

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики**
Факультет Фотоники и оптоинформатики
Кафедра Компьютерной фотоники и видеоинформатики

Отчет по практике

Выполнил:	Антонов А. С.
Группа:	V3316
Преподаватель:	Кудрявцев А. С.

Санкт-Петербург, 2018

Оглавление

Цель проведения практики	3
Задание №1. Распределенная система контроля версий Git	4
Задание №2. Форматирование и стиль	11
Задание №3. TDD: Разработка через тестирование.	14
Отчет по практической работе.....	19
Приложение. Презентация.	22

Цель проведения практики

Освоение навыков использования C++ и изучение приемов разработки программного обеспечения. Практика проходит в компьютерном классе и состоит из лекционных занятий и практических заданий.

Задание №1. Распределенная система контроля версий Git

1) Условия задания.

1. Завести аккаунт на <https://github.com/>.
2. Создать репозиторий для лабораторных работ.
3. Склонировать репозиторий из <https://github.com/> на компьютер (git clone ...).
4. Создать папку lab1.
5. Написать в папке lab1 программу, вычисляющую функцию факториала.
6. Создать .gitignore, добавив в игнорируемые файлы – промежуточные файлы компиляции (объектные файлы и другие временные файлы) и исполняемый файл. В результате должны остаться только файл проекта и исходные файлы.
7. Сделать по крайней мере два коммита. Отправить зафиксированные изменения на <https://github.com/> (git push)
8. Привести результаты работы “git log”.
9. Показать при помощи “git diff <ваш_файл_ваши_коммиты>” изменение любого файла между двумя коммитами.

2) Текст программы факториала.

```
#include <iostream>

unsigned int factorial(unsigned int number) {
    if (number == 0) return 1;
    return number * factorial(number - 1);
}

int main() {
    unsigned int number;
    std::cin >> number;
    std::cout << factorial(number) << std::endl;
}
```

3) Выполненные команды и результаты вывода.

Аккаунт на <https://github.com/>.

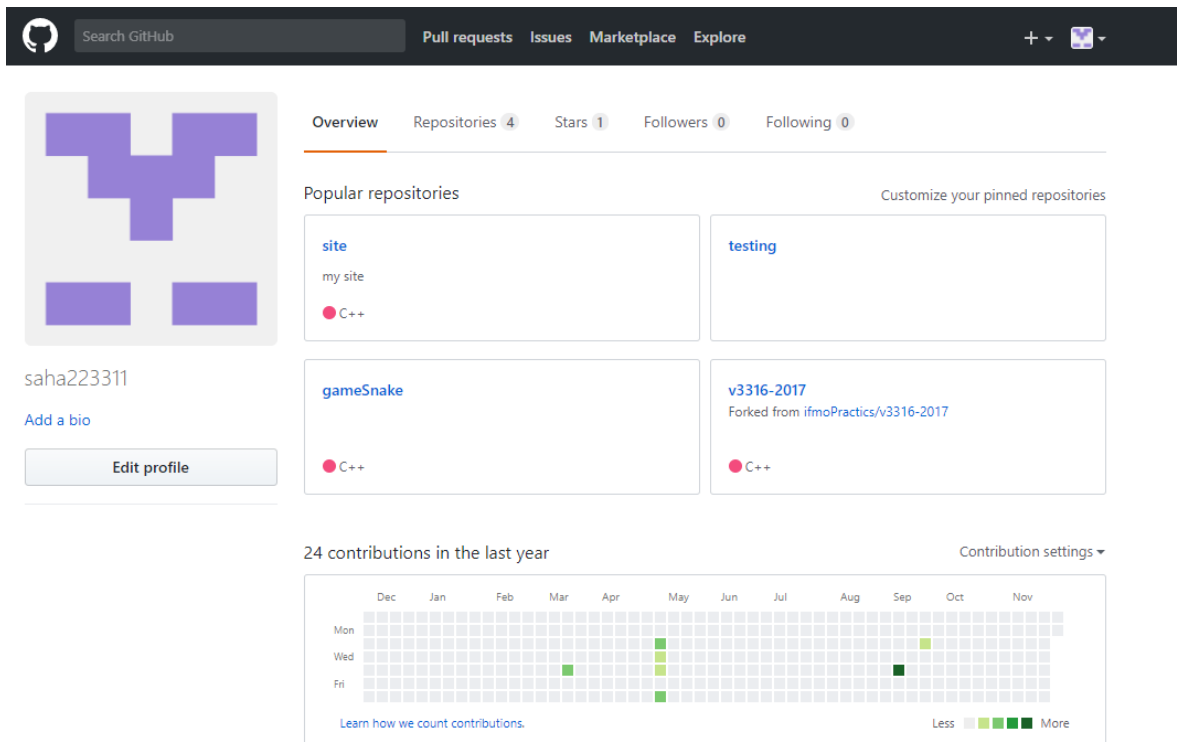


Рис 1. Аккаунт на <https://github.com/>.

Создадим репозиторий для лабораторных работ.

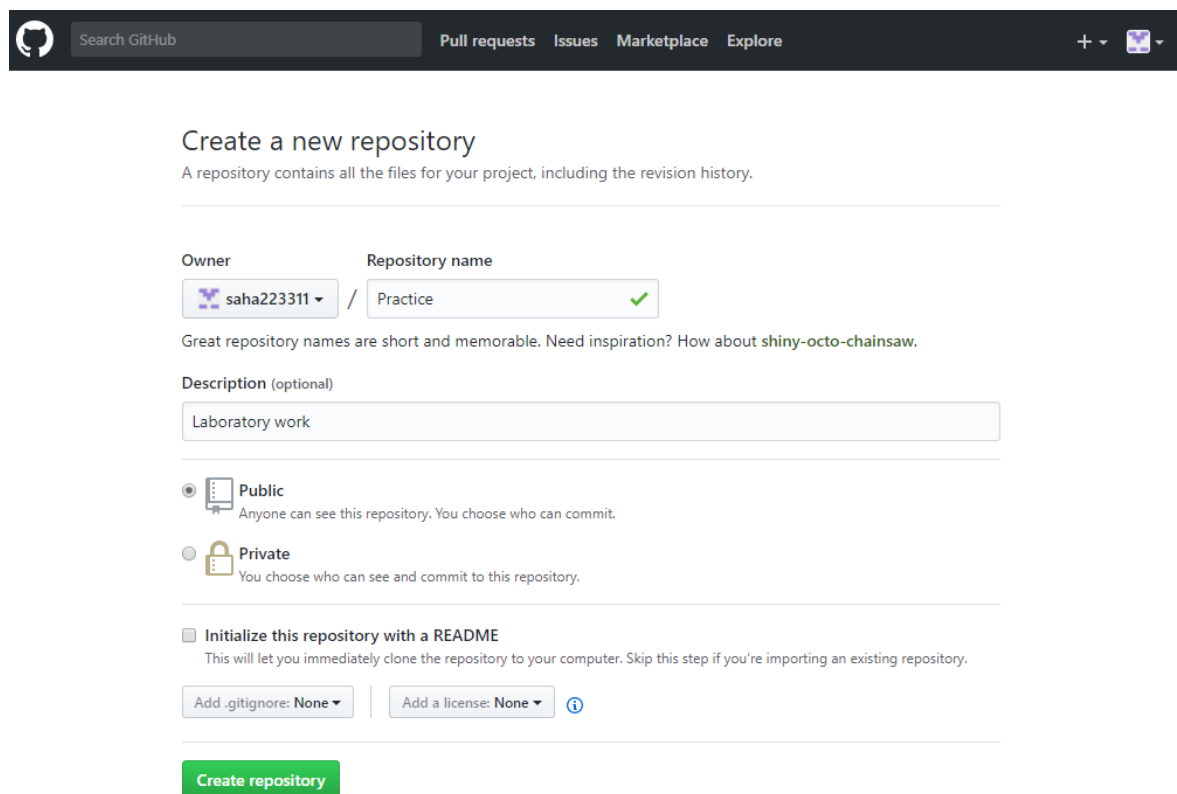
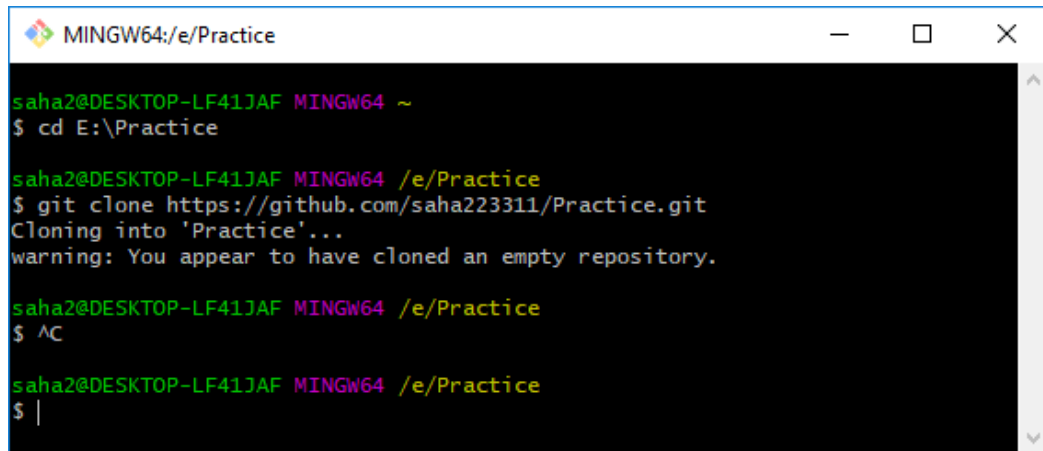


