

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:

Скрипачев И.С.

Группа: V3316

Преподаватель:

Кудрявцев А. С.

Санкт-Петербург, 2017

Оглавление

Цель проведения практики.....	3
Задание №1. Знакомство с системой контроля версий Git	4
Задание №2. «Форматирование и стиль».....	11
Задание №3. «Тестирование.....	14
Проектная работа. Приложение-виджет для андроид.....	18
Приложение А. Презентация с семинара по C++11 на тему: Tuple and tie.....	23

Цель проведения практики

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

Задание №1. Знакомство с системой контроля версий Git

1. Завести аккаунт на github.com. На рис.1 представлен скриншот зарегистрированного пользователя.

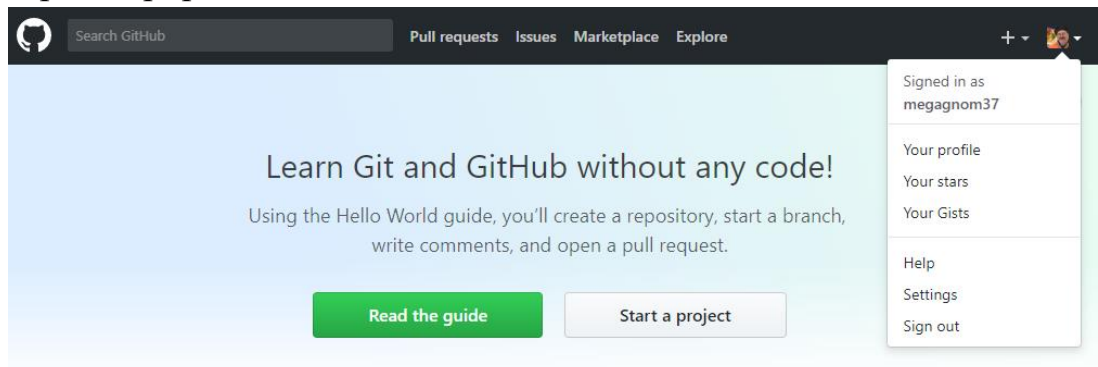


Рисунок 1- аккаунт на github.com

2. Создать репозиторий для лабораторных работ. На Рис.2 представлен процесс создания репозитория и его первоначальная настройка. Рис.3 демонстрирует наличие созданного репозитория.

