

Лабораторная работа №6

Тема: «Строковые данные»

Цель работы: сформировать навыки и умения обработки структурированных типов данных, организованных в виде строковых данных.

Время выполнения: 4 часа.

Теоретические сведения

Мы можем работать со строками в C++ в так называемом C-стиле как с массивами символов, которые оканчиваются на нулевой байт '0'. Однако, что если такой символ не будет найден или в процессе манипуляций со строкой будет удален, то дальнейшие действия с такой строкой могут иметь недетерминированный результат. По этой причине строки в C-стиле считаются небезопасными, и рекомендуется для хранения строк в C++ использовать тип `std::string` из модуля `<string>`.

Объект типа `string` содержит последовательность символов типа `char`, которая может быть пустой. Например, определение пустой строки:

```
std::string message;
```

Также можно инициализировать или присвоить переменной `string` конкретную строку:

```
std::string message {"Hello METANIT.COM!"};
```

```
// или так
```

```
std::string message2 = "Hello METANIT.COM!";
```

```
std::string message3("Hello METANIT.COM!");
```

В данном случае переменная `message` получит копию строкового литерала `"Hello METANIT.COM!"`. В своем внутреннем представлении переменная `message` будет хранить массив символов, который также заканчивается на нулевой байт. Однако реализация типа `string` и предлагаемые им возможности делают работу с этим типом более безопасной.

И можно инициализировать объект `string` другим объектом `string`:

```
std::string hello{"hello world"};
```

```
std::string message {hello}; // message = "hello world"
```

```
// или так
```

```
// std::string message (hello);
```

```
// std::string message = hello;
```

Мы можем вывести подобную строку на консоль:

```
#include <iostream>
```

```
#include <string>
```

```

int main()
{
    std::string message {"Hello METANIT.COM!"};
    std::cout << "Message: " << message << std::endl; // Message: Hello
METANIT.COM!
}

```

Получение и изменение символов строки

Подобно массиву мы можем обращаться с помощью индексов к отдельным символам строки, получать и изменять их:

```

std::string hello {"Hello"};
char c {hello[1]};    // e
hello[0]='M';
std::cout << hello << std::endl;    // Mello

```

Поскольку объект string представляет последовательность символов, то эту последовательность можно перебрать с помощью цикла for. Например, подсчитаем, сколько раз в строке встречается буква "l":

```

#include <iostream>
#include <string>

int main()
{
    unsigned count{}; // счетчик, сколько раз встречается символ
    std::string message{ "Hello World"};
    for(const char c: message)
    {
        if(c == 'l')
        {
            count++;
        }
    }
    std::cout << "Count: " << count << std::endl; // Count: 3
}

```

Чтение строки с консоли

Для считывания введенной строки с консоли, как и для считывания других значений, можно использовать объект std::cin:

```

#include <iostream>
#include <string>

```

```
int main()
{
    std::string name;
    std::cout << "Input your name: ";
    std::cin >> name;
    std::cout << "Your name: " << name << std::endl;
}
```

Консольный вывод:

Input your name: Tom

Your name: Tom

Однако если при данном способе ввода строка будет содержать подстроки, разделенные пробелом, то `std::cin` будет использовать только первую подстроку:

Input your name: Tom Smith

Your name: Tom

Чтобы считать всю строку, применяется метод `getline()`:

```
#include <iostream>
```

```
#include <string>
```

```
int main()
{
    std::string name;
    std::cout << "Input your name: ";
    getline(std::cin, name);
    std::cout << "Your name: " << name << std::endl;
}
```

Метод `getline` принимает два объекта - `std::cin` и переменную, в которую надо считать строку.