Рис 2. Создание репозитория для лабораторных работ.

Склонируем репозиторий из [https://github.com/](https://github.com) на компьютер.



```
MINGW64:/e/Practice

saha2@DESKTOP-LF41JAF MINGW64 ~
$ cd E:\Practice

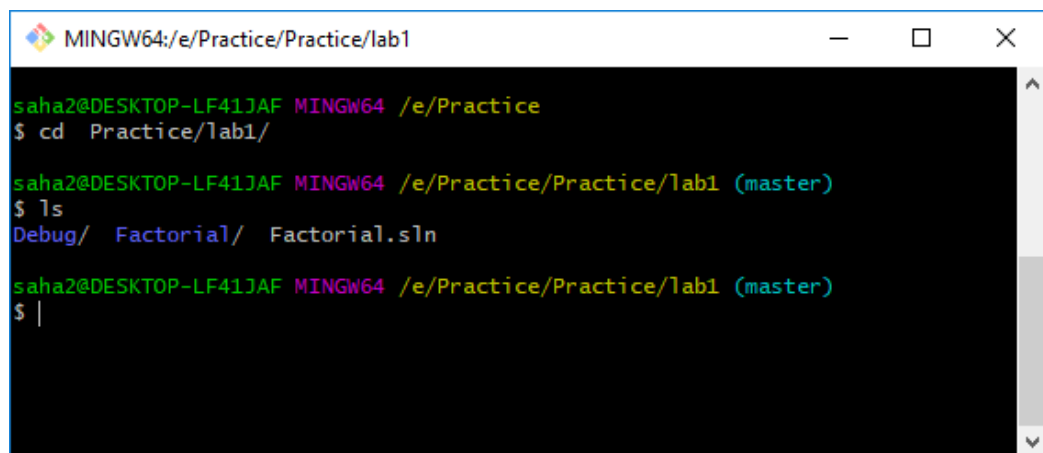
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice
$ git clone https://github.com/saha223311/Practice.git
Cloning into 'Practice'...
warning: You appear to have cloned an empty repository.

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice
$ ^C

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice
$ |
```

Рис 3. Клонирование репозитория на компьютер.

Создадим папку lab1 и напишем там программу, вычисляющую функцию факториала.



```
MINGW64:/e/Practice/Practice/lab1

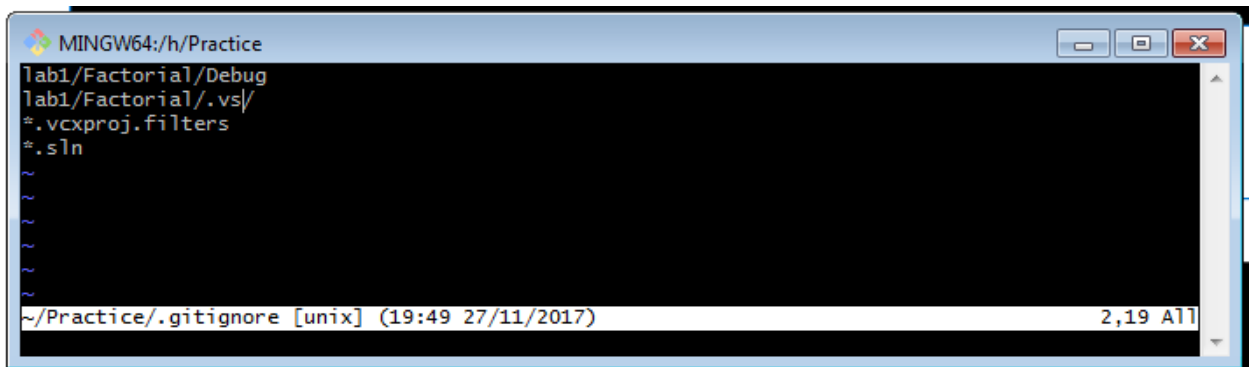
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice
$ cd Practice/lab1/

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice/lab1 (master)
$ ls
Debug/  Factorial/  Factorial.sln

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice/lab1 (master)
$ |
```

Рис 4. Папка с написанной нами программой.

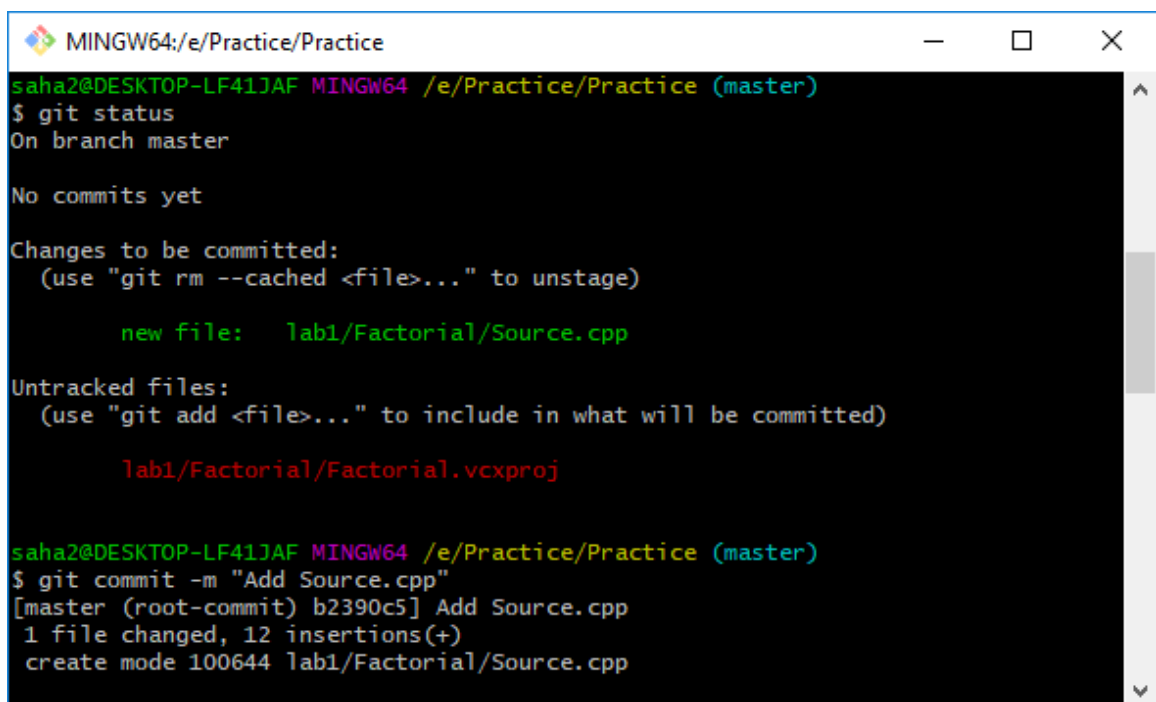
Создать .gitignore, добавив в игнорируемые файлы – промежуточные файлы компиляции (объектные файлы и другие временные файлы) и исполняемый файл. В результате должны остаться только файл проекта и исходные файлы.



```
MINGW64:/h/Practice
lab1/Factorial/Debug
lab1/Factorial/.vs/
*.vcxproj.filters
*.sln
~
~
~
~
~
~/Practice/.gitignore [unix] (19:49 27/11/2017) 2,19 All
```

Рис 5. Содержание файла .gitignore.

