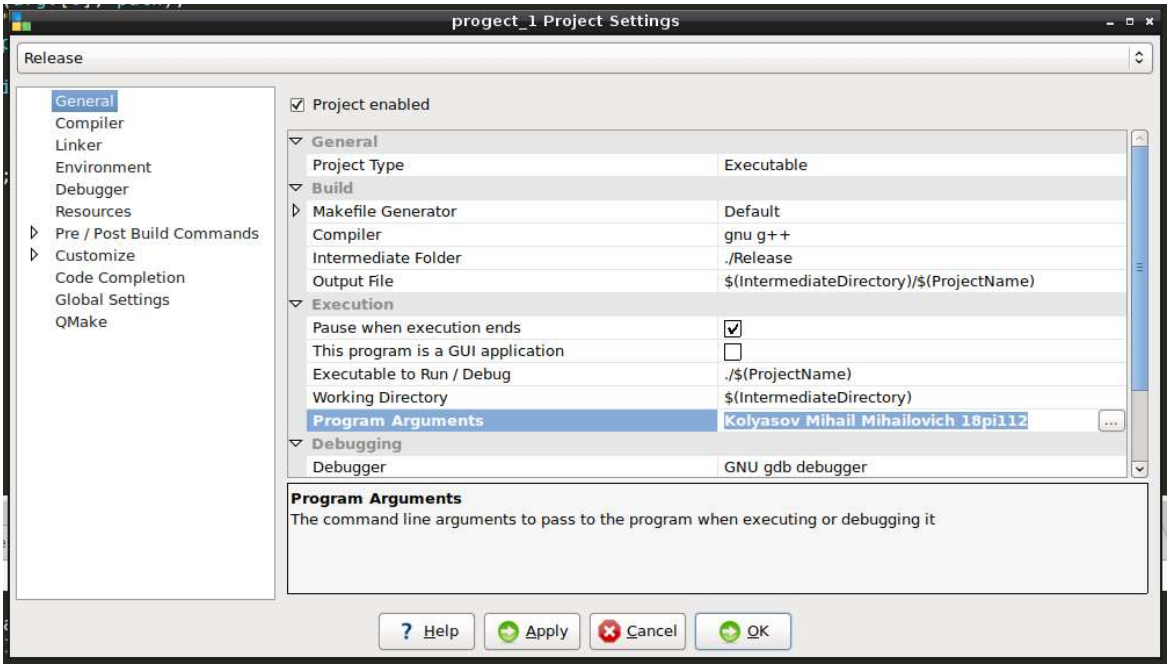

Рисунок 8 - Задание параметров

Запустив программу были получены следующие результаты(рисунок 9).

