

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів  
Кафедра систем управління літальних апаратів

**Лабораторна робота № 9**  
з дисципліни «Алгоритмізація та програмування»  
на тему "Робота з рядками на C ++"

Виконав студент 312 гр.

Кулагін Олексій Костянтинович  
(підпис, дата) (П.І.Б.)

Перевірив  
\_\_\_\_\_ К.Т.Н.,  
доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

**МЕТА РОБОТИ**

Вивчити теоретичний матеріал з основ роботи з низькорівневими рядками на C++ і документацію до класу string, а також алгоритми пошуку в рядку, а також реалізувати обробку рядків на C++ в середовищі Visual Studio.

**ВИКОНАННЯ РОБОТИ**

1) Param 75.

Вхідні данні:

- 1) Введення даних
- 2) Обробка даних (зменшення часу на одну годину)
- 3) Виведення результату

Алгоритм вирішення:

- 1) Ініціалізація часових даних
- 2) Зменшення часу на одну годину
- 3) Виконання головної програми
- 4) Виведення результату

2) Begin 15.

Вхідні данні:

- 1) Введення даних
- 2) Введення площі круга  $S$
- 3) Виведення результату

Алгоритм вирішення:

- 5) Ввести площу круга від користувача
- 6) Обчислити діаметр круга
- 7) Обчислити довжину окружності
- 8) Вивести діаметр і довжину окружності на екран

3) Boolean 17.

Вхідні данні:

- 1) Введення даних
- 2) Введення цілого позитивного числа
- 3) Виведення результату

Алгоритм вирішення:

- 1) Звіряємо введене число, щоб переконатися, що воно трьохзначне і непарне
- 2) Якщо обидва критерії виконуються, повідомляємо, що число є непарним тризначним
- 3) В іншому випадку повідомляємо, що число не підходить

## Лістинг коду вирішення задачі Param 75

```
#include <iostream>
#include <vector>
#include <Windows.h>

using namespace std;

// Определение типа TTime
struct TTime {
    int hours;
    int minutes;
};

// Функция для проверки корректности времени
bool isValidTime(const TTime& t) {
    return (t.hours >= 0 && t.hours < 24 && t.minutes >= 0 && t.minutes < 60);
}

// Процедура, которая уменьшает время на один час
void PrevHour(TTime& t) {
    if (isValidTime(t)) {
        t.hours--;
        if (t.hours < 0) {
            t.hours = 23;
        }
    }
}

// Функция для вывода времени
void printTime(const TTime& t) {
    cout << (t.hours < 10 ? "0" : "") << t.hours << ":"
         << (t.minutes < 10 ? "0" : "") << t.minutes << endl;
}

int main() {

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    // Пять заданных моментов времени
    vector<TTime> times = {
        {10, 30},
        {0, 45},
```

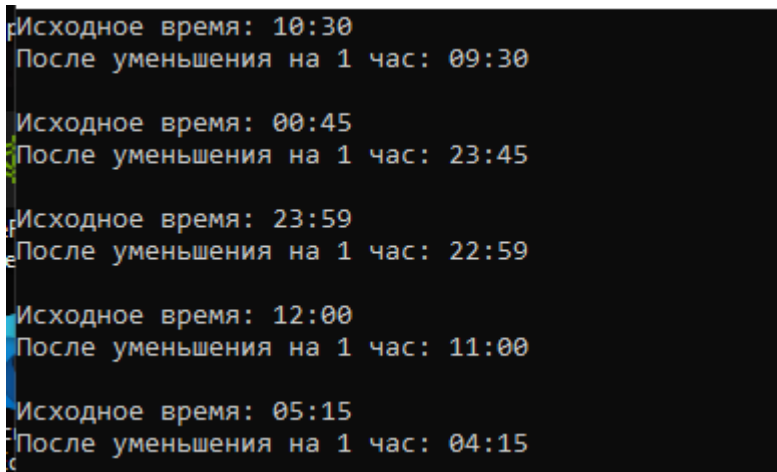
```

        {23, 59},
        {12, 0},
        {5, 15}
    };

    // Применение процедуры PrevHour к каждому времени
    for (TTime& time : times) {
        cout << "Исходное время: ";
        printTime(time);
        PrevHour(time);
        cout << "После уменьшения на 1 час: ";
        printTime(time);
        cout << endl;
    }

    return 0;
}

```



```

Исходное время: 10:30
После уменьшения на 1 час: 09:30

Исходное время: 00:45
После уменьшения на 1 час: 23:45

Исходное время: 23:59
После уменьшения на 1 час: 22:59

Исходное время: 12:00
После уменьшения на 1 час: 11:00

Исходное время: 05:15
После уменьшения на 1 час: 04:15

```

Лістинг коду вирішення задачі Begin 15.

```

#include <iostream>
#include <cmath>
#include <Windows.h>

using namespace std;

```

```

const double PI = 3.14;

// Функция для вычисления диаметра по площади
double calculateDiameter(double S) {
    return sqrt((4 * S) / PI);
}

// Функция для вычисления длины окружности по диаметру
double calculateCircumference(double D) {
    return PI * D;
}

int main() {

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

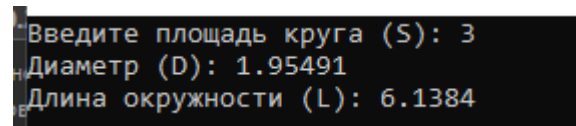
    double S;
    cout << "Введите площадь круга (S): ";
    cin >> S;

    double D = calculateDiameter(S);
    double L = calculateCircumference(D);

    cout << "Диаметр (D): " << D << endl;
    cout << "Длина окружности (L): " << L << endl;

    return 0;
}

```



```

Введите площадь круга (S): 3
Диаметр (D): 1.95491
Длина окружности (L): 6.1384

```

Лістинг коду вирішення задачі Boolean 17.

```

#include <iostream>
#include <Windows.h>

```

```

using namespace std;

bool isOddThreeDigitNumber(int number) {
    // Проверка, что число трехзначное и нечетное
    return (number >= 100 && number <= 999) && (number % 2 != 0);
}

int main() {

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    int number;
    cout << "Введите целое положительное число: ";
    cin >> number;

    if (isOddThreeDigitNumber(number)) {
        cout << "Число является нечетным тризначным." << endl;
    }
    else {
        cout << "Число НЕ является нечетным тризначным." << endl;
    }

    return 0;
}

```

```

Введите целое положительное число: 232
Число НЕ является нечетным тризначным.

```

```

Введите целое положительное число: 233
Число является нечетным тризначным.

```