Сделаем несколько коммитов. Отправим зафиксированные изменения на <https://github.com/>



```
MINGW64:/e/Practice/Practice
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   lab1/Factorial/Source.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    lab1/Factorial/Factorial.vcxproj

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git commit -m "Add Source.cpp"
[master (root-commit) b2390c5] Add Source.cpp
1 file changed, 12 insertions(+)
create mode 100644 lab1/Factorial/Source.cpp
```

Рис 6. Создание коммитов.

```
MINGW64:/e/Practice/Practice

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        lab1/Factorial/Factorial.vcxproj

nothing added to commit but untracked files present (use "git add" to track)

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git add *

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   lab1/Factorial/Factorial.vcxproj

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git commit -m "Add Factorial.vcxproj"
[master 7e21142] Add Factorial.vcxproj
1 file changed, 147 insertions(+)
create mode 100644 lab1/Factorial/Factorial.vcxproj
```

Рис 7. Создание коммитов.

```
MINGW64:/e/Practice/Practice

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   lab1/Factorial/Source.cpp

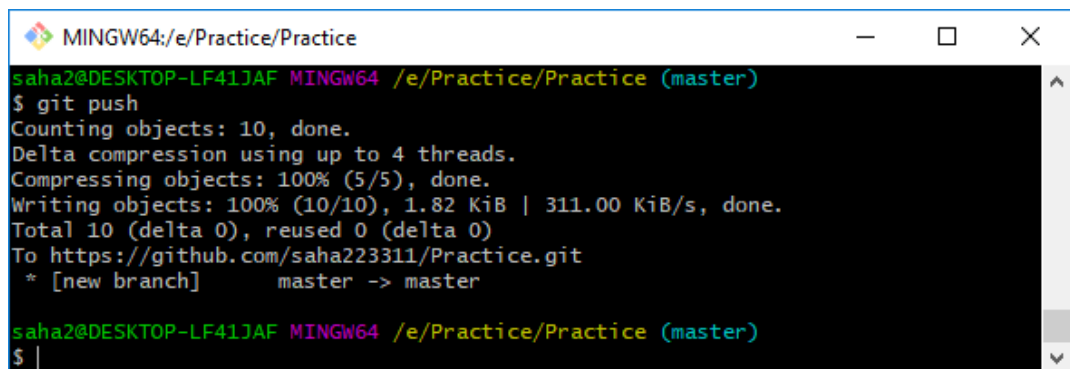
no changes added to commit (use "git add" and/or "git commit -a")

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git add *

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git commit -m "Add comments"
[master 1669c5b] Add comments
1 file changed, 4 insertions(+), 3 deletions(-)

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ |
```

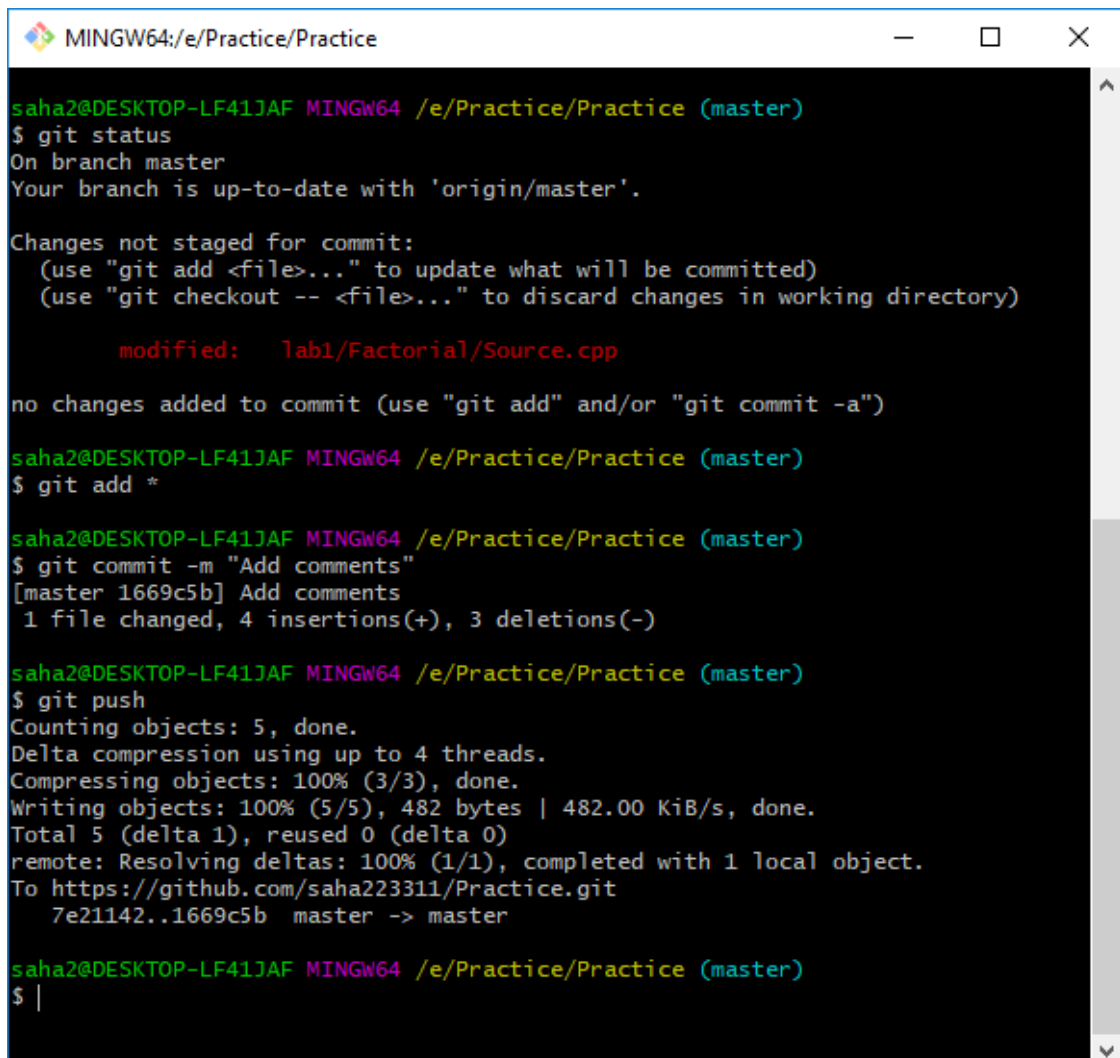
Рис 8. Создание коммитов.



```
MINGW64:/e/Practice/Practice
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git push
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (10/10), 1.82 KiB | 311.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0)
To https://github.com/saha223311/Practice.git
 * [new branch]      master -> master

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ |
```