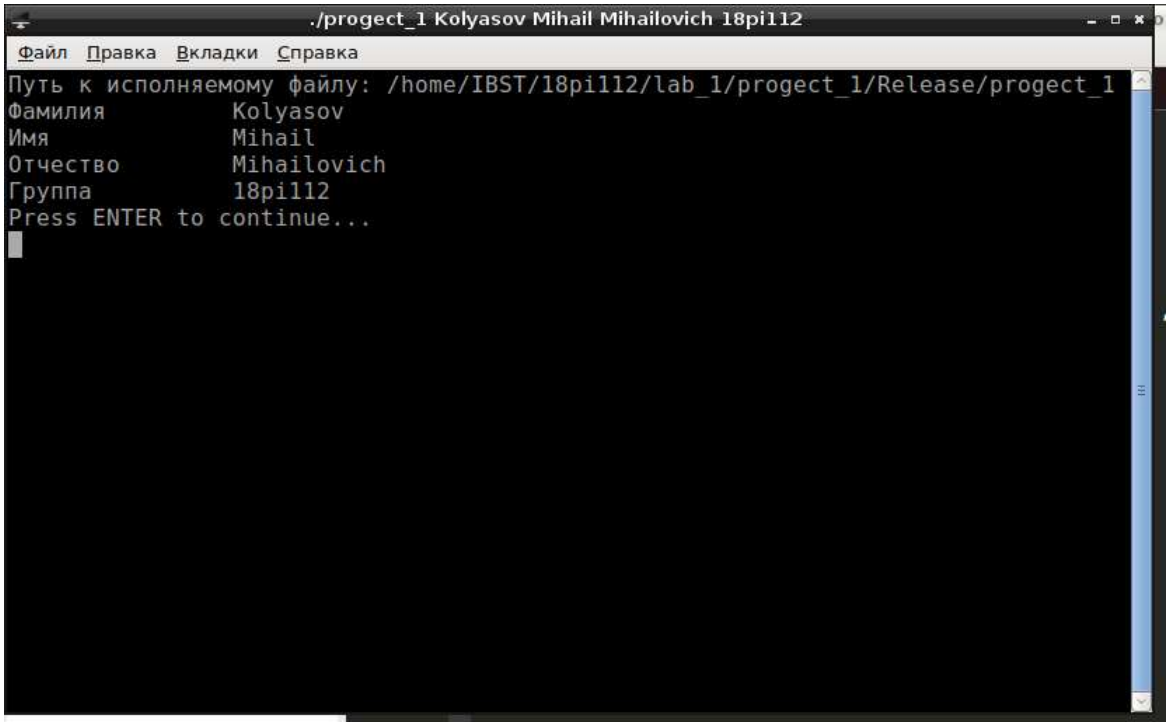

Рисунок 9 - Результаты программы 1 после задания параметров

3.4 Был создан новый консольный проект уже в имеющимся рабочем пространстве. В проект был помещен текст программы и отформатирован.

```
#include <iostream>
using namespace std;
int main(int argc, char** argv)
{
   int n, N;
   int iRes;
   double fRes;
   // ввод данных
   // значение N – номер зачетки, где
   //              П заменяется на 1
   //              И заменяется на 2
   //              Т заменяется на 3
   // значение n – четыре младших цифры N
   // Пример 18ПИ105 => N = 1812105, n = 2105
   cout << "Введите N: ";
   cin >> N;
   cout << "Введите n: ";
   cin >> n;
   //  деление
```

```
    iRes = N / n;
    fRes = N / n;
    fRes = 1.0 * N / n;
    //  умножение и деление
    iRes = N / 10 * n;
    iRes = N * n / 10;
    // сложение
    iRes = N + 2140000000;
    iRes = N + 2150000000;
    // присваивание
    iRes = fRes;
    // xor
    iRes = iRes ^ iRes;
    return iRes;
}
```

Далее была проведена сборка, ошибок не обнаружено. Была выполнена пошаговая отладка программы, наблюдая за переменными iRes и fRes на каждом шаге. Для этого программа была запущена с отладчиком с помощью клавиши F5, но перед этим были поставлены точки остановы на 5 и 33 строках соответственно. Далее спускаясь по шагу вниз с помощью клавиши F10, были занесены результаты изменения выбранных переменных в таблицу 1.

Таблица 1

№ шага	fRes	iRes
1	6,95e-310	0
2	6,95e-310	0
3	6,95e-310	0
4	6,95e-310	0
5	6,95e-310	858
6	858	858
7	858,075757575	858
8	858,075757575	382717632
9	858,075757575	-467786675
10	858,075757575	21418112113
11	858,075757575	-21433155184

12	858,075757575	858
13	858,075757575	0

3.5 Был создан новый консольный проект уже в имеющимся рабочем пространстве. В проект был помещен текст программы и отформатирован.

```
#include <iostream>
using namespace std;
/* функция преобразует символы строки к верхнему регистру
 * Параметры:
 *     s - указатель на строку с нулевым окончанием
 * Результат:
 *     преобразует строку "на месте" */
void toUpperCase(char * s)
{
   int i = 0;
   while(s[i] != 0) {
      s[i] = s[i] - 32;
      i++;
   }
}
// главная функция программы
// проверяем работу функции toUpperCase()
int main()
{
   char test[] = "test";
   toUpperCase(test);
   cout << test << endl;
   char hello[] = "Hello! World";
   toUpperCase(hello);
   cout << hello << endl;
   return 0;
}
```

Далее была проведена сборка, ошибок не обнаружено. После запуска, программа выдала следующие результаты(рисунок 10). По своей задаче программа должна была заменить строчные буквы в строке на прописные, для первой строки задание выполнено корректно, что нельзя сказать про вторую.