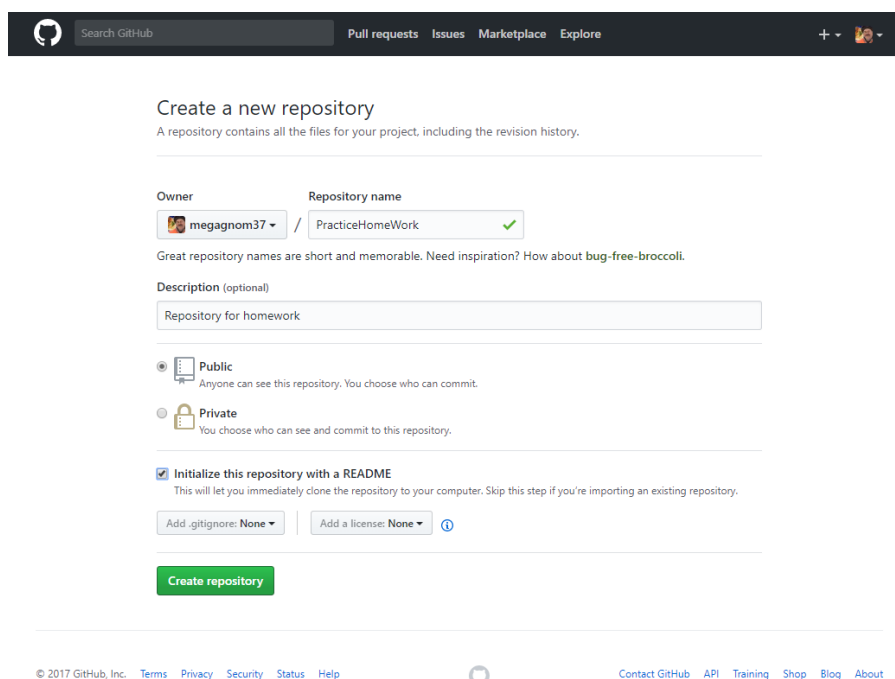


Рисунок 2-создание репозитория

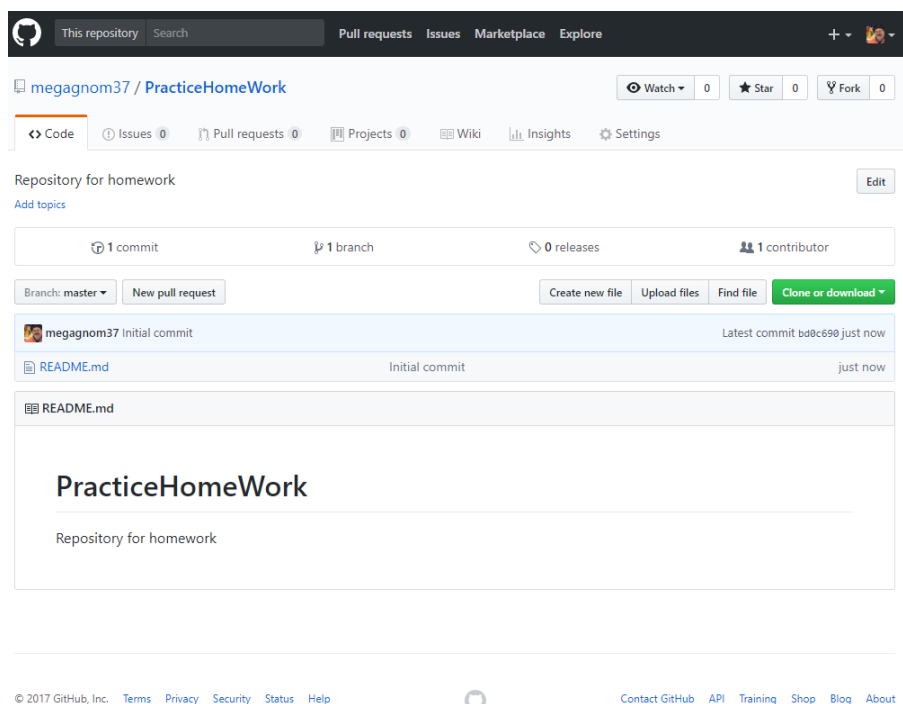


Рисунок 3-скриншот созданного репозитория

3. Склонировать репозиторий на компьютер. На Рис.4 можно видеть, что для клонирования репозитория нужно иметь ссылку на него. Копируем и в консоли с помощью команды `git clone <LINK>` клонируем репозиторий в нужное нам место Рис. 5.

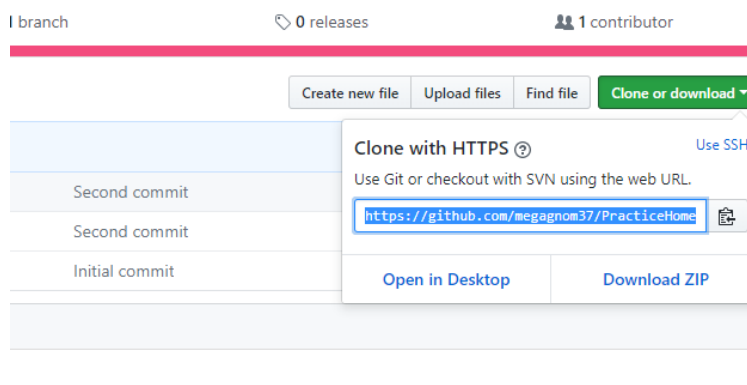


Рисунок 4-ссылка на репозиторий

```

MINGW64:/c/temp/work
Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work
$ git clone https://github.com/megagnom37/PracticeHomeWork.git
Cloning into 'PracticeHomeWork'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work
$

```

Рисунок 5-команда `git clone` <LINK>

4. Создать папку lab1. Создаем папку с именем lab1

```

MINGW64:/c/temp/work/PracticeHomeWork
Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work
$ git clone https://github.com/megagnom37/PracticeHomeWork.git
Cloning into 'PracticeHomeWork'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work
$ cd PracticeHomeWork/

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ mkdir lab1

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ |

```

Рисунок 6-создание папки

5. Написать в папке lab1 программу, вычисляющую функцию факториала.

```

// Source.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int f_factor(int num) {
6     return (num != 1) ? (f_factor(num - 1)*num) : 1;
7 }
8

```

Вывод

```

Показать выходные данные из: Сборка
1>factorial.vcxproj -> C:\temp\work\PracticeHomeWork\lab1\factorial\Debug\fact
===== Сборка: успешно: 1, с ошибками: 0, без изменений: 0, пропущено: 0 =

```

Рисунок 8-функция нахождения факториала

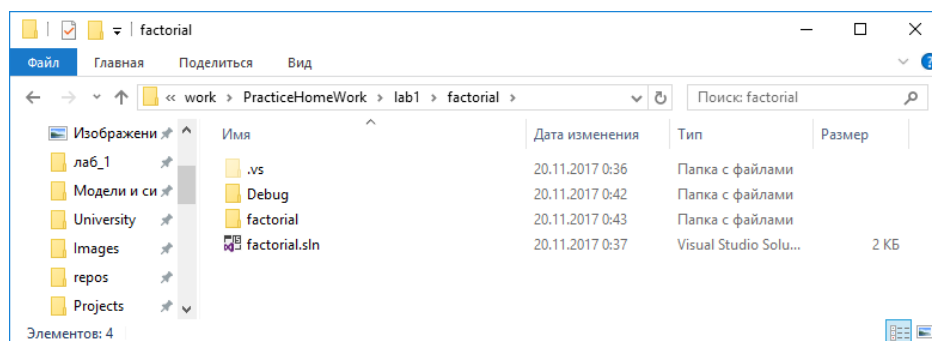


Рисунок 9-директория с созданной программой

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.