Рис 9. Выполнение команды push.



```
MINGW64:/e/Practice/Practice
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

       modified:   lab1/Factorial/Source.cpp

no changes added to commit (use "git add" and/or "git commit -a")

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git add *

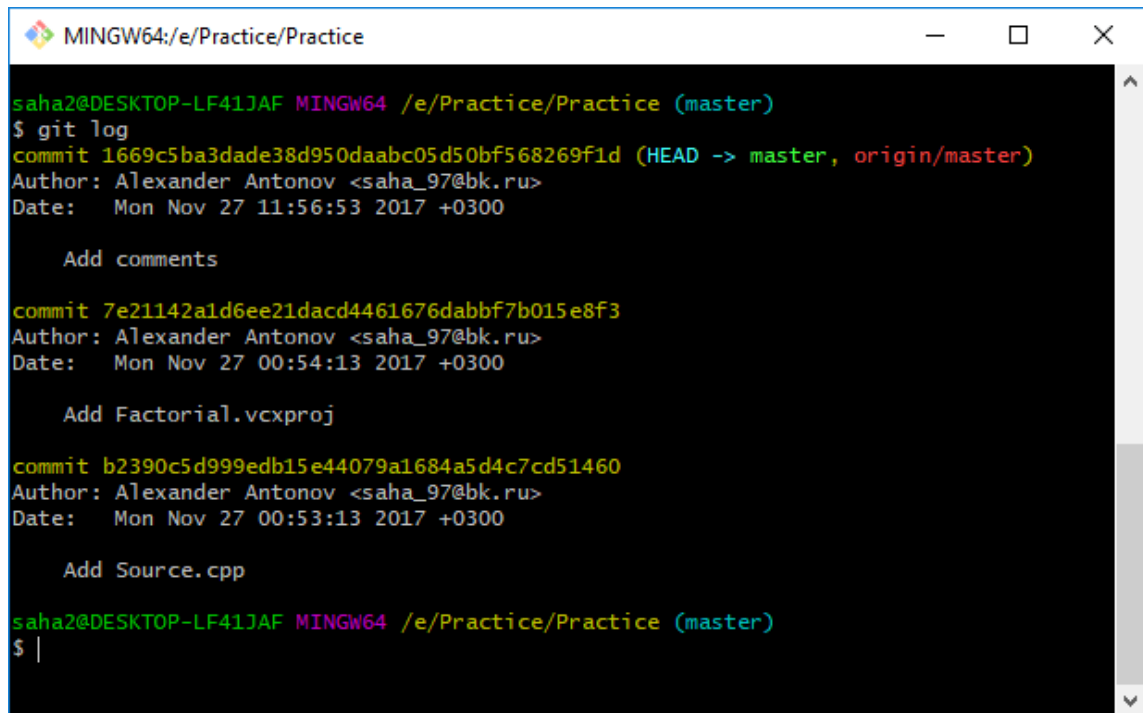
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git commit -m "Add comments"
[master 1669c5b] Add comments
 1 file changed, 4 insertions(+), 3 deletions(-)

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 482 bytes | 482.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/saha223311/Practice.git
 7e21142..1669c5b  master -> master

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ |
```

Рис 10. Создание коммитов.

Приведем результаты работы “git log”.



```
MINGW64:/e/Practice/Practice
saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ git log
commit 1669c5ba3dade38d950daabc05d50bf568269f1d (HEAD -> master, origin/master)
Author: Alexander Antonov <saha_97@bk.ru>
Date: Mon Nov 27 11:56:53 2017 +0300

    Add comments

commit 7e21142a1d6ee21dacd4461676dabbf7b015e8f3
Author: Alexander Antonov <saha_97@bk.ru>
Date: Mon Nov 27 00:54:13 2017 +0300

    Add Factorial.vcxproj

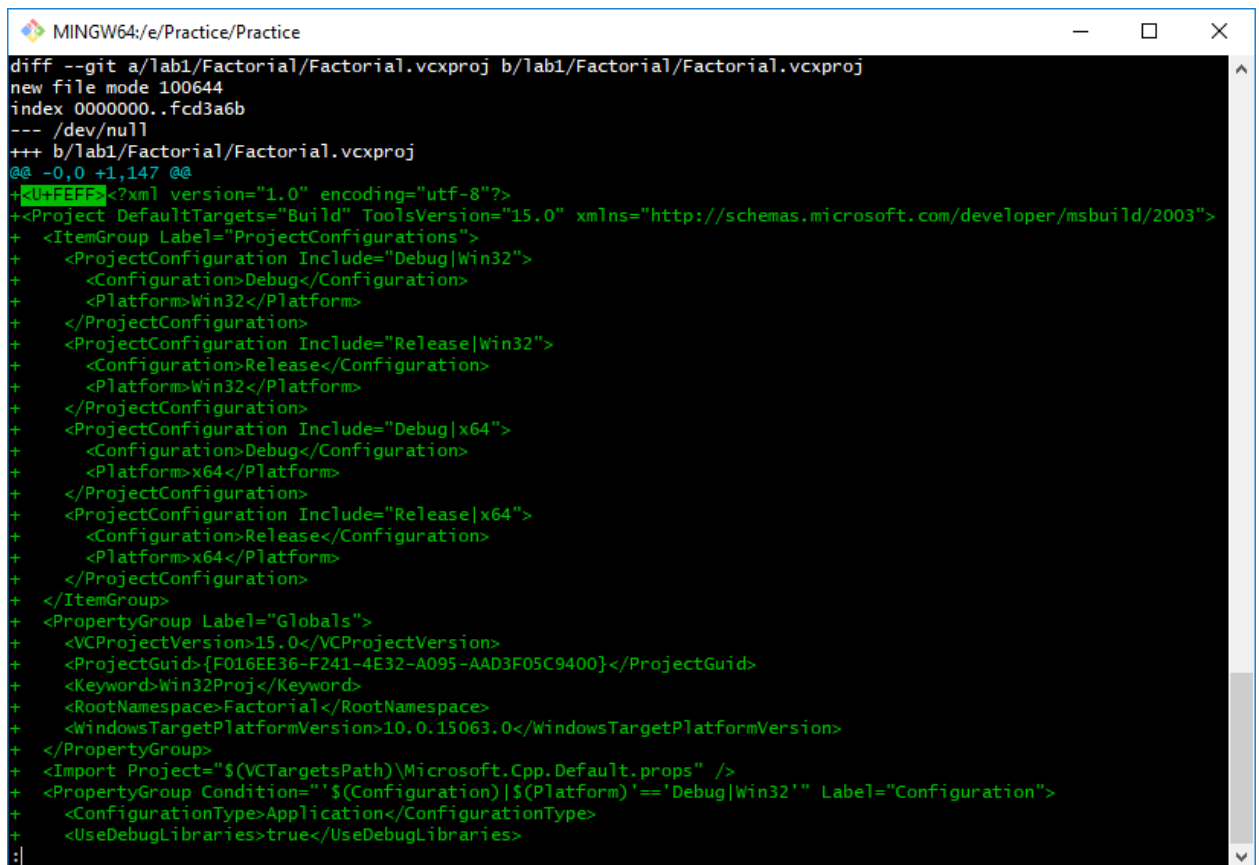
commit b2390c5d999edb15e44079a1684a5d4c7cd51460
Author: Alexander Antonov <saha_97@bk.ru>
Date: Mon Nov 27 00:53:13 2017 +0300

    Add Source.cpp

saha2@DESKTOP-LF41JAF MINGW64 /e/Practice/Practice (master)
$ |
```

Рис 11. Результат работы git log.

Покажем при помощи “git diff <ваш_файл_ваши_коммиты>” изменение любого файла между двумя коммитами.



```
MINGW64:/e/Practice/Practice
diff --git a/lab1/Factorial/Factorial.vcxproj b/lab1/Factorial/Factorial.vcxproj
new file mode 100644
index 0000000..fcd3a6b
--- /dev/null
+++ b/lab1/Factorial/Factorial.vcxproj
@@ -0,0 +1,147 @@
+<?xml version="1.0" encoding="utf-8"?>
+<Project DefaultTargets="Build" ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
+  <ItemGroup Label="ProjectConfigurations">
+    <ProjectConfiguration Include="Debug|Win32">
+      <Configuration>Debug</Configuration>
+      <Platform>Win32</Platform>
+    </ProjectConfiguration>
+    <ProjectConfiguration Include="Release|Win32">
+      <Configuration>Release</Configuration>
+      <Platform>Win32</Platform>
+    </ProjectConfiguration>
+    <ProjectConfiguration Include="Debug|x64">
+      <Configuration>Debug</Configuration>
+      <Platform>x64</Platform>
+    </ProjectConfiguration>
+    <ProjectConfiguration Include="Release|x64">
+      <Configuration>Release</Configuration>
+      <Platform>x64</Platform>
+    </ProjectConfiguration>
+  </ItemGroup>
+  <PropertyGroup Label="Globals">
+    <VCTargetsPath>$(VCTargetsPath)\Microsoft.Cpp.Default.props</VCTargetsPath>
+    <ProjectGuid>{F016EE36-F241-4E32-A095-AAD3F05C9400}</ProjectGuid>
+    <Keyword>Win32Proj</Keyword>
+    <RootNamespace>Factorial</RootNamespace>
+    <WindowsTargetPlatformVersion>10.0.15063.0</WindowsTargetPlatformVersion>
+  </PropertyGroup>
+  <Import Project="$(VCTargetsPath)\Microsoft.Cpp.Default.props" />
+  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'" Label="Configuration">
+    <ConfigurationType>Application</ConfigurationType>
+    <UseDebugLibraries>true</UseDebugLibraries>
```

Рис 12. Результат работы git diff.

