

## Лабораторная работа №4

### std::string

Представлять строки в виде `char*` так же неудобно, как и представлять массивы в виде `int*`. В стандартной библиотеке C++ есть специальный класс для представления строк: `std::string`. Он умеет хранить длину строки и содержит ряд готовых функций для работы со строками. Чтобы иметь возможность использовать строки в C++, сначала нужно подключить заголовочный файл `<string>`.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str = "Hello!";
    cout<<str[1]<<endl; //Выведет 'e'
    cout<<str.length()<<endl; //Выведет 6
    return 0;
}
```

Как и с обычными переменными, вы можете инициализировать или присваивать им значения:

```
std::string name("Sasha");
name = "Masha"; // присваиваем переменной myName строковый литерал "Masha"
```

**Вывод строк.** Выводить строку можно с помощью `cout`.

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

using namespace std;

int main()
{
    std::string name("Sasha");
    name = "Masha";
    cout<<name;
    return 0;
}
```

**Ввод строк.** А вот с `std::cin` уже немного по-другому. С помощью `cin` можно вводить строки только по одному слову, так как `cin` автоматически разбивает входную последовательность символов по пробелам и переносам строк. Если нужно ввести строку, содержащую пробелы, можно воспользоваться функцией `getline`, которая разбивает строки только по переносу символов.

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

using namespace std;

int main()
{
    cout << "Enter your full name: ";
    string myName;
    getline(cin, myName); // считываем целую строку текста в переменную myName

    cout << "Enter your age: ";
    string myAge;
    getline(cin, myAge); // считываем целую строку текста в переменную myAge

    cout << "Your name is " << myName << " and your age is " << myAge;
}
```

**Задача 1.** Считать две строки и вывести их конкатенацию (сумму строк).

**Задача 2.** Сохранить слова в связный список и вывести их в обратном порядке. Для простоты можно воспользоваться вводом через `cin`. Можно считать, что вводятся будут только символы английского алфавита, разделённые пробелами.

**Задача 3.** Напишите программу, которая подсчитает количество английских букв в тексте. Проверку того, что символ является английской буквой вынесите в отдельную функцию.

**Задача 4.** Напишите шифровщик и дешифратор текстов, использующий шифр Цезаря. Программа должна шифровать только английские буквы. Все остальные символы остаются в тексте как есть.

Шифр Цезаря — это вид шифра, в котором каждая буква смещается на  $k$  позиций по алфавиту. Например, в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее.

Пример шифровки со сдвигом 2:

abc zuzu → cde bwbw

**Задача 5.** Напишите генератор паролей из букв латинского алфавита и цифр. Генератор должен получать на вход длину пароля и выводить в консоль новый пароль. Сделайте отдельную функцию «randomChar» для генерации одного символа из пароля. Эта функция должна возвращать случайный символ из строки допустимых символов. Строка допустимых символов должна включать строчные и прописные английские буквы, а также цифры.

Ниже приведён пример функции random(a, b), которая генерирует случайные числа в диапазоне от a до b включительно.

```
#include <iostream>
#include <cmath>
#include <ctime>

using namespace std;

//Случайное число от a до b включительно
int random(int a, int b)
{
    return a+rand()%(b-a+1);
}

int main()
{
    srand(time(0));
    for (int i=0; i<100; i++)
    {
        cout<<random(1, 10)<<endl;
    }
    return 0;
}
```