

**федеральное государственное автономное образовательное  
учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)  
(Факультет информационных технологий)**

**(Институт Принтмедиа и информационных технологий)  
Кафедра Информатики и информационных технологий**

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Дисциплина:** Введение в программирование

**Тема:** Основы программирования на С и С++

**Выполнил(а): студент(ка) группы** 231-332

Карпова Елизавета Александровна

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_  
(Дата) (Подпись)

**Проверил:** \_\_\_\_\_  
(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_  
(Дата) (Подпись)

**Замечания:** \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

**Москва 2023**

# Лабораторная работа №1 Основы программирования на «С, С++».

**Цель:** Получить практические навыки в использовании языка С и С++ для создания программ.

## 1. Постановка задачи

1. Разработать алгоритмы и программы, реализующие задания, приводимые ниже.
2. Оформить отчет, содержащий следующие пункты:
  1. Титульный лист.
  2. Название и цель работы.
  3. Постановка задачи.
  4. Блок-схемы алгоритмов и листинги программы с комментариями.

## 2. Задания

1. Используя средства языка С и С++ (printf(), cout) напишите программу, выводящую на экран ваше любимое стихотворение. Для разбиения на строки используйте подходящую управляющую последовательность.

Блок схема алгоритма:



```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");
    cout << "Ночь, улица, фонарь, аптека,\n\
Бессмысленный и тусклый свет.\n\
Живи еще хоть четверть века -\n\
Всё будет так. Исхода нет.\n";
    cout << "\n";
    cout << "Умрёшь - начнёшь опять сначала\n\
И повторится всё, как встарь:\n\
Ночь, ледяная рябь канала,\n\
Аптека, улица, фонарь.\n";
    return 0;
}
```

Консоль отладки Microsoft Visual Studio

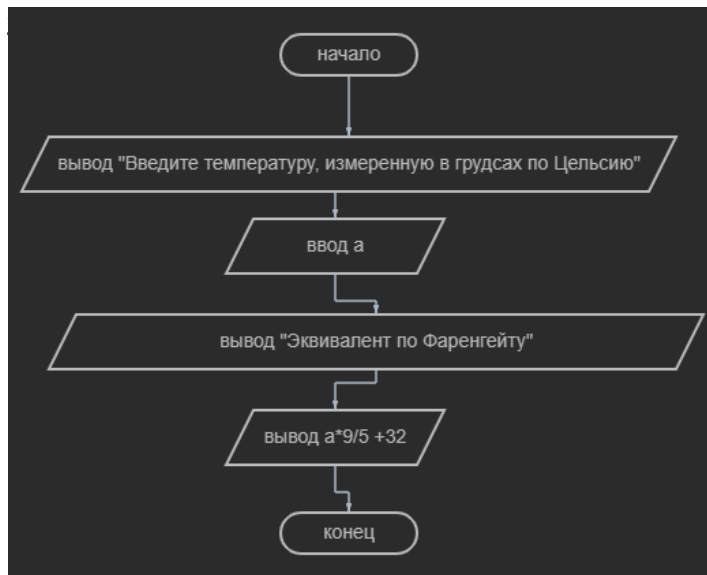
Ночь, улица, фонарь, аптека,  
Бессмысленный и тусклый свет.  
Живи еще хоть четверть века -  
Всё будет так. Исхода нет.

Умрёшь - начнёшь опять сначала  
И повторится всё, как встарь:  
Ночь, ледяная рябь канала,  
Аптека, улица, фонарь.

C:\Users\karp\Desktop\Политех\C++\lab1\g\lab11.exe (процесс 3240) завершил работу.

2. Температуру, измеренную в градусах по Цельсию, можно перевести в градусы по Фаренгейту путем умножения на  $9/5$  и сложения с числом 32. Напишите программу, запрашивающую температуру в градусах по Цельсию и отображающую ее эквивалент по Фаренгейту.