Консольный вывод:

Input your name: Tom Smith

Your name: Tom Smith

Индивидуальные задания к лабораторной работе №6

Строки

1. Ввести строку. Удалить все повторяющиеся символы и вывести результат на экран.
2. Ввести строку. Вывести на экран самое длинное слово в строке.
3. Ввести строку. Посчитать количество слов в строке и результат вывести на экран.
4. Ввести две строки. Сравнить строки и в самой короткой удалить все слова, которые начинаются на строчную букву. Вывести полученную строку на экран.
5. Ввести строку. Посчитать количество строчных и прописных букв в строке и вывести результат на экран.
6. Ввести две строки. Сравнить строки и, в самой длинной строке отсортировать ее элементы по убыванию (допускается использование только цифровых строк).
7. Ввести две строки. Сравнить строки и в самой короткой удалить все слова, которые начинаются на прописную букву. Вывести полученную строку на экран.
8. Ввести строку. Удалить из строки все лишние пробелы. Результат вывести на экран.
9. Ввести строку. Посчитать количество гласных букв русского алфавита в строке. Результат вывести на экран.
10. Ввести строку. Из строки удалить все символы, не являющиеся цифрами. Вывести получившуюся строку на экран.
11. Ввести две строки. Сравнить строки и вывести самую короткую на экран.
12. Ввести строку. Посчитать количество символов в ней и вывести на экран информацию о видах символов (например, 10 букв и 12 цифр).
13. Ввести две строки. Проверить строки на равенство друг другу.
14. Ввести строку. В данной строке заменить все четные строчные буквы на прописные.
15. Ввести две строки. Сравнить строки и, в самой короткой строке отсортировать ее элементы по возрастанию (допускается использование только цифровых строк).
16. Дана строка. Вывести подстроку, расположенную между первой и второй точками исходной строки. Если в строке менее двух точек, то вывести всю исходную строку.
17. Дана строка, состоящая из латинских слов, разделенных пробелами. Определить количество слов в строке.

18. Дана строка, состоящая из латинских слов, разделенных пробелами. Определить количество слов, которые начинаются и заканчиваются одной и той же буквой (содержат хотя бы одну букву "w").

19. Дана строка, состоящая из латинских слов, разделенных пробелами. Определить количество слов, которые содержат ровно три буквы "o".

20. Дана строка, состоящая из латинских слов, разделенных пробелами. Определить длину самого короткого слова.

21. Дана строка, состоящая из латинских слов, разделенных пробелами. Вывести строку, содержащую эти же слова, но разделенные одним символом "." (точка). В конце точку не ставить.

22. Дана строка, состоящая из латинских слов, разделенных пробелами. Вывести строку, содержащую эти же слова, разделенные одним пробелом, но расположенные в обратном порядке.

23. Дана строка, состоящая из латинских слов, разделенных пробелами. Преобразовать каждое слово в строке, удалив из него все последующие вхождения первой буквы этого слова (количество пробелов между словами не изменять).

24. Дана строка, состоящая из латинских слов, разделенных пробелами. Вывести строку, содержащую эти же слова, (разделенные одним пробелом, но расположенные в алфавитном порядке).

25. Дана строка-предложение на латинском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.

26. Дана строка-предложение на латинском языке. Подсчитать количество содержащихся в строке знаков препинания.

27. Дана строка-предложение на латинском языке. Вывести самое короткое слово в предложении. Если таких слов несколько, то вывести первое из них.

28. Дана строка-предложение, содержащая избыточные пробелы. Преобразовать ее так, чтобы между словами был ровно один пробел.

29. Дана строка, содержащая полное имя файла, то есть имя диска, список каталогов (путь), собственно имя и расширение. Выделить из этой строки имя и расширение файла.

30. Дана строка, содержащая полное имя файла. Выделить из строки название последнего каталога (без символов "\"). Если файл содержится в корневом каталоге, то вывести символ "\" вместо названия каталога.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть лабораторной работы.

2. Реализовать индивидуальное задание по вариантам, представленные в теоретических сведениях, сделать скриншоты работающих программ. Написать комментарии.

3. Написать отчет, содержащий:

1. Титульный лист, на котором указывается:

а) полное наименование министерства образования и название учебного заведения;

б) название дисциплины;

в) номер практического занятия;

г) фамилия преподавателя, ведущего занятие;

д) фамилия, имя и номер группы студента;

е) год выполнения лабораторной работы.

2. Индивидуальное задание из раздела «Теоретические сведения» с кодом, комментариями и скриншотами работающих программ.

3. Построение блок-схем.