Задание №2. Форматирование и стиль

1. Условия задания и контрольный пример.

1881. Длинное условие задачи

Ограничение времени: 0.5 секунды

Ограничение памяти: 64 МБ

Пока Федя писал условие для задачи «Летопись GOV», ему пришло в голову, что бумаги для печати условий может не хватить. А тут ещё оказалось, что в его текстовом редакторе нет возможности определить количество страниц в тексте. Однако Федя не растерялся и решил сам написать программу, рассчитывающую количество страниц по введённому тексту.

Он знал, что на каждой странице h строк, а в каждой строке w символов. Между каждой парой соседних слов в одной строке ставится ровно один пробел. Если слово не помещается в строку, Федя не делит его на слоги для переноса и не ставит пробел, а просто пишет всё слово с начала новой строки.



Исходные данные

В первой строке записаны целые числа h, w, n — количество строк на странице, символов в строке и слов в тексте задачи, соответственно ($1 \leq h, w \leq 100$; $1 \leq n \leq 10\,000$). В следующих n строках записан текст условия, которое написал Федя, по одному слову в строке. Слова в условии непусты, имеют длину не более w и состоят только из строчных и заглавных латинских букв и знаков препинания (точка, запятая, восклицательный и вопросительный знаки). Суммарная длина всех слов не превосходит 10 000.

Результат

Выведите количество страниц в тексте условия задачи.

Пример

исходные данные	результат
3 5 6 To be or not to be	2

Рис 13. Условие задачи и контрольный пример.

2. Схема алгоритма решения задачи.

Был разработан класс `algorithm`, который как раз-таки занимается выполнением поставленной перед нами задачи.

Основную роль играет метод данного класса `SetNewWord`, в котором реализован основной алгоритм. Работа его заключается в том, мы получая новую строку, начинаем ее обрабатывать. А именно, проверяем, если при суммировании новой строки к текущей у нас не происходит выхода из диапазона, то суммируем их. В противном случае увеличим количество строк

на текущей странице которая будет инициализирована новым переданным словом.

Также при увеличении строк, необходимо проверять не произошло ли у нас переполнение строк на данной странице. Если произошло, то создаем новую страницу, на которой инициализируем новую строку равную новой введенной строке.

Результатом задачи будет являться возвращаемое значение метода `GetResult`.

3. Текст программы.

```
#include <iostream>
#include <string>

class algorithm {
public:
    algorithm(int aCountOfString, int aCountOfSymbol)
        : countOfString_(aCountOfString), countOfSymbol_(aCountOfSymbol) {}
    //
    void SetNewWord(std::string aWord) {
        if (currentWord_ + aWord.length() + (currentWord_ > 0 ? 1 : 0) <= countOfSymbol_) {
            currentWord_ += aWord.length() + (currentWord_ > 0 ? 1 : 0);
        }
        else {
            ++currentString_;
            currentWord_ = aWord.length();
        }
    }
    //
    int GetResult() {
        return currentString_ / countOfString_ + ((currentString_ % countOfString_) > 0 ? 1 :
0);
    }
    //
private:
    int countOfString_;
    int countOfSymbol_;
    //
    int currentString_ = 1;
    int currentWord_ = 0;
};

void main() {
    int countOfString, countOfSymbol, numberOfWord;
    std::string currentWord;
    std::cin >> countOfString >> countOfSymbol >> numberOfWord;
    //
    algorithm alg(countOfString, countOfSymbol);
    //
    for (int i = 0; i < numberOfWord; i++) {
        std::cin >> currentWord;
        alg.SetNewWord(currentWord);
    }
}
```

```
//
std::cout << alg.GetResult();
}
```

4. Скриншот с сайта timus.

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
7665942	18:40:07 10 дек 2017	IPRIT	1205. На метро или пешком?	Clang++ 4.0.1	Wrong answer	7	0.015	1 400 КБ
7665941	18:40:07 10 дек 2017	saha223311	1881. Длинное условие задачи	Visual C++ 2017	Accepted		0.015	288 КБ
7665940	18:39:38 10 дек 2017	Razumnyy(MISIS)	1766. Шаптай-Болтай	Python 2.7	Time limit exceeded	1	0.546	1 376 КБ
7665939	18:39:35 10 дек 2017	IPRIT	1471. Расстояние в дереве	Visual C++ 2017	Wrong answer	3	0.001	296 КБ
7665938	18:39:12 10 дек 2017	Hiruzen Sarutobi	1014. Произведение цифр	Clang++ 4.0.1	Wrong answer	3	0.001	452 КБ
7665937	18:39:06 10 дек 2017	Freeza	1014. Произведение цифр	G++ 7.1	Wrong answer	2	0.001	416 КБ
7665936	18:38:16 10 дек 2017	Zhao ZiLong	1014. Произведение цифр	G++ 7.1	Wrong answer	2	0.001	420 КБ
7665935	18:38:16 10 дек 2017	Tenshinhan	1423. Басня о строке	G++ 7.1	Wrong answer	8	0.265	59 868 КБ
7665934	18:37:55 10 дек 2017	IPRIT	1205. На метро или пешком?	Clang++ 4.0.1	Wrong answer	7	0.001	1 400 КБ

Рис 14. Скриншот результата проверки программы с сайта timus.

Задание №3. TDD: Разработка через тестирование.

1) Условия задания и пример.

1635. Мнемоника и палиндромы

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Студенту Васечкину чудовищно не повезло на экзамене. Из 42 билетов он не подготовил только последний, и ему достался именно этот билет. Теперь же Васечкин сидел перед преподавателем и не мог ответить ни на один вопрос. Однако преподаватель пребывал в весьма добром расположении духа и решил дать Васечкину последний шанс на тройку. Он спросил его, как называется предмет, который Васечкин сейчас сдаёт. К несчастью, Васечкин не смог вспомнить точную аббревиатуру, припоминая лишь, что в названии фигурировала чья-то безопасность, какие-то программы и аппараты и, кажется, информатика...

К передаче Васечкин решил выучить название предмета. Чтобы лучше запомнить эту длинную строку, он решил разбить её на палиндромы и запомнить каждый из них по отдельности. Конечно, количество палиндромов в разбиении должно быть минимально возможным.

Исходные данные

В первой строке записано название предмета, который сдавал Васечкин, — непустая строка, содержащая только маленькие латинские буквы. Длина строки не превосходит 4000 символов.

Результат

В первой строке выведите минимальное количество палиндромов, на которое можно разбить название предмета. Во второй строке через пробел выведите палиндромы из оптимального разбиения. Если возможных ответов несколько, выведите любой.

Примеры

исходные данные	результат
pasoib	6 p a s o i b
zzzqxx	3 z z z q x x
wasitacatisaw	1 wasitacatisaw

Рис 15. Условие задачи и контрольный пример.

2) Схема алгоритма решения задачи.

Результат работы алгоритма можно сказать заключается в простом переборе. Мы проходимся в цикле по каждому символу и от него в обе стороны идем и сравниваем на равенство другие получившиеся символы. Причем будет рассмотрено два варианта. Первый – с учетом того что получившийся палиндром имеет нечетное количество символов, второй – четное. Таким образом, выбираем палиндромы из нашей строки.

