

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

**ОТЧЁТ ПО  
ВЫПОЛНЕННЫМ ЗАДАЧАМ НА C++**  
Дисциплина: «Основы алгоритмизации и программирования»

Выполнил:  
Студент группы ИВТ-21-26  
Безух Владимир Сергеевич

Проверил:  
Доцент кафедры ИТАС  
Полякова Ольга Андреевна

Пермь,  
Ноябрь 2021

## Разадресация и разыменование

Когда мы говорим о разадресации, то имеем в виду, что хотим получить вместо возвращаемого значения его непосредственный адрес. Разыменование противоположная операция, позволяющая вернуть находящееся по адресу значение.

```
int a = 10;

/*
 * Разыменует адрес переменной и вернёт значение, которое по нему находится.
 * Компилятор знает о переменной именно её адрес, но за счёт абстракции нам не обязательно
 * задумываться об этом.
 */
cout << a << endl;

// Разадресуем переменную и попросим вернуть непосредственно адрес, где хранится значение.
cout << &a << endl;

// Сохраним в переменной b типа указатель непосредственный адрес переменной a.
int *b = &a;

// Как и обычная переменная, указатель вернёт своё значение, – в случае указателя это адрес.
cout << b << endl;

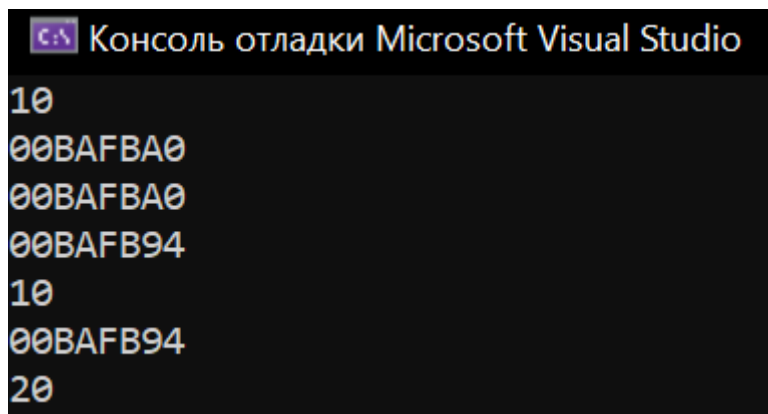
// Ничто не мешает нам вернуть адрес памяти, где хранится значение переменной-указателя.
cout << &b << endl;

// Или мы можем разыменовывать значение-адрес и вернуть значение, которое по нему хранится.
cout << *b << endl;

// Указатели могут указывать на другие указатели сколько угодно раз.
int **c = &b;
cout << c << endl;

// Но через указатель с нельзя напрямую изменить значение переменной a.
// Зато можно через одномерный указатель.
*b = 20;

cout << a << endl;
```



```
Консоль отладки Microsoft Visual Studio
10
00BAFBA0
00BAFBA0
00BAFB94
10
00BAFB94
20
```

## Небольшая задача

Чему будет равно...

```
#include <iostream>

using namespace std;

int main()
{
    int i = 1;

    cout << ++i * i++;

}
```

На первый взгляд может показаться, что в соответствии с приоритетами выполнения операций, получается:

- 1)  $++i * 1$  — ( $++i$  сначала вернул значение, а затем увеличил его на единицу)
- 2)  $3 * 1$  — ( $++i$  сначала увеличил значение, а потом вернул)
- 3) Ответ 3

Однако запустив код в разных средах можно получить такие ответы, как 4 ( $2 * 2$ ) и 6 ( $3 * 2$ ). Эти ответы напрямую зависят от настроек компилятора и уровней его оптимизации. Поэтому по правилам языка C++ такое поведение считается неопределённым (**undefined behavior**).

Дело в том, что между предыдущей и следующей точками следования, скалярный объект должен иметь своё сохранённое значение, изменённое **не более одного раза** при вычислении выражения, в противном случае поведение не определено.

Нужно всегда избегать неопределённого поведения!

## Постановка задач

**Первое задание:**

Считая, что кубический фут равен 7.481 галлона, напишите программу, запрашивающую у пользователя число галлонов и выводящую на экран эквивалентный объем в кубических футах.

**Второе задание:**

Напишите программу, генерирующую вывод:

10

20

19

Используйте представление числа 10 в виде целой константы. Для вывода 20 используйте присваивание, для вывода 19 операцию декремента.

