

Лабораторная работа №6

Тема: Разработка и отладка алгоритмов и программ по обработке строк символов.

Цель занятия: Выработать умения и навыки работы со средой программирования Microsoft Visual Studio.

Время выполнения: 2 часа.

Содержание работы и последовательность ее исполнения.

1. Запустить Microsoft Visual Studio выбрав пункт меню «Пуск → Все программы → Microsoft Visual Studio → Microsoft Visual Studio».
2. Для каждого задания создавать отдельный пустой консольный проект (см. л.р. 1).

Задание 1. Посимвольная обработка и формирование строк

Вариант 1

Дана строка. Подсчитать количество содержащихся в ней цифр.

Вариант 2

Дана строка. Подсчитать количество содержащихся в ней прописных латинских букв.

Вариант 3

Дана строка. Преобразовать в ней все строчные буквы (латинские) в прописные.

Вариант 4

Дана строка. Преобразовать в ней все прописные латинские буквы (латинские) в строчные.

Вариант 5

Дана строка. Подсчитать количество содержащихся в ней знаков препинания.

Вариант 6

Дана непустая строка S. Вывести строку, содержащую символы строки S, между которыми вставлено по одному пробелу.

Вариант 7

Дано четное число $N (> 0)$ и символы C1 и C2. Вывести строку длины N, которая состоит из чередующихся символов C1 и C2, начиная с C1.

Вариант 8

Дан символ C, изображающий цифру или букву. Если C изображает цифру, то вывести строку «digit», если букву – вывести строку «char».

Вариант 9

Дано целое число $N (1 \leq N \leq 26)$. Сформировать и вывести строку из N первых прописных букв латинского алфавита.

Вариант 10

Дана строка. Вывести строку, содержащую те же символы, но расположенные в обратном порядке.

Вариант 11

Дано целое число $N (> 0)$ и символ C. Сформировать и вывести строку длины N, которая состоит из символов C.

Вариант 12

Дано целое число $N (1 \leq N \leq 26)$. Сформировать и вывести строку из N последних строчных букв латинского алфавита в обратном порядке (начиная с буквы «z»).

Вариант 13

Дана строка. Подсчитать количество содержащихся в ней строчных латинских букв.

Вариант 14

Дана строка. Подсчитать количество содержащихся в ней отличных от цифр символов.

Вариант 15

Дана строка, изображающая целое положительное число. Вывести сумму цифр этого числа.

Задание 2. Обработка строк с помощью стандартных функций. Поиск и замена

В заданиях, связанных с поиском и заменой подстрок, можно считать, что исходная строка не содержит перекрывающихся вхождений требуемых подстрок. Кроме этого, можно также считать, что удаление или замена любого вхождения подстроки не приведет к появлению в строке новых вхождений данной подстроки.

Вариант 1

Дано целое число $N (> 0)$ и строка S . Преобразовать строку S в строку длины N следующим образом: если длина строки S больше N , то отбросить первые символы, если длина строки S меньше N , то в ее начало добавить символы «.» (точка).

Вариант 2

Даны целые положительные числа $N1$ и $N2$ и строки $S1$ и $S2$. Получить из этих строк новую строку, содержащую первые $N1$ символов строки $S1$ и последние $N2$ символов строки $S2$ (в указанном порядке).

Вариант 3

Дан символ C и строка S . Удвоить каждое вхождение символа C в строку S .

Вариант 4

Дан символ C и строки S , $S0$. Перед каждым вхождением символа C в строку S вставить строку $S0$.

Вариант 5

Дан символ C и строки S , $S0$. После каждого вхождения символа C в строку S вставить строку $S0$.

Вариант 6

Даны строки S и $S0$. Проверить, содержится ли строка $S0$ в строке S . Если содержится, то вывести True, если не содержится, то вывести False.

Вариант 7

Даны строки S и $S0$. Найти количество вхождений строки $S0$ в строку S .

Вариант 8

Даны строки S и $S0$. Удалить из строки S первую подстроку, совпадающую с $S0$. Если совпадающих подстрок нет, то вывести строку S без изменений.

Вариант 9

Даны строки S и $S0$. Удалить из строки S последнюю подстроку, совпадающую с $S0$. Если совпадающих подстрок нет, то вывести строку S без изменений.

Вариант 10

Даны строки S и $S0$. Удалить из строки S все подстроки, совпадающие с $S0$. Если совпадающих подстрок нет, то вывести строку S без изменений.

Вариант 11

Даны строки S , $S1$ и $S2$. Заменить в строке S первое вхождение строки $S1$ на строку $S2$.

Вариант 12

Даны строки S , $S1$ и $S2$. Заменить в строке S последнее вхождение строки $S1$ на строку $S2$.

Вариант 13

Даны строки S , $S1$ и $S2$. Заменить в строке S все вхождения строки $S1$ на строку $S2$.

Вариант 14

Дана строка, содержащая по крайней мере один символ пробела. Вывести подстроку, расположенную между первым и вторым пробелом исходной строки. Если строка содержит только один пробел, то вывести пустую строку.

Вариант 15

Дана строка, содержащая по крайней мере один символ пробела. Вывести подстроку, расположенную между первым и последним пробелом исходной строки. Если строка содержит только один пробел, то вывести пустую строку.

Задание 3. Анализ и преобразование слов в строке

Каждая программа должна непрерывно считывать с клавиатуры строки символов и проделывать с ними операции указанные в задании. Прекратить выполнение программы необходимо в том случае если введена пустая строка.

Пример выполнения:

Написать программу для выделения из введенного предложения слов и вывода их в столбик в алфавитном порядке.