3) Перечисление возможных ошибочных ситуаций и их обработка.

- ввод строки, в которой количество символов больше 4000.

Обработка с помощью проверки на длину вводимой строки.

- ввод строки, в которой задана пустая строка.

Обработка строки с помощью проверки на то, пустая она или нет.

- нарушение следующего правила: ввод строки, содержащего только маленькие латинские буквы.

Обработка строки с помощью проверки каждого символа.

4) Скриншот результата с timus.

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
7675617	16:33:03 17 дек 2017	saha223311	1635. Мнемоника и палиндромы	Visual C++ 2017	Accepted		0.156	40 428 КБ
7675616	16:32:51 17 дек 2017	kisma	1502. Точки домино	Python 3.6	Wrong answer	1	0.062	228 КБ
7675615	16:32:26 17 дек 2017	kisma	1502. Точки домино	Python 3.6	Wrong answer	1	0.078	228 КБ
7675614	16:31:08 17 дек 2017	AlexandrVoynov'~	1219. Symbolic Sequence	G++ 7.1	Accepted		0.062	348 КБ
7675613	16:29:23 17 дек 2017	Ilia	1111. Квадраты	G++ 7.1	Wrong answer	2	0.001	456 КБ
7675612	16:28:22 17 дек 2017	ShoshinNikita	1131. Копирование	Visual C++ 2017	Accepted		0.001	276 КБ
7675611	16:28:14 17 дек 2017	Faris NyanNyan	2015. Женя переезжает из общежития	G++ 7.1	Wrong answer	5	0.015	416 КБ
7675610	16:27:35 17 дек 2017	Bulat	1355. К вопросу о лысине	Visual C++ 2017	Accepted		0.001	180 КБ

Рис 16. Скриншот результата проверки программы с сайта timus.

5) Скриншот с покрытием кода.

LCOV - code coverage report

Current view: top level - /e/Practice/laba3QT				Hit		Total	Coverage
Test: coverage.info				Lines:	55	55	100.0 %
Date: 2017-12-25 17:42:08				Functions:	0	0	-
				Branches:	0	0	-
Filename	Line Coverage		Functions	Branches			
algorithm.cpp	<div></div>	100.0 %	42 / 42	-	0 / 0	-	0 / 0
algorithm.h	<div></div>	100.0 %	2 / 2	-	0 / 0	-	0 / 0
main.cpp	<div></div>	100.0 %	11 / 11	-	0 / 0	-	0 / 0

Generated by: [LCOV version 1.0](#)

Рис 17. Скриншот результата проверки покрытия кода с сайта timus.

Результат покрытия основного кода данной программы.

LCOV - code coverage report

Current view: top level - e/Practice/laba3QT - algorithm.cpp (source / functions)		Hit	Total	Coverage
Test: coverage.info		Lines:	42	42
Date: 2017-12-25 17:42:08		Functions:	0	0
		Branches:	0	0

Branch data	Line data	Source code
1	:	#include "algorithm.h"
2	:	:
3	:	7 : std::string algorithm::Execute(std::string aLine) {
4	:	14 : std::string error = this->CheckData(aLine);
5	:	:
6	:	7 : if (error != "done"){
7	:	4 : return error;
8	:	}
9	:	7 : }
10	:	3 : line = aLine;
11	:	3 : result = "";
12	:	3 : int length = line.length();
13	:	6 : std::vector<int> palin(4000);
14	:	int dp[4001], next[4000];
15	:	7 : }
16	:	28 : for (int i = 0; i < length; ++i) {
17	:	25 : int pos1 = i, pos2 = i;
18	:	7 : }
19	:	89 : while (pos1 >= 0 && pos2 < length && line[pos1] == line[pos2]) {
20	:	32 : palin[pos1].push_back(pos2);
21	:	--pos1; ++pos2;
22	:	}
23	:	7 : }
24	:	25 : pos1 = i; pos2 = i + 1;
25	:	31 : while (pos1 >= 0 && pos2 < length && line[pos1] == line[pos2]) {
26	:	3 : palin[pos1].push_back(pos2);
27	:	--pos1; ++pos2;
28	:	}
29	:	7 : }
30	:	3 : }
31	:	3 : dp[length] = 0;
32	:	28 : for (int i = length - 1; i >= 0; --i) {
33	:	25 : dp[i] = length - i;

Рис 18. Скриншот результата проверки покрытия кода с сайта *timus*.

6) Текст программы.

Файл main.cpp

```
#include <string>
#include "algorithm.h"

#define TAP_COMPILE
#include "libtap\cpp_tap.h"

int main() {
    //
    algorithm alg;
    //
    plan_tests(7);
    //
    ok(alg.Execute("pasuib") == "p a s o i b", "Result of \"pasuib\" = \"p a s o i b\"");
    ok(alg.Execute("zzzqxx") == "zzz q xx", "Result of \"zzzqxx\" = \"zzz q xx\"");
    ok(alg.Execute("wasitacatisaw") == "wasitacatisaw", "Result of \"wasitacatisaw\" = \"wasitacatisaw\"");
    //
    ok(alg.Execute(std::string("a", 5000)) == "error1", "Result of std::string(\"a\", 5000) = error1");
    ok(alg.Execute("") == "error2", "Result of \"\" = error2");
    ok(alg.Execute("pas0ib") == "error3", "Result of \"pas0ib\" = error3");
    ok(alg.Execute("zzzqxx44") == "error3", "Result of \"zzzqxx44\" = error3");
    //
    return exit_status();
    return 0;
}
```


Файл algorithm.h

```
#pragma once

#include <string>
#include <vector>

class algorithm {
public:
    algorithm() = default;
    ~algorithm() = default;
    //
    std::string Execute(std::string aLine);
    std::string CheckData(std::string aLine);
private:
    int n_;
    std::string line_;
    std::string result_;
};
```

Файл algorithm.cpp

```
#include "algorithm.h"

std::string algorithm::Execute(std::string aLine) {
    std::string error = this->CheckData(aLine);
    //
    if (error != "done") {
        return error;
    }
    //
    line_ = aLine;
    result_ = "";
    int length = line_.length();
    std::vector<int> palin[4000];
    int dp[4001], next[4000];
    //
    for (int i = 0; i < length; ++i) {
        int pos1 = i, pos2 = i;
        //
        while (pos1 >= 0 && pos2 < length && line_[pos1] == line_[pos2]) {
            palin[pos1].push_back(pos2);
            --pos1; ++pos2;
        }
        //
        pos1 = i; pos2 = i + 1;
        while (pos1 >= 0 && pos2 < length && line_[pos1] == line_[pos2]) {
            palin[pos1].push_back(pos2);
            --pos1; ++pos2;
        }
    }
    //
    dp[length] = 0;
    for (int i = length - 1; i >= 0; --i) {
        dp[i] = length - i;
        next[i] = i + 1;
        //
        for (int j = palin[i].size() - 1; j >= 0; --j) {
            if (dp[i] > 1 + dp[palin[i][j] + 1]) {
                dp[i] = 1 + dp[palin[i][j] + 1];
            }
        }
    }
}
```

```

        next[i] = palin[i][j] + 1;
    }
}
//
n_ = dp[0];
int pos = 0;
while (pos != length) {
    if (pos != 0) result_ += " ";
    for (int i = pos; i < next[pos]; ++i){
        result_ += line_[i];
    }
    pos = next[pos];
}
//
return result_;
}

std::string algorithm::CheckData(std::string aLine){
    if (aLine.length() > 4000){
        return "error1";
    }
    //
    if (aLine.empty()){
        return "error2";
    }
    //
    for (int i = 0; i < aLine.length(); i++){
        if (aLine[i] < 97 || aLine[i] > 122){
            return "error3";
        }
    }
    //
    return "done";
}

```

Отчет по практической работе.