На рис.10 показан созданный файл `.gitignore`. На Рис. 11 видно содержимое данного файла с игнорируемыми расширениями.

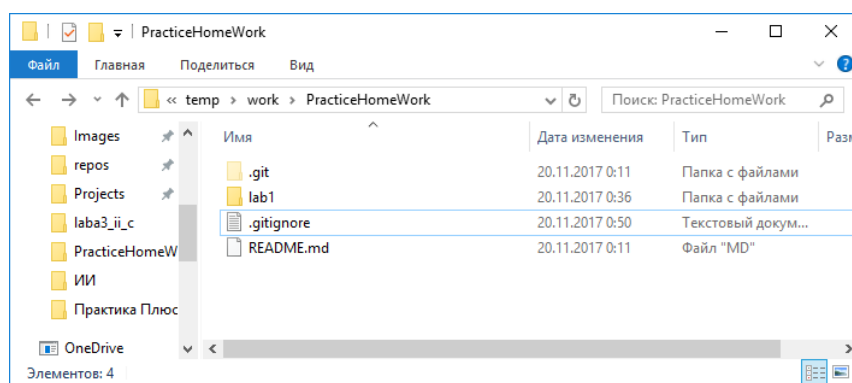


Рисунок 10-файл .gitignore

```

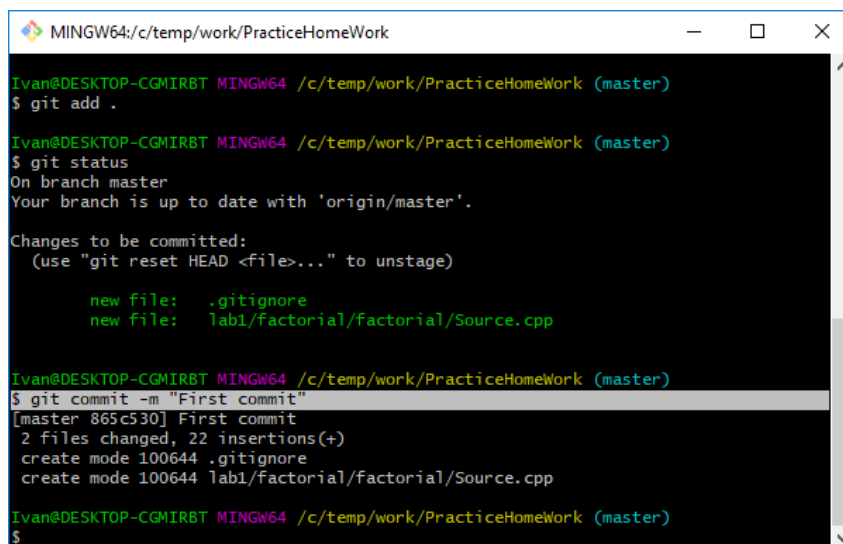
Файл  Правка  Формат  Вид  Справка
*.ipch
*.suo
*.VC.db
*.exe
*.ilk
*.pdb
*.vcxproj.filters
*.log
*.obj
*.idb
*.pdb
*.tlog
*lastbuildstate

```

Рисунок 11-содержимое файла .gitignore

7. Сделать по крайней мере два коммита.

Рис.12 commit до изменения файла с программой.



```

MINGW64:/c/temp/work/PracticeHomeWork
Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git add .

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

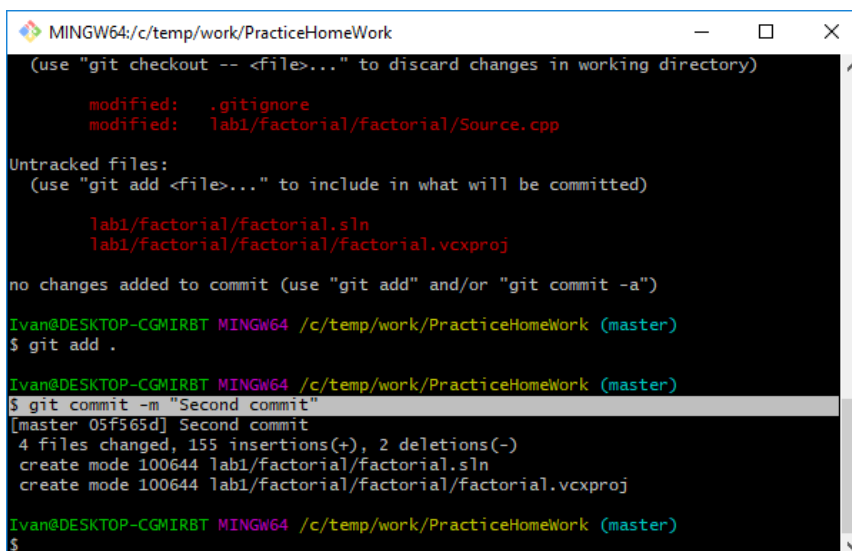
        new file:   .gitignore
        new file:   lab1/factorial/factorial/Source.cpp

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git commit -m "First commit"
[master 865c530] First commit
 2 files changed, 22 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 lab1/factorial/factorial/Source.cpp

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$
  