**Третье задание:**

Напишите программу для умножения двух чисел и вывода на экран результата. Числа задаёт пользователь.

**Четвёртое задание:**

Вывести последовательность символов в обратном порядке.

**Пятое задание:**

Написать программу, которая переводит температуру в градусах по Фаренгейту в градусы Цельсия по формуле:  $C = 5/9 \cdot (F - 32)$

**Шестое задание:**

Заданы два момента времени в течение дня. Написать программу для нахождения промежутка времени между ними.

**Анализ задач**

- 1) Достаточно воспользоваться формулой конвертации.
- 2) Выводим числа, как указано в задаче.
- 3) Находим произведение двух чисел.
- 4) Разворачиваем последовательность символов любым удобным способом.
- 5) Достаточно воспользоваться формулой конвертации.
- 6) Для сравнения моментов времени переводим каждый из них в общее количество секунд, прошедших с начала дня. Затем модуль разницы в секундах преобразуем обратно.

В коде к задачам нет проверки вводимых данных, т.е. изначально предполагается, что абсолютно все вводимые данные корректны и совместимы.

## **Описание переменных**

1)

**const float GALLONS** = getGallons(); — количество галлонов.

**const float CUBIC\_FEET** = gallonsToCubicFeetConversion(GALLONS); — количество кубических футов.

2)

**int number; const int TEN;** — необходимые переменные.

3)

**float FIRST\_NUMBER, SECOND\_NUMBER;** — множители произведения.

4)

**const string INPUT\_STRING** = getString(); — строка для обратного вывода.

5)

**const float FAHRENHEIT** = getFahrenheit(); — количество градусов Фаренгейта.

**const float CELSIUS** = fahrenheitToCelsiusConversion(FAHRENHEIT); — количество градусов Цельсия.

6)

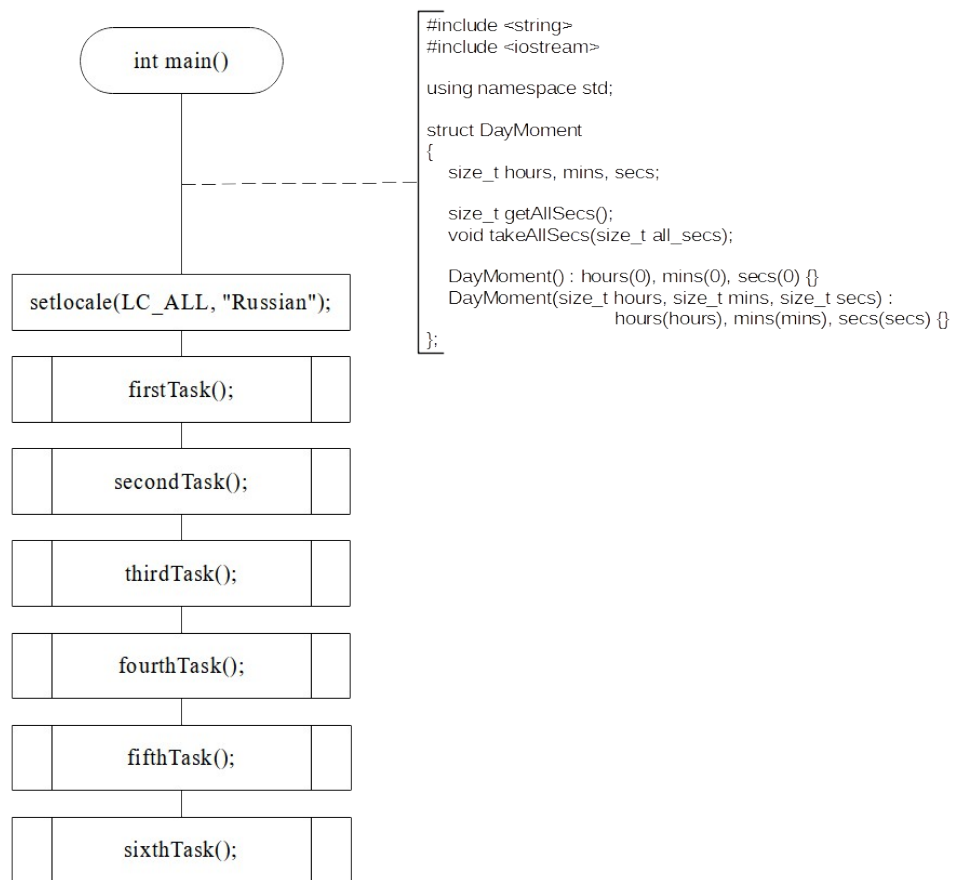
**struct DayMoment** { size\_t hours, mins, secs; ... } — структура момента времени суток.