Назначение программного обеспечения

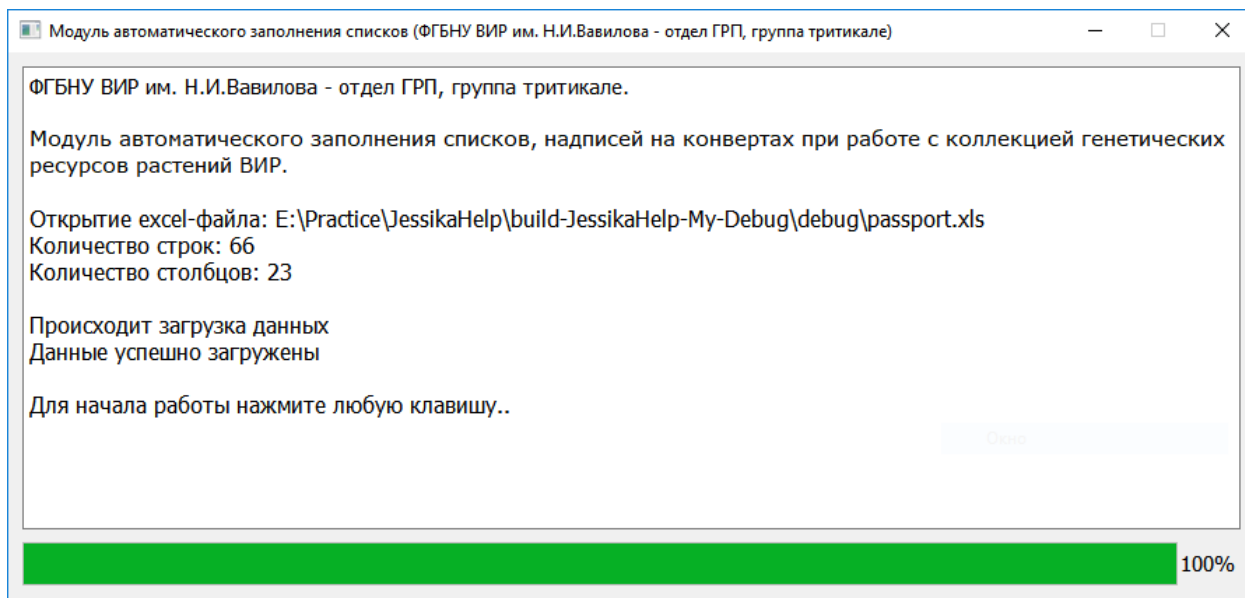
Программное обеспечение предназначено для составления списка образцов и печати этикеток, малых и больших конвертов, используемых при упаковке зерна.

Возможности ПО

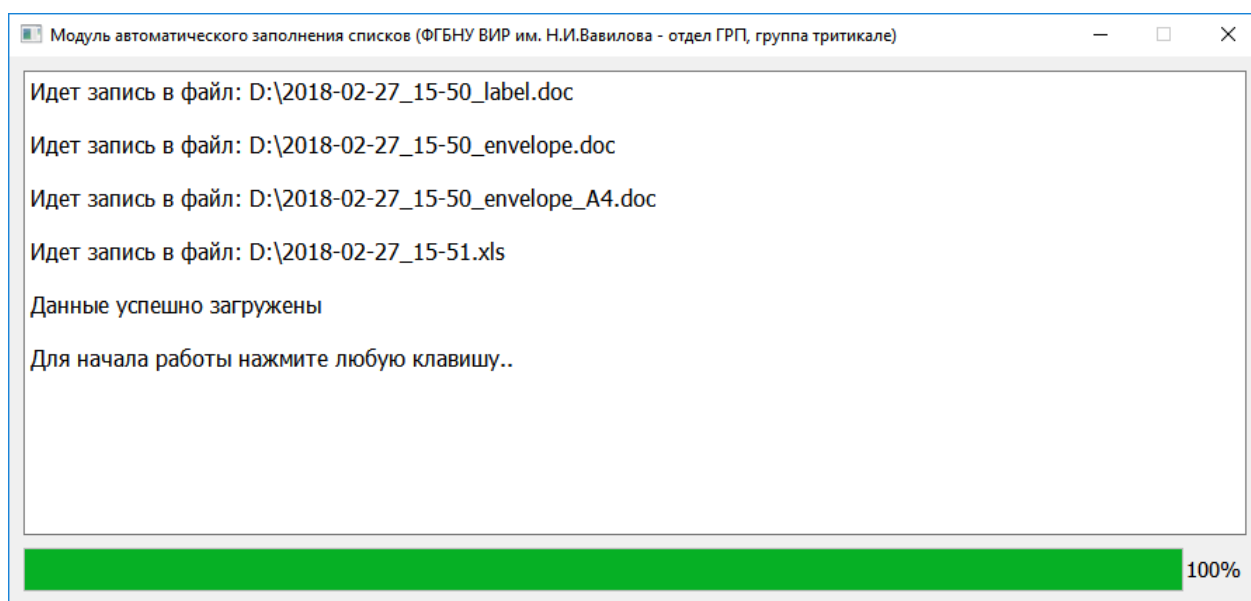
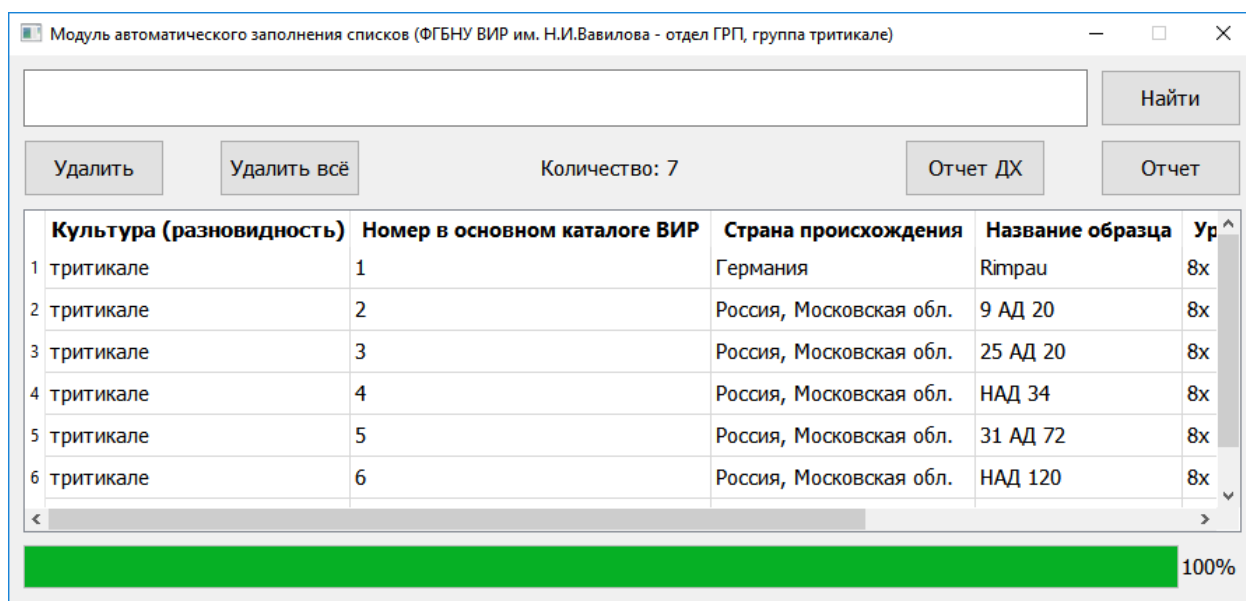
- составление списка выбранных образцов по указанным номерам из общего списка образцов
- формирование для составленного списка doc-файлов (формат ms word), для последующей печати этикеток, малых и больших конвертов

Описание ПО

Программное обеспечение работает с общим списком образцов, хранящемся в файле с именем «passport.xls» (формат ms excel). После запуска ПО начинает считывать общий список образцов. После загрузки общего списка ПО имеет вид, представленный на рисунке.



При работе ПО имеет вид, представленный на рисунке ниже:



Для **добавления** образца необходимо указать его номер в каталоге ВИР и нажать кнопку «Найти».