```

Рисунок 12- первый commit

Рис.13 commit после внесения изменений в файл с программой.



```

MINGW64:/c/temp/work/PracticeHomeWork
(use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .gitignore
        modified:   lab1/factorial/factorial/Source.cpp

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

        lab1/factorial/factorial/sln
        lab1/factorial/factorial/factorial.vcxproj

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

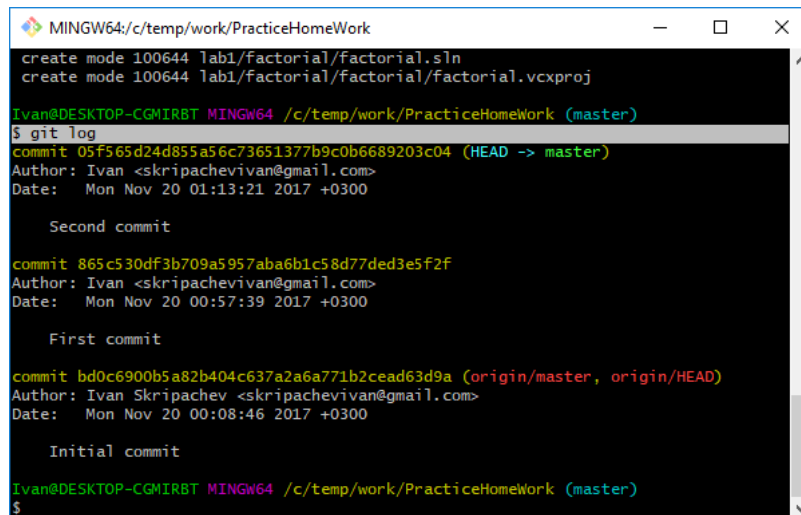
Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git add .

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git commit -m "Second commit"
[master 05f565d] Second commit
 4 files changed, 155 insertions(+), 2 deletions(-)
 create mode 100644 lab1/factorial/factorial/sln
 create mode 100644 lab1/factorial/factorial/factorial.vcxproj

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$
  
```

Рисунок 13- второй commit

8. Привести результаты работы “git log”. Рис.14 показывает нам результат работы команды git log, а именно информацию о первоначальном инициализирующем ‘commit’ и двух последующих, добавленных нами вручную.



```
MINGW64/c/temp/work/PracticeHomeWork
create mode 100644 lab1/factorial/factorial.sln
create mode 100644 lab1/factorial/factorial/factorial.vcxproj

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git log
commit 05f565d24d855a56c73651377b9c0b6689203c04 (HEAD -> master)
Author: Ivan <skripachevivan@gmail.com>
Date: Mon Nov 20 01:13:21 2017 +0300

    Second commit

commit 865c530df3b709a5957aba6b1c58d77ded3e5f2f
Author: Ivan <skripachevivan@gmail.com>
Date: Mon Nov 20 00:57:39 2017 +0300

    First commit

commit bd0c6900b5a82b404c637a2a6a771b2cead63d9a (origin/master, origin/HEAD)
Author: Ivan Skripachev <skripachevivan@gmail.com>
Date: Mon Nov 20 00:08:46 2017 +0300

    Initial commit

Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$
```