**DayMoment FIRST\_DAY\_MOMENT** = getDayMoment(),

**SECOND\_DAY\_MOMENT** = getDayMoment(); — объекты структуры для заданных моментов времени суток.

**DayMoment DIFF\_TIME;** — разница во времени между двумя моментами времени суток.

## Блок-схема



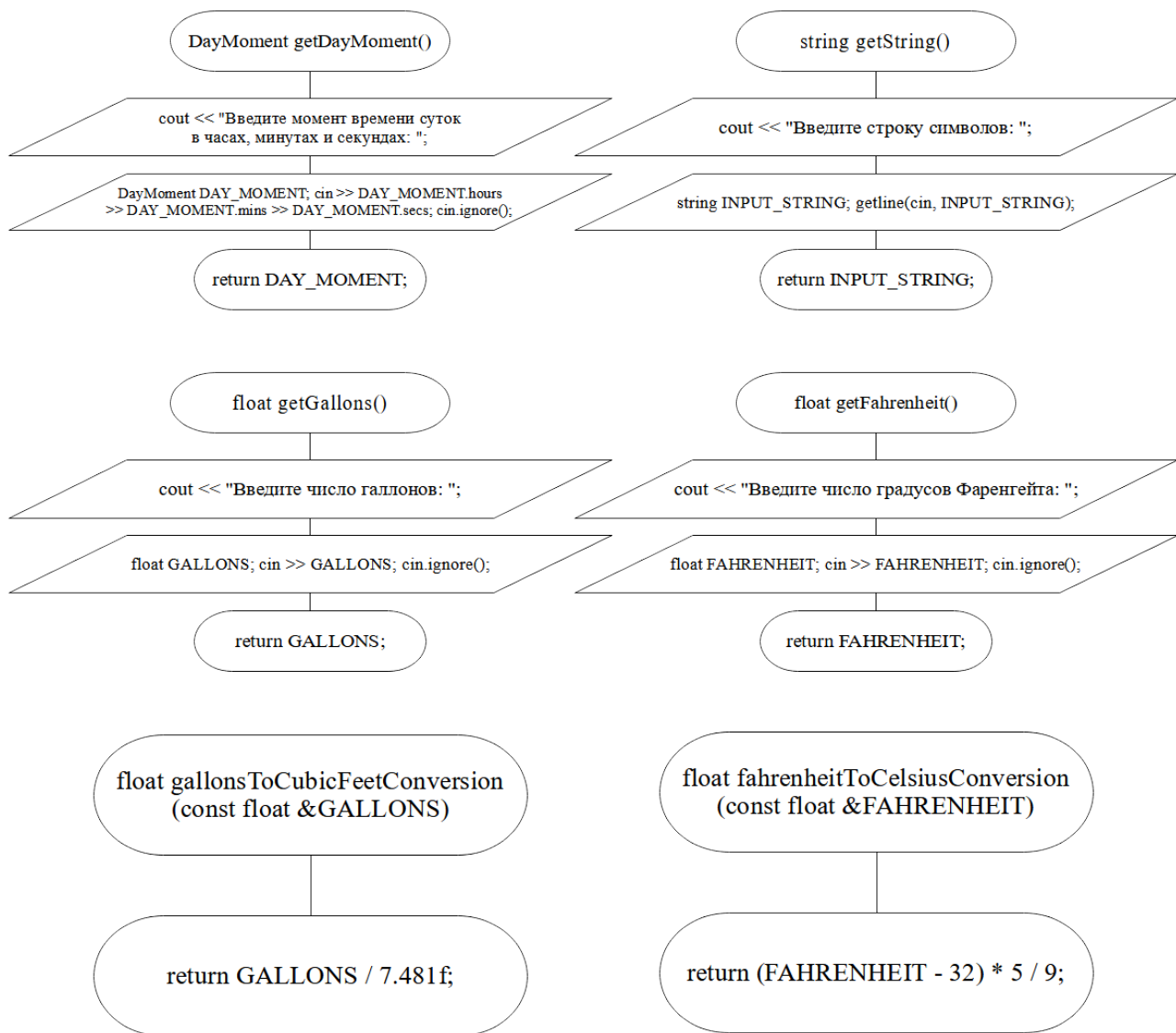