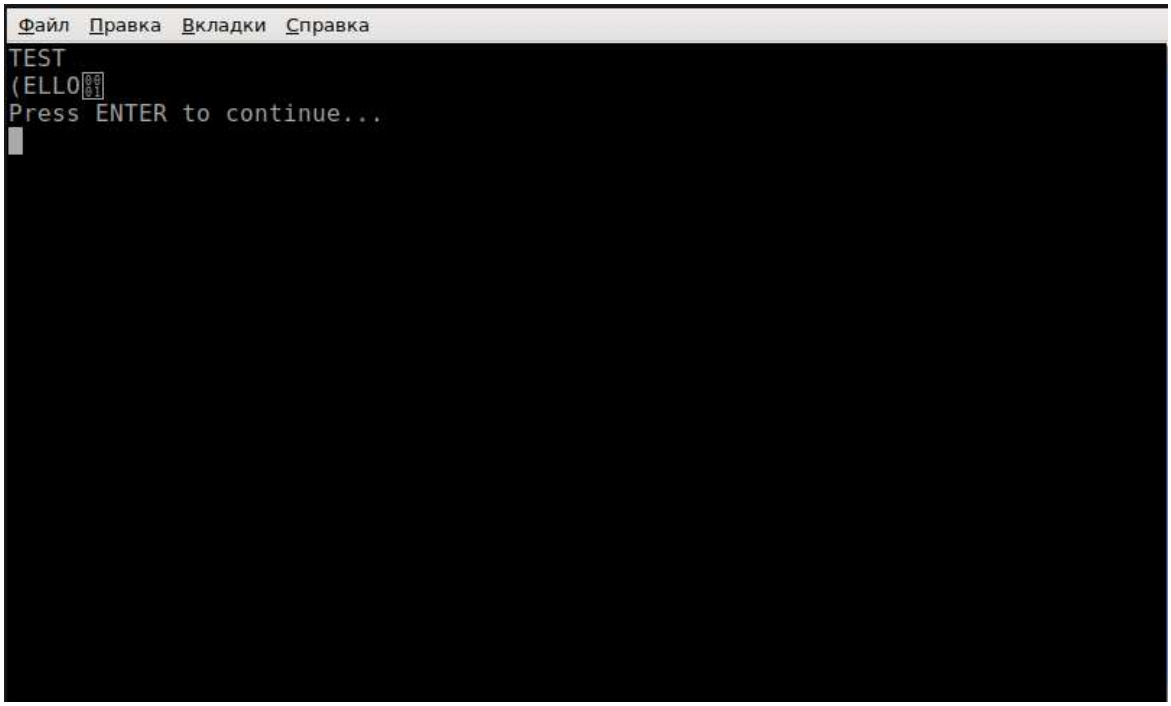

Рисунок 10 - Результат работы программы 3.

3.6 Была поставлена точка остановы на заданной функции и применено для нее игнорирование первого срабатывания. Для этого в пункте меню Edit Breakpoint выставлен необходимый параметр(рисунок 11).

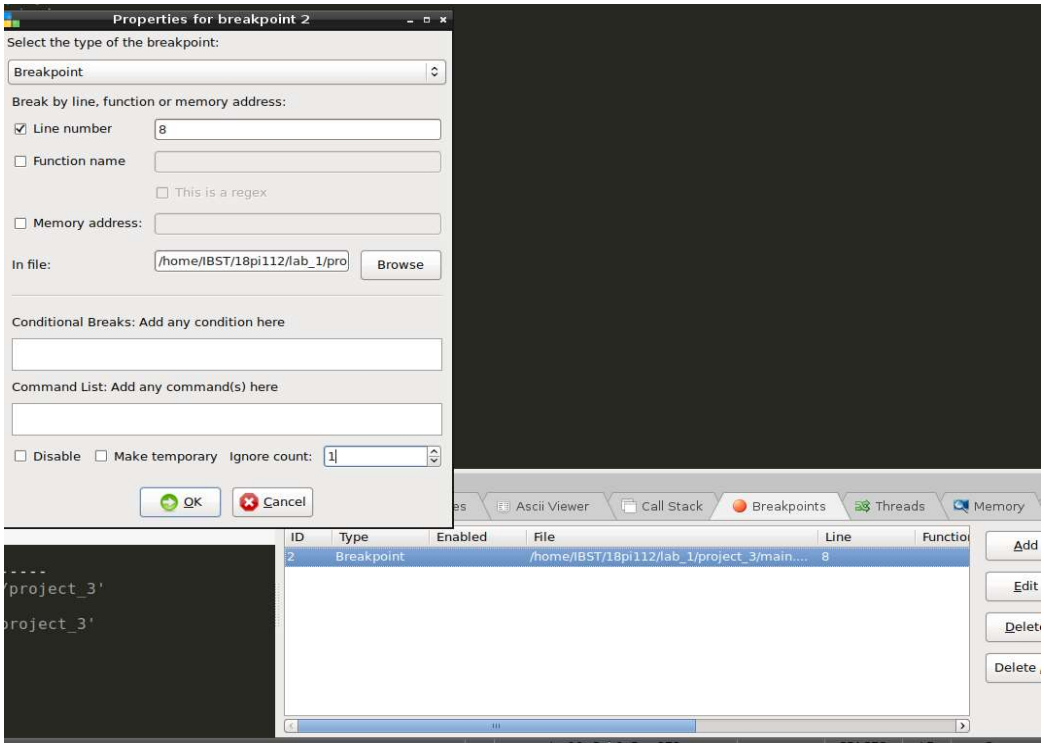

Рисунок 11 - Игнорирование первого срабатывания

Программа была запущена в режиме отладки, наблюдая за значениями переменных s,i

Ошибка при выполнении программы заключается в том, что она применима только к строчный символам, а во второй строке встречается как прописные символы, не буквенной символ и пробел. И код пробела равняется коду, который мы вычитаем, что приводит к получению кода завершения 0.

3.7 Одним из вариантов решения поставленной задачи является введение условия, которое проверяет является ли данный символ строчной буквой. В истинном случае подвергать его обработке, в противном пропускать.

4 Вывод

В процессе выполнения лабораторной работы были получены навыки при работе со средой для создания приложений CodeLite.