Рисунок 14- команда git log

9. Показать при помощи “git diff” изменение файлов между двумя коммитами. Результат выполнения команды показан на рис.15.

```

MINGW64; c:/temp/work/PracticeHomeWork
Ivan@DESKTOP-CGMIRBT MINGW64 /c/temp/work/PracticeHomeWork (master)
$ git diff 865c530df3b709a5957aba6b1c58d77ded3e5f2f bd0c6900b5a82b404c637a2a6a771b2
cead63d9a
diff --git a/.gitignore b/.gitignore
deleted file mode 100644
index 5b5fa6b..0000000
--- a/.gitignore
+++ /dev/null
@@ -1,15 +0,0 @@
-*.ipch
-*.suo
-*.VC.db
-*.exe
-*.ilk
-*.pdb
-*.sln
-*.vcxproj
-*.vcxproj.filters
-*.log
-*.obj
-*.idb
-*.pdb
-*.tlog
-*.lastbuildstate
\ No newline at end of file
diff --git a/lab1/factorial/factorial/Source.cpp b/lab1/factorial/factorial/Source.cpp
deleted file mode 100644
index 270919d..0000000
--- a/lab1/factorial/factorial/Source.cpp
+++ /dev/null
@@ -1,7 +0,0 @@
-#include <iostream>
-...skipping...
diff --git a/.gitignore b/.gitignore
deleted file mode 100644
index 5b5fa6b..0000000
--- a/.gitignore
+++ /dev/null
@@ -1,15 +0,0 @@
-*.ipch
-*.suo
-*.VC.db
-*.exe
-*.ilk
-*.pdb
-*.sln
-*.vcxproj
-*.vcxproj.filters
-*.log
-*.obj
-*.idb
-*.pdb
-*.tlog
-*.lastbuildstate
\ No newline at end of file
diff --git a/lab1/factorial/factorial/Source.cpp b/lab1/factorial/factorial/Source.cpp
deleted file mode 100644
index 270919d..0000000
--- a/lab1/factorial/factorial/Source.cpp
+++ /dev/null
@@ -1,7 +0,0 @@
-#include <iostream>
-
-using namespace std;
-
-int f_factor(int num) {
-    return (num != 1) ? (f_factor(num - 1)*num) : 1;
-}

```

Рисунок 75- команда git diff

Задание №2. «Форматирование и стиль»

1) Условия задания:

1014. Произведение цифр

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Ваша задача — найти минимальное положительное целое число Q такое, что произведение цифр числа Q в точности равняется N .

Исходные данные

Целое число N ($0 \leq N \leq 10^9$).

Результат

Выведите целое число Q . Если такого числа не существует, выведите -1 .

Пример

исходные данные	результат
10	25

2) Описание алгоритма:

На каждом шаге цикла пытаемся сократить значение (входное) на максимальный делитель (максимальная цифра – 9). Если число делится на него без остатка, то к текущему полученному числу Q добавляем недостающее число с умножением на коэффициент разрядности. Обновляем этот коэффициент. Меняем значение текущего числа на его целую часть от деления. Если в результате переданное значение равно 1, то возвращаем найденное Q . Иначе возвращаем -1 (по условию).

В начале метода проверяем и обрабатываем исключительные ситуации:

- Переданное число равно 0 (возвращаем 10)
- Переданное число равно 1 (возвращаем 1)

3) Текст программы:

```
#include <iostream>

long long MulOfNumbers(long long aNumber) {
    if (aNumber == 0) {
        return 10;
    } else if (aNumber == 1) {
        return 1;
    }
    //
    long long q(0);
    long long multiplier(1);
    //
    for (long long i(9); i > 1; --i) {
        while ((aNumber % i) == 0) {
            q = q + (multiplier * i);
            multiplier = multiplier * 10;
            aNumber = aNumber / i;
        }
    }
    //
    return (aNumber == 1) ? q : -1;
}

int main() {
    long long n;
    std::cin >> n;
    std::cout << MulOfNumbers(n) << std::endl;
    return 0;
}
```

4) Исходные данные, используемые при вводе в программу и результат выполнения программы:

Таблица 1. Значения входных и выходных данных.

Исходные данные	Результат
0	10
1	1
6	6
10	25

120	358
333	-1
557	-1
1678	-1
150000	5555568
1000000000	555555555888

5) Отчет о результатах работы:

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
7648723	16:53:02 3 дек 2017	ISkripachev	1014. Произведение цифр	Visual C++ 2017	Accepted		0.015	288 КБ

Рисунок 16. Результат проверки задания на timus.

Задание №3. «Тестирование

1) Условия задания:

1025. Демократия в опасности

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Вступление

В одном из островных государств Карибского бассейна все решения традиционно принимались простым большинством голосов на общем собрании граждан, которых, к счастью, было не очень много. Одна из местных партий, стремясь прийти к власти как можно более законным путем, смогла добиться некоторой реформы избирательной системы. Главным аргументом было то, что население острова в последнее время значительно возросло, и проведение общих собраний перестало быть легкой задачей.

Суть реформы состояла в следующем: с момента введения ее в действие все избиратели острова делились на K групп (необязательно равных по численности). Голосование по любому вопросу теперь следовало проводить отдельно в каждой группе, причем считалось, что группа высказывается «за», если «за» голосует более половины людей в этой группе, а в противном случае считалось, что группа высказывается «против». После проведения голосования в группах подсчитывалось количество групп, высказавшихся «за» и «против», и вопрос решался положительно в том и только том случае, когда групп, высказавшихся «за», оказывалось более половины общего количества групп.

Эта система вначале была с радостью принята жителями острова. Когда первые восторги рассеялись, очевидны стали, однако, некоторые недостатки новой системы. Оказалось, что сторонники партии, предложившей систему, смогли оказать некоторое влияние на формирование групп избирателей. Благодаря этому, они получили возможность проводить некоторые решения, не обладая при этом реальным большинством голосов!