`size_t DayMoment::getAllSecs()`

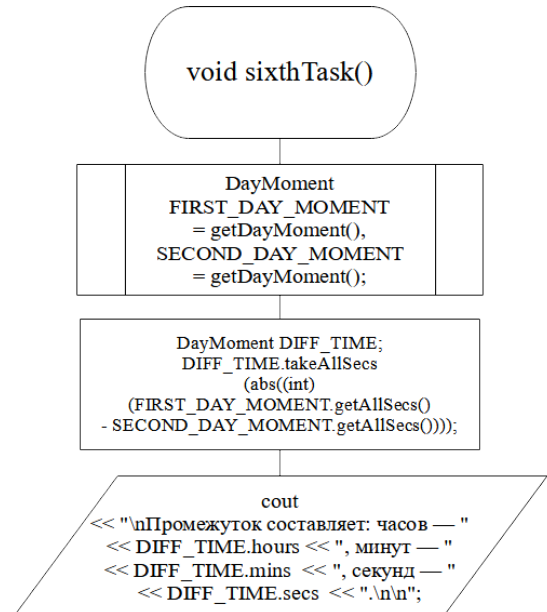
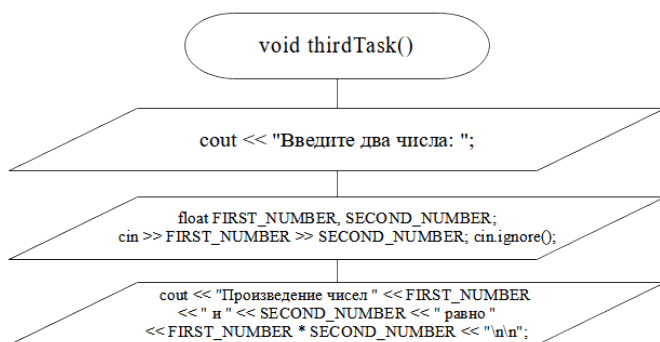
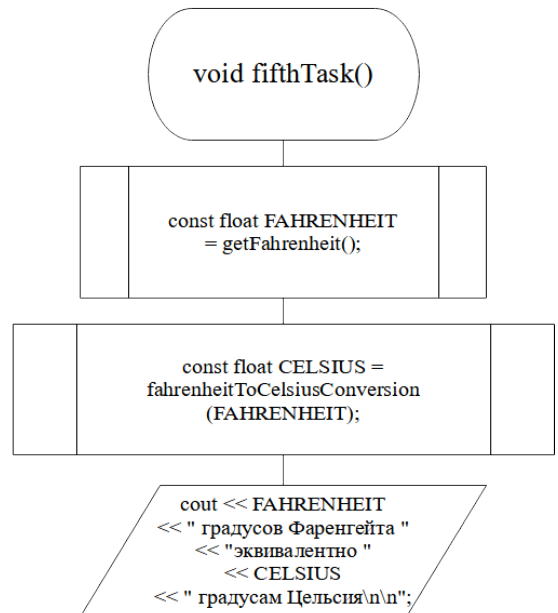
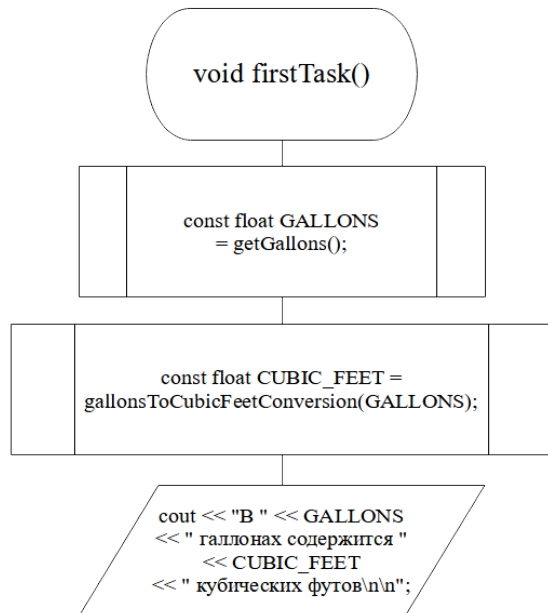
`return  
hours * 3600 + mins * 60 + secs;`