Блок схема алгоритма:



```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(LC_ALL, "Russian");
7      int a;
8      cout << "Введите температуру, измеренную в градусах по Цельсию\n";
9      cin >> a;
10     cout << "Эквивалент по Фаренгейту\n";
11     cout << a * 9 / 5 + 32;
12     return 0;
13 }
```

Консоль отладки Microsoft Visual Studio

```
Введите температуру, измеренную в градусах по Цельсию
100
Эквивалент по Фаренгейту
212
```

3. Две дроби  $a/b$  и  $c/d$  можно сложить следующим образом:  $a/b + c/d = (a*d + b*c)/(b*d)$

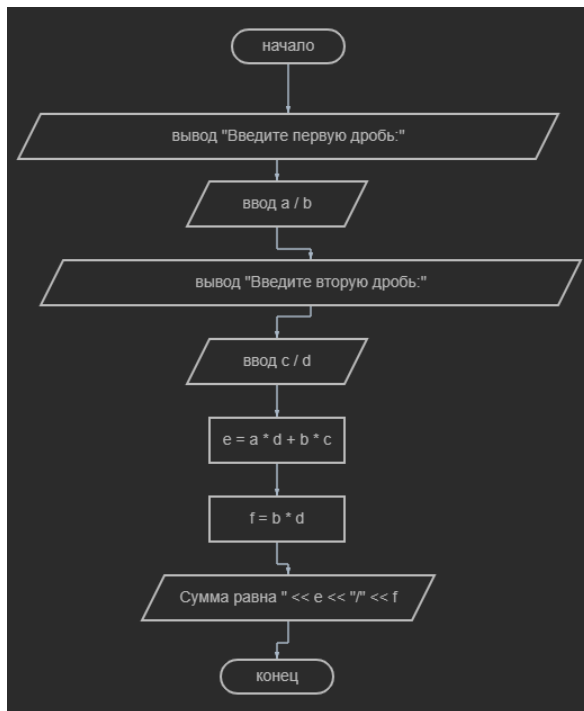
Например,  $1/4 + 2/3 = (1*3 + 4*2)/4*3 = 11/12$

Напишите программу, запрашивающую у пользователя значения двух дробей, а затем выводящую результат, также записанный в форме дроби. Взаимодействие программы с пользователем может выглядеть, например, следующим образом:

Введите первую дробь: 1/2 Введите  
вторую дробь: 2/5 Сумма равна 9/10

Вы можете использовать тот факт, что операция извлечения `>>` может считывать более одного значения за раз: `cin >> a >> dummychar >> b;`

Блок схема алгоритма:



```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(LC_ALL, "Russian");
7      char dummy;
8      int a, b, c, d, e, f;
9      cout << "Введите первую дробь: ";
10     cin >> a >> dummy >> b;
11     cout << "Введите вторую дробь: ";
12     cin >> c >> dummy >> d;
13     e = a * d + b * c;
14     f = b * d;
15     cout << "Сумма равна " << e << "/" << f;
16     return 0;
17 }
```

Консоль отладки Microsoft Visual Studio

Введите первую дробь: 1 / 2  
Введите вторую дробь: 2 / 5  
Сумма равна 9/10

#### 4. Устаревшая денежная система Великобритании состояла из фунтов, шиллингов и пенсов. 1 фунт был равен 20 шиллингам, а 1 шиллинг — 12 пенсам.

Для записи использовалась система, состоящая из знака £ и трех десятичных значений, разделенных точками. Например, запись £5.2.8 обозначала 5 фунтов, 2 шиллинга и 8 пенсов (пенс — множественное число от пенни). Современная денежная система, принятая в 50-е годы XX века, состоит только из фунтов и пенсов, причем один фунт равен 100 пенсам. Такой фунт называют десятичным. Таким образом, в новой денежной системе указанная сумма будет обозначаться как £5.13 (если быть точнее, £5.1333333). Напишите программу, которая будет преобразовывать сумму, записанную в старом формате (фунты, шиллинги, пенсы), в новый формат (фунты, пенсы). Форматом взаимодействия программы с пользователем может являться следующий:

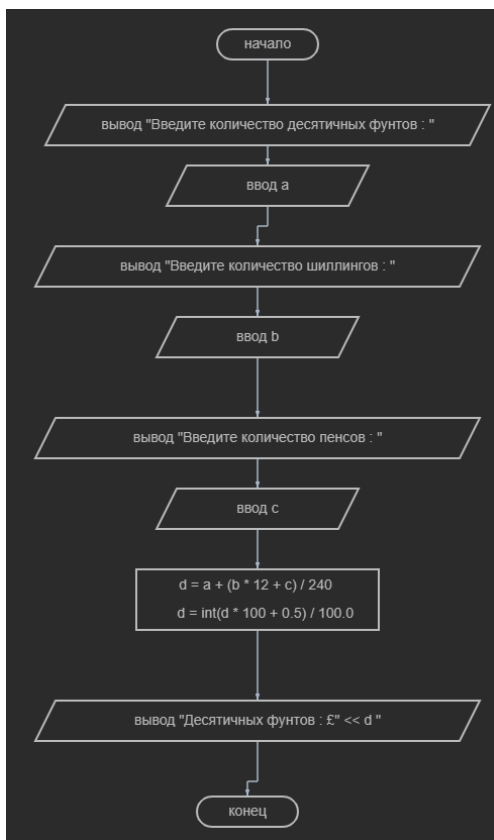
Введите количество фунтов: 7

Введите количество шиллингов: 17

Введите количество пенсов: 9  
Десятичных фунтов: £7.89

В большинстве компиляторов для представления знака £ используется десятичный код 156. Некоторые компиляторы позволяют скопировать знак фунта прямо из таблицы символов Windows.

Блок-схема алгоритма:



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     setlocale(LC_ALL, "Russian");
7     int a;
8     float b, c, d;
9     cout << "Введите количество фунтов : ";
10    cin >> a;
11    cout << "Введите количество шиллингов : ";
12    cin >> b;
13    cout << "Введите количество пенсов : ";
14    cin >> c;
15    d = a + (b * 12 + c) / 240; //формула перевода в новый формат
16    d = int(d * 100 + 0.5) / 100.0;
17    cout << "Десятичных фунтов : £" << d;
18    return 0;
19 }
```

Ln: 19, Col: 2



Run



Share

Command Line Arguments



Введите количество фунтов :

7



Введите количество шиллингов :

17



Введите количество пенсов :

9

Десятичных фунтов : £7.89

5. Напишите программу, выполняющую действия, обратные тем, которые описаны в предыдущем упражнении 4, то есть запрашивающую у пользователя сумму, указанную в десятичных фунтах, и переводящую ее в старую систему

фунтов, шиллингов и пенсов. Пример взаимодействия программы с пользователем может выглядеть так:

Введите число десятичных фунтов: 3.51

Эквивалентная сумма в старой форме записи: £3.10.2

Обратите внимание на то, что если вам придется присваивать вещественное значение (например, 12.34) переменной целого типа, то его дробная часть (0,34) будет потеряна, а целая переменная получит значение 12. Чтобы избежать предупреждения со стороны компилятора, используйте явное преобразование типов. Можно использовать операторы, подобные приведенным ниже:

```
float decpounds: // сумма в десятичных фунтах
```

```
float decfrac: // десятичная дробная часть
```

```
int pounds: // сумма в старых фунтах
```

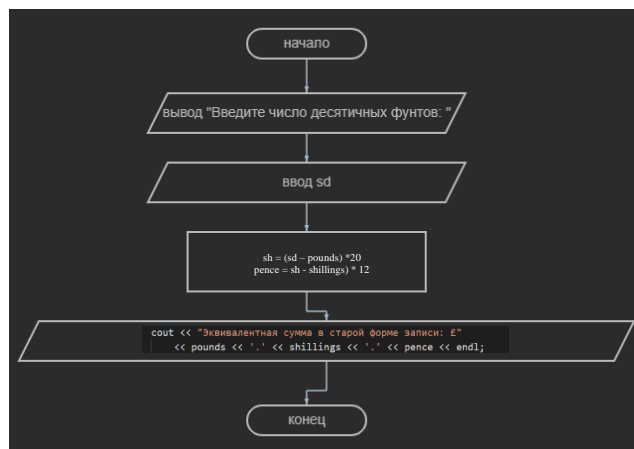
```
pounds = static_cast<int>(decpounds); //отбрасывание
```

```
//дробной части
```

```
decfrac = decpounds - pounds; //выделение дробной части
```

Чтобы получить число шиллингов, следует умножить на 20 значение переменной decfrac. Аналогичным образом можно получить число пенсов.

Блок схема алгоритма:



```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    float sd; // сумма в десятичных фунтах
    float sh; // десятичная дробная часть
    int pounds; // сумма в старых фунтах
    cout << "Введите число десятичных фунтов: ";
    cin >> sd;
    pounds = static_cast<int>(sd); // отбрасывание дробной части
    sh = (sd - pounds) * 20; // выделение дробной части и расчет шиллингов
    int shillings = static_cast<int>(sh); // перевод в тип int
    int pence = static_cast<int>((sh - shillings) * 12); // расчет пенсов
    cout << "Эквивалентная сумма в старой форме записи: £"
        << pounds << "." << shillings << "." << pence << endl;
}
```