Пусть, например, на острове были сформированы три группы избирателей, численностью в 5, 5 и 7 человек соответственно. Тогда партии достаточно иметь трех сторонников в каждой из первых двух групп, и она сможет провести решение всего шестью голосами «за», вместо девяти, необходимых при общем голосовании.

Задача

Вам надо написать программу, которая определяет по заданному разбиению избирателей на группы минимальное количество сторонников партии, достаточное, при некотором распределении их по группам, для принятия любого решения.

Исходные данные

В первой строке записано нечётное число K — количество групп избирателей ($1 \leq K \leq 101$). Во второй строке через пробел записаны K нечётных чисел, которые задают количество избирателей в группах. Население острова не превосходит 9999 человек.

Результат

Выведите минимальное количество сторонников партии, способное решить исход голосования.



Пример

исходные данные	результат
3 5 7 5	6

2) Описание алгоритма:

Сортируем значения количества людей по возрастанию. Берем количество групп на 1 больше половины. Берем количество людей на 1 больше половины. Добавляем к результату полученное значение.

3) Результаты тестирования

LCOV - code coverage report							
Current view: top level - /c/Temp/tdd/tst_example				Hit	Total	Coverage	
Test: coverage.info				Lines:	29	29	100.0 %
Date: 2017-12-18 00:38:48				Functions:	2	2	100.0 %
				Branches:	12	18	66.7 %
Filename	Line Coverage		Functions	Branches			
elections.cpp		100.0 %	9 / 9	100.0 %	1 / 1	100.0 %	4 / 4
main.cpp		100.0 %	20 / 20	100.0 %	1 / 1	57.1 %	8 / 14

LCOV - code coverage report							
Current view: top level - c/Temp/tdd/tst_example - elections.cpp (source / functions)				Hit	Total	Coverage	
Test: coverage.info				Lines:	9	9	100.0 %
Date: 2017-12-18 00:38:48				Functions:	1	1	100.0 %
				Branches:	4	4	100.0 %

Branch data	Line data	Source code
1	:	: #include "elections.h"
2	:	: #include <algorithm>
3	:	:
4	:	:
5	:	5 : int ElectionsNumber(int aCount, int* pData) {
6	[+ +]:	5 : if(aCount <= 0) {
7	:	1 : return -1;
8	:	}
9	:	4 : int result(0);
10	:	4 : int countForResult = (aCount + 1) / 2;
11	[+ +]:	4 : std::sort(pData, pData + aCount);
12	:	60 : for(int i(0); i < countForResult; i++) {
13	:	56 : result += (pData[i] + 1) / 2;
14	:	}
15	:	4 : return result;
	:	}

4) Ссылка на репозиторий с заданием

https://github.com/megagnom37/PracticeHomeWork/tree/master/lab_3_tdd/test

5) Результат прохождения задания на timus

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
7676308	01:47:09 18 дек 2017	ISkripachev	1025. Демократия в опасности	Visual C++ 2017	Accepted		0.001	188 КБ

6) Текст программы

```

main.cpp
#include "elections.h"

#define MY_DEF_USE_LIBTAP
#ifdef MY_DEF_USE_LIBTAP

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

using namespace std;

int main(int, char *[]) {
    //
    plan_tests(5);
    int* arr = new int[1000];
    //
    arr[0] = 5;
    arr[1] = 7;
    arr[2] = 5;
    ok(ElectionsNumber(3, arr) == 6, "count of 3 groups with data: [5,7,5] is
equal 6");
    //
    arr[0] = 3;
    arr[1] = 5;
    arr[2] = 8;
    arr[3] = 9;
    ok(ElectionsNumber(4, arr) == 5, "count of 4 groups with data: [3,5,8,9] is
equal 5");
    //
    ok(ElectionsNumber(0, arr) == -1, "invalid count of groups");
    //
    arr[0] = 3;
    ok(ElectionsNumber(1, arr) == 2, "count of 1 groups with data: [3] is equal
2");
    //
    for(int i(0); i < 101; i++) {
        arr[i] = 99;
    }
    ok(ElectionsNumber(101, arr) == 2550, "count of 101 groups with data: [99,
..., 99] is equal 2550");

    delete[] arr;
    return exit_status(); // ВЫВОД ОТЧЕТА ПО ТЕСТАМ

    return 0;
}

#endif

```



```
elections.cpp
#include "elections.h"
#include <algorithm>

int ElectionsNumber(int aCount, int* pData) {
    if(aCount <= 0) {
        return -1;
    }
    int result(0);
    int countForResult = (aCount + 1) / 2;
    std::sort(pData, pData + aCount);
    for(int i(0); i < countForResult; i++) {
        result += (pData[i] + 1) / 2;
    }
    return result;
}
```

Проектная работа. Приложение-виджет для андроид

1) Условия задания:

Создать приложение-виджет для устройств на платформе Android.