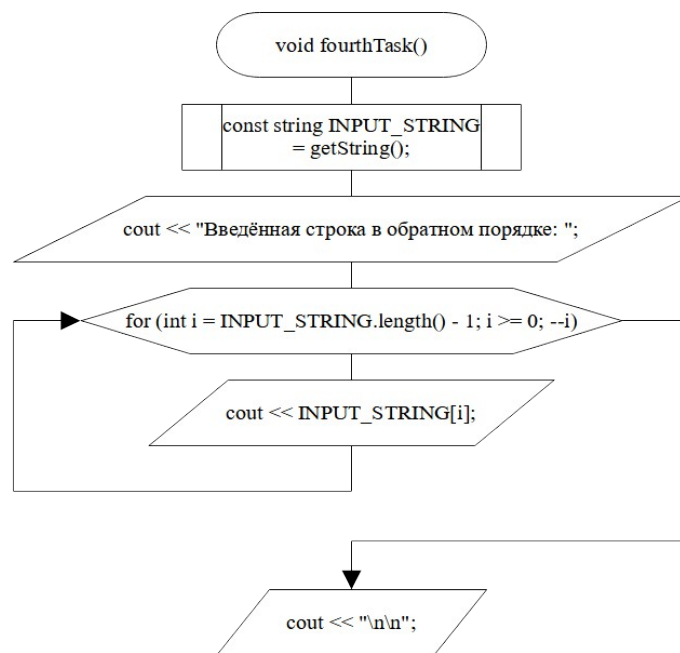
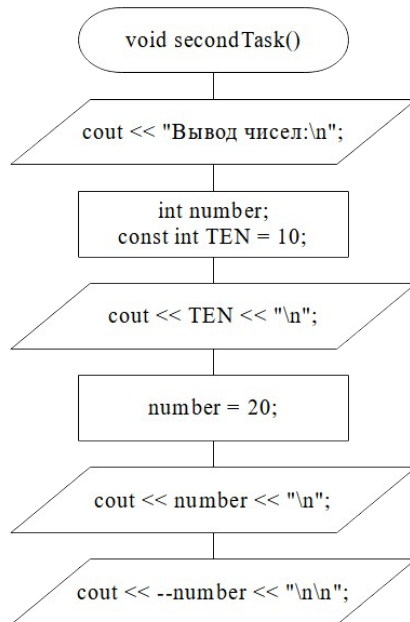
`void DayMoment::takeAllSecs  
(size_t all_secs)`

```
hours = all_secs / 3600; all_secs %= 3600;  
mins = all_secs / 60; all_secs %= 60;  
secs = all_secs;
```









## Исходный код

```
/*
 * This code is licensed under the Creative Commons
 * Attribution - NonCommercial - NoDerivatives 4.0 International License.
 * To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/
 * or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
 *
 *
 * https://bezukh.wixsite.com/blog
 * https://github.com/BezukhVladimir
 *
 * © Developed by Bezukh Vladimir. All right reserved.
 */

/*
 * Developed by Bezukh Vladimir
 * November 2021
 * ИБТ-21-26
 *
 * Описание заданий.
 */

#include <string>
#include <iostream>

using namespace std;

struct DayMoment
{
    size_t hours, mins, secs;

    size_t getAllSecs();
    void takeAllSecs(size_t all_secs);

    DayMoment() : hours(0), mins(0), secs(0) {}
    DayMoment(size_t hours, size_t mins, size_t secs) :
        hours(hours), mins(mins), secs(secs) {}
};

size_t DayMoment::getAllSecs()
{
    return hours * 3600 + mins * 60 + secs;
}

void DayMoment::takeAllSecs(size_t all_secs)
{
    hours = all_secs / 3600; all_secs %= 3600;
    mins = all_secs / 60;    all_secs %= 60;
    secs = all_secs;
}

DayMoment getDayMoment()
{
    cout << "Введите момент времени суток в часах, минутах и секундах: ";

    DayMoment DAY_MOMENT;
    cin >> DAY_MOMENT.hours >> DAY_MOMENT.mins >> DAY_MOMENT.secs; cin.ignore();

    return DAY_MOMENT;
}
```

```

string getString()
{
    cout << "Введите строку символов: ";

    string INPUT_STRING; getline(cin, INPUT_STRING);

    return INPUT_STRING;
}

float getGallons()
{
    cout << "Введите число галлонов: ";

    float GALLONS; cin >> GALLONS; cin.ignore();

    return GALLONS;
}

float getFahrenheit()
{
    cout << "Введите число градусов Фаренгейта: ";

    float FAHRENHEIT; cin >> FAHRENHEIT; cin.ignore();

    return FAHRENHEIT;
}

float gallonsToCubicFeetConversion(const float &GALLONS)
{
    return GALLONS / 7.481f;
}

float fahrenheitToCelsiusConversion(const float &FAHRENHEIT)
{
    return (FAHRENHEIT - 32) * 5 / 9;
}

void firstTask()
{
    // Перевод галлонов в кубические футы

    const float GALLONS = getGallons();

    const float CUBIC_FEET =
        gallonsToCubicFeetConversion(GALLONS);

    cout << "В " << GALLONS << " галлонах содержится "
         << CUBIC_FEET << " кубических футов\n\n";
}

void secondTask()
{
    /*