Для **удаления** неверно добавленного образца необходимо выбрать этот образец в списке и нажать кнопку «Удалить».

Для **формирования списка** образцов необходимо нажать кнопку «Отчет», при этом будут сформированы следующие 4 файла:

- 2014-12-28_20-59.xls – список выбранных образцов (формат ms excel).
Имя файла составлено в формате «год-месяц-число_час-минута».

- 2014-12-28_20-59_envelope.doc – файл (формат ms word), содержащий раскройку малых конвертов (по два конверта на лист A4) для образцов из списка (см. рис 1).
- 2014-12-28_20-59_envelope_A4.doc - файл (формат ms word), содержащий раскройку больших конвертов (один конверта на лист A4) для образцов из списка (см. рис 1).
- 2014-12-28_20-59_label.doc – файл (формат ms word), содержащий этикетки (до 24х этикеток на лист A4) для образцов из списка.

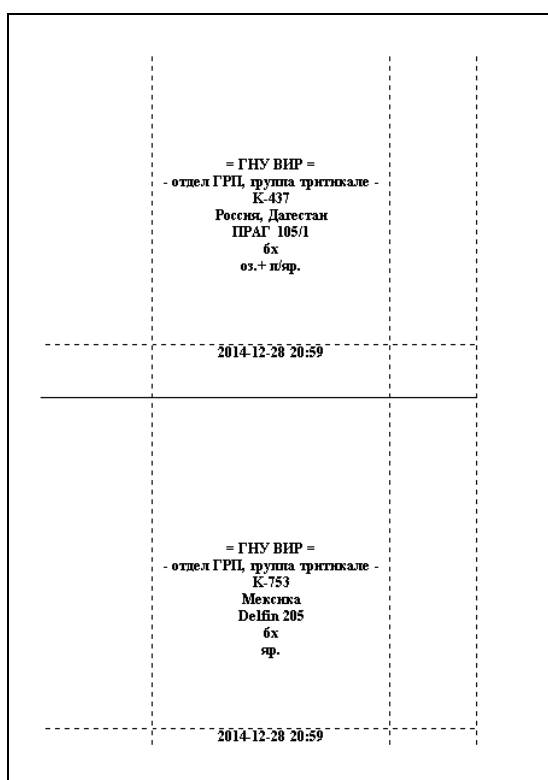


Рис. - 1 малых конверта
на листе A4

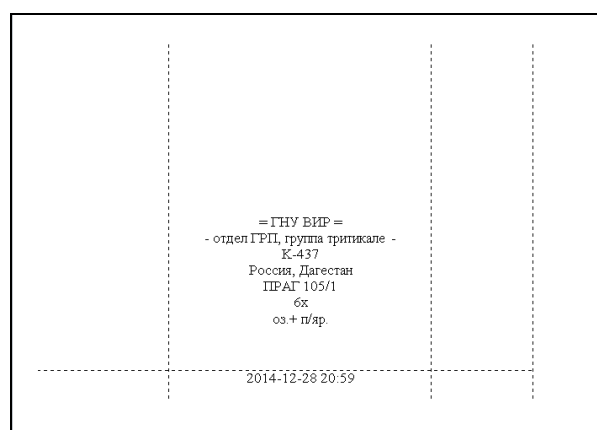


Рис. - 2 Большой конверт на листе
A4

Технические сведения

Программное обеспечение разработано в среде Qt Creator 4.0.2 на языке программирования C++.

Приложение. Презентация.



ITMO UNIVERSITY

```
int a[10] = { 1, 2, 3, 4, 5, 6, };
```

ITMO *is* than a
UNIVERSITY

ITMO UNIVERSITY

```
struct A {  
    int a;  
    int b;  
};  
  
class B {  
public:  
    int a;  
    int b;  
};  
  
union C {  
    int a;  
    int b;  
};  
  
int main() {  
    A a = { 1, 2 };  
    B b = { 2, 3 };  
    C c = { 1 };  
}
```

ITMO *is* than a
UNIVERSITY

```
std::vector<int> vec;  
vec.push_back(1);  
vec.push_back(7);  
vec.push_back(-3);  
vec.push_back(4);  
vec.push_back(-15);  
vec.push_back(22);  
...
```

```
int a[] = { 1, 7, -3, 4, -15, 22 ... };  
std::vector<int> vec(&a[0], &a[sizeof(A) / sizeof(int) - 1]);
```

```
class A{  
public:  
    A() : m_i(7) {}  
private:  
    int m_i;  
};  
  
int main(){  
    A a();  
}
```

```
class A{
public:
    A() : m_i(7) {}
private:
    int m_i;
};

int main(){
    A a{};
    A b = {};
};
```

```
class A {
public:
    A(int i, int j, int k) : m_i(i), m_j(j), m_k(k){ }
private:
    int m_i;
    int m_j;
    int m_k;
};

int main() {
    A a{ 1, 2, 3 }; //Эквивалентно A a(1, 2, 3)
};
```

```
A foo() {
    return { 1, 2, 3 }; //Эквивалентно return A(1, 2, 3)
}
```



```
void foo(A a) {
    a;
}

int main() {
    foo({ 1, 2, 3 }); //Эквивалентно foo(A(1, 2, 3))
};
```

```
int* a = new int[5]{ 1, 2, 3, 4, 5 };
```

- 1) Число с плавающей точкой в целое.
- 2) Из *long double* в *double*, или *float* и из *double* во *float*, за исключением случаев, когда это константное значение и оно умещается в “суженный” тип.
- 3) Из целого числа(или *enum*) в число с плавающей запятой, за исключением случаев, когда это константное значение и оно умещается в тип с плавающей точкой.
- 4) Из целого числа(или *enum*) более высокого порядка, в целое число более низкого(например из *long long* в *long*), за исключением случаев, когда это константное значение и оно умещается в “суженный” тип.

Список инициализации

`std::initializer_list<T>`

ITMO is more than a
UNIVERSITY

```
std::vector<int> vec = { 1, 2, 3, 4, 5, 6 };

std::map<std::string, int> map =
{ { "first", 1 }, { "second", 2 }, { "third", 3 } };
```

ITMO is more than a
UNIVERSITY

```
std::vector(const std::initializer_list<T>& list) {
    ....
}

или

std::vector(std::initializer_list<T> list) {
    ....
}
```

ITMO is more than a
UNIVERSITY

```
#include <iostream>
#include <initializer_list>

class A {
public:
    A(std::initializer_list<int> list) {
        for (auto& item : list)
            std::cout << "item=" << item << "\n";
    }
};

int main() {
    A a{ 23,321,321,3,213,213,12 };
}
```

Какой конструктор будет вызван?

```
struct S {
    S(std::initializer_list<int>); // 1
    S(double, double);           // 2
};

int main() {
    S { 1., 2. };
}
```

1) 1 2) 2 3) not compile

Какой конструктор будет вызван?

```
struct S {
    S(std::initializer_list<int>); // 1
    S(double, double);           // 2
};

int main() {
    S { 1., 2. };
}
```

1) 1 2) 2 3) not compile

Какой тип имеют x и y ?

```
auto x = { 0 };
auto y { 0 };
```

- | | |
|---|------------------------------------|
| 1) x - initializer_list
y - initializer_list | 3) x - int
y - int |
| 2) x - initializer_list
y - int | 4) x - int
y - initializer_list |

ITMO More than a
UNIVERSITY

Какой тип имеют x и y ?

```
auto x = { 0 };
auto y { 0 };
```

- | | |
|---|------------------------------------|
| 1) x - initializer_list
y - initializer_list | 3) x - int
y - int |
| 2) x - initializer_list
y - int | 4) x - int
y - initializer_list |

ITMO More than a
UNIVERSITY

Скомпилируется ли код?

```
std::vector<std::vector<int>> v1{ { { { {} } } } } };
```

- 1) да 2) нет

ITMO More than a
UNIVERSITY

Скомпилируется ли код?

```
std::vector<std::vector<int>> v1{ { { { } } } } };
```

1) да 2) нет

IT,MOre than a
UNIVERSITY

Спасибо

www.ifmo.ru

IT,MOre than a
UNIVERSITY