Виджет должен отображать время прибытия автобуса к остановке и оставшееся время до его ухода от остановки. Данная информация должна отображаться для последнего ушедшего, следующего и после следующего автобусов.

Информация о расписании автобуса задается через Google-таблицы.

2) Описание приложения:

Внешний вид виджета можно увидеть на рис. 17.

Виджет визуально делится на несколько составляющих:

- Имя для расписания
- Визуализированная кнопка обновления информации о ближайших автобусах
- Блок из трех цветных строк и двух столбцов.
 - Левый столбец – время прибытия автобуса на остановку
 - Правый столбец – время до/после прибытия/отправления автобуса к/от остановки
 - Строки:
 - Красная – последний ушедший автобус
 - Зеленая – ближайший прибывающий
 - Желтая – следующий после ближайшего прибывающего

Виджет занимает пространство размером 2x2 на рабочем столе устройства.

Настройка виджета происходит при его создании. При создании виджета приложение предложит пользователю выбрать файл в формате CSV в одной из локальных директорий. После выбора пользователем нужного файла, программа автоматически проанализирует файл и сгенерирует виджет относительно введенных данных. Образец корректно заполненных CSV таблиц представлены на рис. 18.

Создание и экспорт таблиц в формате CSV можно делать даже онлайн (например в Google таблицах), что позволяет пользоваться виджетом без установки дополнительного софта.

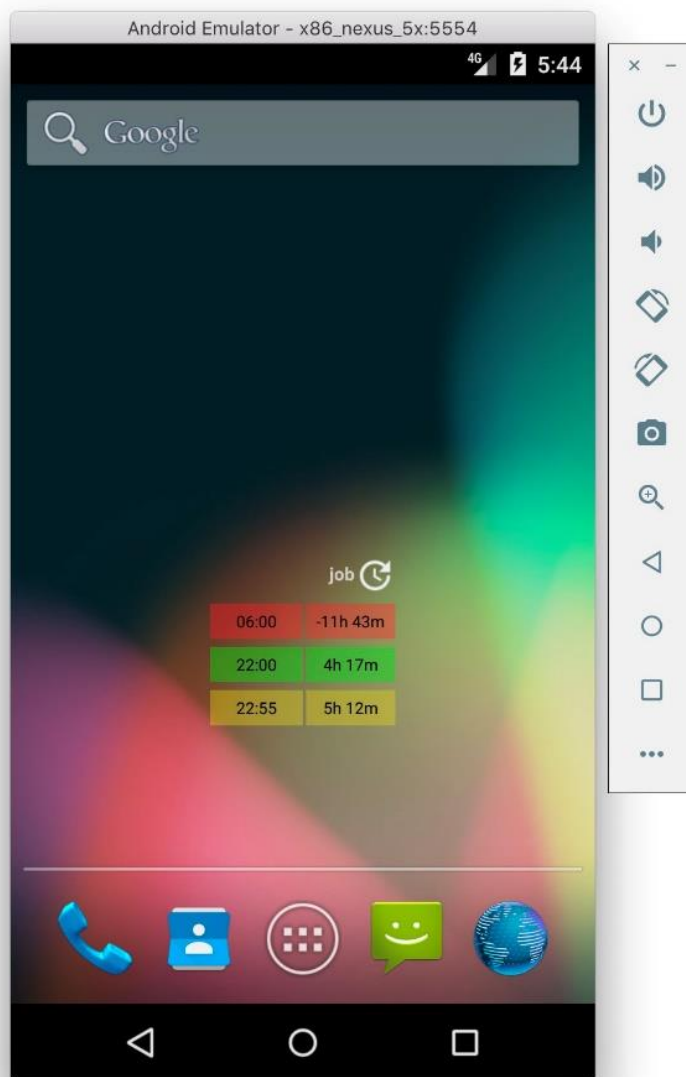
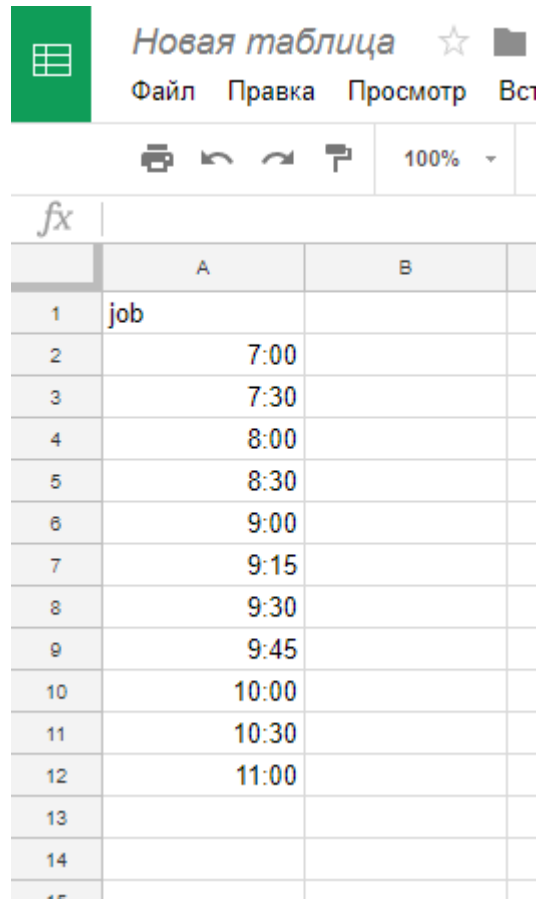