```

```

* Вывод в консоль целой константы со значением 10.
* Вывод в консоль 20, вывод в консоль 19 через операцию декремента.
*/

cout << "Вывод чисел:\n";

int number;
const int TEN = 10;

cout << TEN << "\n";

number = 20;

cout << number << "\n";

cout << --number << "\n\n";
}

void thirdTask()
{
    // Произведение двух чисел

    cout << "Введите два числа: ";

    float FIRST_NUMBER, SECOND_NUMBER;
    cin >> FIRST_NUMBER >> SECOND_NUMBER; cin.ignore();

    cout << "Произведение чисел " << FIRST_NUMBER << " и " << SECOND_NUMBER
        << " равно " << FIRST_NUMBER * SECOND_NUMBER << "\n\n";
}

void fourthTask()
{
    // Обратный вывод последовательности символов

    const string INPUT_STRING = getString();

    cout << "Введённая строка в обратном порядке: ";

    for (int i = INPUT_STRING.length() - 1; i >= 0; --i)
        cout << INPUT_STRING[i];

    cout << "\n\n";
}

void fifthTask()
{
    // Перевод градусов Фаренгейта в градусы Цельсия

    const float FAHRENHEIT = getFahrenheit();

    const float CELSIUS =
        fahrenheitToCelsiusConversion(FAHRENHEIT);

    cout << FAHRENHEIT << " градусов Фаренгейта эквивалентно "
        << CELSIUS << " градусам Цельсия\n\n";
}

void sixthTask()
{
    // Разница между двумя моментами времени суток

```

```

DayMoment FIRST_DAY_MOMENT = getDayMoment(),
          SECOND_DAY_MOMENT = getDayMoment();

DayMoment DIFF_TIME;
DIFF_TIME.takeAllSecs(
    abs((int) (FIRST_DAY_MOMENT.getAllSecs() - SECOND_DAY_MOMENT.getAllSecs())));

cout << "\nПромежуток составляет: часов - "
    << DIFF_TIME.hours << ", минут - "
    << DIFF_TIME.mins << ", секунд - "
    << DIFF_TIME.secs << ".\n\n";
}

int main()
{
    setlocale(LC_ALL, "Russian");

    firstTask();
    secondTask();
    thirdTask();
    fourthTask();
    fifthTask();
    sixthTask();
}

```

**Скриншот консольного интерфейса программы**

## Консоль отладки Microsoft Visual Studio

Введите число галлонов: 4

В 4 галлонах содержится 0.534688 кубических футов

Вывод чисел:

10

20

19

Введите два числа: 21 42

Произведение чисел 21 и 42 равно 882

Введите строку символов: 700 tnegs

Введённая строка в обратном порядке: agent 007

Введите число градусов Фаренгейта: 451

451 градусов Фаренгейта эквивалентно 232.778 градусам Цельсия

Введите момент времени суток в часах, минутах и секундах: 04 20 17

Введите момент времени суток в часах, минутах и секундах: 04 19 38

Промежуток составляет: часов - 0, минут - 0, секунд - 39.

## Анализ результатов

1) Корректная конвертация галлонов в кубические футы.



- 2) Корректный вывод.
- 3) Корректное произведение двух чисел.
- 4) Корректный вывод в обратной последовательности произвольной строки символов.
- 5) Корректная конвертация градусов Фаренгейта в градусы Цельсия.
- 6) Корректный подсчёт промежутка между двумя моментами времени одних суток.

В коде к задачам отсутствует модуль проверки вводимых данных, т.е. изначально предполагается, что абсолютно все вводимые данные корректны и совместимы. Работоспособность кода гарантирована только в этом случае.