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[200]; // строка для ввода предложения
    char words[50][50]; // массив для выделения слов

    while(true) { // вечный цикл считывания строк
        gets(str); // считываем строку
        if(strlen(str)==0) return 0; // если строка пустая, то выходим
        strcat(str, " "); // добавляем к строке пробел, для упрощения обработки
        int numwords=0; // кол-во найденных слов
        // цикл извлечения слов
        while(strlen(str)!=0) {
            // удаляем все пробелы в начале строки
            while(str[0]==' ') strcpy(str,&str[1]);
            char *tmp=strchr(str, ' '); // ищем позицию пробела в строке
            if(tmp!=NULL) { // если пробел найден, т.е. позиция не нулевая
                strcpy(words[numwords],str); // копируем строку в массив
                words[numwords][tmp-str]='\0'; // отсекаем слово по размеру
                strcpy(str,tmp); // копируем текст после найденного слова, т.е. удаляем найденное
                numwords++; // увеличиваем счетчик слов
            }
        }
        // сортировка пузырьком
        for(int i=0; i<numwords; i++)
            for(int j=0; j<numwords; j++)
                if(strcmp(words[i],words[j])<0) {
                    char tmp[50];
                    strcpy(tmp,words[i]);
                    strcpy(words[i],words[j]);
                    strcpy(words[j],tmp);
                }

        // выводим результат
        for(int i=0; i<numwords; i++)
            printf("%20s\n",words[i]);
    }
    return 0;
}
```

Вариант 1

Исправить регистр во введенном предложении, т.е. каждое новое предложение должно начинаться с большой буквы, все остальные буквы должны быть малыми.

Вариант 2

Определить, содержится ли в предложении введенное с клавиатуры слово.

Вариант 3

Проверить введенное предложение на содержание двузначных чисел.

Вариант 4

Выделить и вывести на экран слова из предложений, начинающиеся с гласных букв.

Вариант 5

Вывести на экран предложение, меняя местами каждые два соседних слова.

Вариант 6

Удалить из предложения все запятые и вывести его на экран.

Вариант 7

Определить, сколько в предложении слов, состоящих не более чем из четырех букв.

Вариант 8

Выделить и вывести на экран цитаты, то есть участки предложения, заключенные в кавычки.

Вариант 9

Подсчитать и вывести на экран количества слов во введенном предложении.

Вариант 10

Выделить и вывести на экран слова из предложений, заканчивающиеся на согласные буквы.

Вариант 11

Выделить из предложения все слова и вывести их по отдельности в столбик.

Вариант 12

Определить начинается ли введенное предложение на тире, перед которым могут находиться пробелы.

Вариант 13

Заменить во введенном предложении каждую первую букву слова на прописную.

Вариант 14

Заменить во введенном предложении цифры от 0 до 9 словами «ноль», «один», «девять».

Вариант 15

Найти в предложении самое длинное слово.

Задание 4. Дополнительные задания на обработку строк

Вариант 1

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

Вариант 2

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

Вариант 3

Дана строка, содержащая полное имя файла. Выделить из этой строки название первого каталога (без символов «\»). Если файл содержится в корневом каталоге, то вывести символ «\».

Вариант 4

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

Вариант 5

Дана строка-предложение на латинском языке. Зашифровать ее, выполнив циклическую замену каждой буквы на следующую за ней в алфавите и сохранив при этом регистр букв («А» перейдет в «В», «а» — в «а», «В» — в «С», «z» — в «а» и т. д.). Знаки препинания и пробелы не изменять.

Вариант 6

Дана строка-предложение на латинском языке и число K ($0 < K < 10$). Зашифровать строку, выполнив циклическую замену каждой буквы на букву того же регистра, расположенную в алфавите на K -й позиции после шифруемой буквы (например, для $K = 2$ «А» перейдет в «С», «а» — в «с», «В» — в «D», «z» — в «b» и т. д.). Знаки препинания и пробелы не изменять.

Вариант 7

Дано зашифрованное предложение на латинском языке (способ шифрования описан выше) и кодовое смещение K ($0 < K < 10$). Расшифровать предложение.

Вариант 8

Дано зашифрованное предложение на латинском языке (способ шифрования описан выше) и его расшифрованный первый символ C . Найти кодовое смещение K и расшифровать предложение.

Вариант 9

Дана строка-предложение. Зашифровать ее, поместив вначале все символы, расположенные на четных позициях строки, а затем, в обратном порядке, все символы, расположенные на нечетных позициях (например, строка «Программа» превратится в «ргамамроП»).

Вариант 10

Дано предложение, зашифрованное по правилу, описанному выше. Расшифровать это предложение.

Вариант 11

Дана строка, содержащая цифры и строчные латинские буквы. Если буквы в строке упорядочены по алфавиту, то вывести 0; в противном случае вывести номер первого символа строки, нарушающего алфавитный порядок.

Вариант 12

Дана строка, содержащая латинские буквы и круглые скобки. Если скобки расставлены правильно (то есть каждой открывающей соответствует одна закрывающая), то вывести число 0. В противном случае вывести или номер позиции, в которой расположена первая ошибочная закрывающая скобка, или, если закрывающих скобок не хватает, число -1.

Вариант 13

Дана строка, содержащая латинские буквы и скобки трех видов: «()», «[]», «{}». Если скобки расставлены правильно (то есть каждой открывающей соответствует закрывающая скобка того же вида), то вывести число 0. В противном случае вывести или номер позиции, в которой расположена первая ошибочная скобка, или, если закрывающих скобок не хватает, число -1.

Вариант 14

Дана строка. Подсчитать и вывести в виде таблицы количество уникальных символов входящих в предложение.

Вариант 15

Дана строка. Проверить, является ли строка верным арифметическим выражением. Учитывать круглые скобки и простейшие арифметические операции. Числа в выражении могут быть вещественными.