Рисунок 17. Запущенное приложение-виджет на Android-устройстве.



Новая таблица ☆

Файл Правка Просмотр Всп

100%

fx

	A	B
1	job	
2	7:00	
3	7:30	
4	8:00	
5	8:30	
6	9:00	
7	9:15	
8	9:30	
9	9:45	
10	10:00	
11	10:30	
12	11:00	
13		
14		
15		

Рисунок 18. Пример корректно заполненной CSV таблицы в сервисе Google таблицы.

3) Результаты тестирования

Coverage Summary for Class: TimeFunction (com.buswidget.xab.busproject2)

Class	Class, %	Method, %	Line, %
TimeFunction	100% (1/ 1)	100% (5/ 5)	100% (42/ 42)

```

1 package com.buswidget.xab.busproject2;
2 import java.util.ArrayList;
3
4 /**
5  * Created by ivanskripacev on 25.02.18.
6  */
7
8 public class TimeFunction {
9
10     public static final int LAST = 0;
11     public static final int NEXT = 1;
12     public static final int AFTER_NEXT = 2;
13
14     public static final int MIN_IN_24H = 1440;
15
16     public static Integer ConvertTimeToInt(String time){
17         Integer result;
18
19         String[] hoursAndMinutes = time.split(":");
20         if(hoursAndMinutes.length < 2) return -1;
21
22         int hours = Integer.parseInt(hoursAndMinutes[0]) * 60;
23         if(hours >= MIN_IN_24H || hours < 0) return -2;
24
25         int minutes = Integer.parseInt(hoursAndMinutes[1]);
26         if(minutes > 59 || minutes < 0) return -2;
27
28         result = hours + minutes;
29
30         return result;
31     }
32
33     public static String ConvertIntToTime(Integer time){
34         String result = new String("");
35
36         if(Math.abs(time) >= MIN_IN_24H) return "incorrect";
37
38         Integer hours = time / 60;
39         Integer minutes = time % 60;
40
41         if (Math.abs(hours) > 0){
42             minutes = Math.abs(minutes);
43             result += hours.toString() + "h ";
44             result += minutes.toString() + "m";
45         }else{
46             result += minutes.toString() + "m";
47         }
48
49         return result;
50     }

```

```

52     public static Integer CalcDiffOfTime(Integer time, Integer curTime){
53         Integer result;
54
55         if(Math.abs(time) >= MIN_IN_24H) return -1;
56         if(Math.abs(curTime) >= MIN_IN_24H) return -2;
57
58         if (time < curTime){
59             result = (time + MIN_IN_24H) - curTime;
60         }
61         else{
62             result = time - curTime;
63         }
64
65         return result;
66     }
67
68     public static Integer FoundBusTime(ArrayList<Integer> timesArray, Integer curTime, int ty
69         Integer result = -1;
70
71         if(curTime >= MIN_IN_24H || curTime < 0) return -1;
72         if(typeTime > 3 || typeTime < 0) return -2;
73         if(timesArray.isEmpty()) return -3;
74
75         for(int i=(timesArray.size()-1); i >= 0; i--){
76             if(timesArray.get(i) <= curTime){
77                 result = i;
78                 break;
79             }
80         }
81
82         if(result == -1){
83             result = timesArray.size()-1;
84         }
85
86         if(typeTime == NEXT){
87             result++;
88         }
89
90         if(typeTime == AFTER_NEXT){
91             result += 2;
92         }
93
94         if(result > (timesArray.size()-1))
95             result -= (timesArray.size());
96
97         return result;
98     }
99 }

```

4) Ссылка на репозиторий проекта

<https://github.com/megagnom37/PracticeHomeWork/tree/master/Project>

Ссылка на тестовый репозиторий

https://github.com/megagnom37/PracticeHomeWork/tree/master/Project/BusProject2/test_report

**Приложение А. Презентация с семинара по C++11 на тему:
Tuple and tie.**

Ссылка на презентацию:

<https://drive.google.com/drive/folders/0B-W-Xq4KaDMnLWpYVXlpUXB4VGM>