

Тема 9. ОБРАБОТКА СТРОК

Цель занятия: овладение навыками алгоритмизации и программирования задач, обрабатывающих символьную информацию; получение практических навыков в организации ввода-вывода строк, их обработки, в использовании встроенных процедур и функций работы со строками.

Теоретические сведения

Для символьных данных в C++ введен тип *char*.

Строка на языке C++ – это массив символов, заканчивающийся нуль-символом. Пример определения строки в программе:

```
int main()
{
    char str[10];
    return 0;
}
```

В строковом литерале всегда на 1 символ больше, чем используется при записи; он всегда заканчивается нулевым символом '\0'.

Например:

```
sizeof("Бор") = 4;
```

Присвоить значение строке с помощью оператора присваивания нельзя, так как для массивов не определена операция прямого присваивания. Поместить строку в символьный массив можно либо при вводе, либо с помощью инициализации:

```
char s1[] = "ABCDEF";           //инициализация строки
char s2[]={ 'A','B','C','D','E','F','\0'}; //инициализация строки
```

Для ввода и вывода символьных данных в библиотеке языка C определены следующие функции:

int getchar(void) – осуществляет ввод одного символа из входного потока, при этом она возвращает один байт информации (символ) в виде значения типа *int*. Это сделано для распознавания ситуации, когда при чтении будет достигнут конец файла.

int putchar (int c) – помещает в стандартный выходной поток символ *c*.

char* gets(char*s) – считывает строку *s* из стандартного потока до появления символа '\n', сам символ '\n' в строку не заносится.

int puts(const char* s) записывает строку в стандартный поток, добавляя в конец строки символ '\n', в случае удачного завершения возвращает значение больше или равное 0 и отрицательное значение (EOF=-1) в случае ошибки.

Пример

1. *char s[20];*

cin>>s; //ввод строки из стандартного потока

cout<<s; //вывод строки в стандартный поток

Результат работы программы:

При вводе строки "123 456 789", чтение байтов осуществляется до первого пробела (при отсутствии пробела до конца строки), т.е. в строку *s* занесется только первое слово строки "123/0", следовательно, выведется: 123.

2. *char s[20];*

gets(s); //ввод строки из стандартного потока

puts(s); //вывод строки в стандартный поток

Результат работы программы:

При вводе строки “123 456 789”, чтение байтов осуществляется до символа ‘\n’, т.е. в s занесется строка “123 456 789\n\0”, при выводе строки функция *puts* возвращает еще один символ ‘\n’, следовательно, будет выведена строка “123 456 789\n\n”.

3. *char s[20];*

scanf("%s",s); /*ввод строки из стандартного потока*/

printf("%s",s); /*вывод строки в стандартный поток*/

Результат работы программы:

При вводе строки “123 456 789”, чтение байтов осуществляется до первого пробела, т.е. в строку s занесется только первое слово строки “123/0”, следовательно, выведется: 123. Т.к. s – имя массива, т.е. адрес его первого элемента, операция & в функции *scanf* не используется.

4. Прочитать несколько строк текста можно так:

char str[100];

cin.getline(str, 100, '');*

Второй параметр этой функции (100) задаёт максимальную длину строки, третий (*') – по какому символу прекратить ввод.

Функции работы с символьными строками (объявлены в заголовочном файле **<stdlib.h>**):

*double atof (const char *str)* // преобразует строку str в вещественное число
//типа double

*int atoi (const char *str)* // преобразует строку str в целое число типа int

*long atol (const char *str)* // преобразует строку str в целое число типа long

*char *itoa (int v, char *str, int baz)* //преобразует целое v в строку str. При
//изображении числа используется основание baz
//(2<=baz<=36)

*char *ltoa (long v, char *str, int baz)* // преобразует длинное целое v в строку str. При
//изображении числа используется основание baz
//(2<=baz<=36)

*char *ultoa (unsigned long v, char *str, int baz)* // преобразует беззнаковое длинное
//целое v в строку str второй строки

Функции работы с символьными строками (объявлены в заголовочном файле **<string.h>**):

char strcpy(char* p, const char* q)* //копирование из q в p

char strcat(char* p, const char* q)* //добавление q к p

size_t strlen(const char p)* //длина p (не считая конца строки)

int strcmp(const char p, const char* q)* //сравнение p и q; меньше (-1), равно (0),
//больше(1)

*char *strchr(const char *, int)* // находит первое вхождение заданного
//символа в строке

*char *strncat(char *, const char *, size_t)* // добавляет заданное число символов
//второй строки в конец первой

*int strncmp(const char *, const char *, size_t)* // сравнивает заданное число символов
//двух строк

*char *strncpy(char *, const char *, size_t)* // копирует заданное число символов
//второй строки в первую строку

*char *strrchr(const char *, int)* // находит последнее вхождение заданного
//символа в строке

*char *strstr(const char *, const char *)* // находит первое вхождение второй

//строки как подстроки первой

Из файла <ctype.h>:

```
int isdigit(int); // определяет, цифра или нет
int isalpha(int); //буква
int isupper(int); //буква в верхнем регистре
int islower(int); //буква в нижнем регистре
int isspace(int); //символ – разделитель
int ispunct(int); //символ пунктуации (ни один из вышеупомянутых)
int isalnum(int); //буква или цифра
int toupper(int); //перевод в верхний регистр
int tolower(int); //перевод в нижний регистр
```

Задача 9.1. Задать строку (массив символов) и подсчитать количество вопросительных знаков в ней:

```
#include <iostream>
using namespace std;
int main()
{
    int i, nQuestCount=0;
    cout<<"InputLine: ";
    char sInputLine[20];
    cin >> sInputLine;
    for(i = 0; i < strlen(sInputLine); i++)
        if(sInputLine[i]=='?') ++ nQuestCount;
    cout<<"QuestCount= " <<nQuestCount << "\n";
    return 0;
}
```

Задание 9.1. Задать строку (массив символов) и подсчитать количество цифр, обрамленных знаком *.

Задача 9.2. Составить программу для вычеркивания из слов текста всех букв, стоящих на нечетных позициях после буквы "a".

```
#include <iostream>
using namespace std;
int main()
{
    char s1[]="asdf garts aaag";
    char s2[17];
    cout<<s1<<"\n";
    int j = 0;
    s2[j++] = s1[0];
    for(int i=1; s1[i]!='\0'; ++i)
    {
        if(s1[i-1] == 'a' && i%2 == 1)
            continue;
        else
            s2[j++] = s1[i];
    }
    s2[j] = '\0';
    strcpy_s(s2, s1); //копировать s2 в s1
}
```

```
printf("After deleting letters:\n");
cout<<s1;
}
```

Задание 9.2. Составить программу для вычеркивания из слов текста всех букв, стоящих на четных позициях после символа "-".

Словами будем называть группы символов, разделённые пробелами (одним или несколькими) и не содержащие пробелов внутри себя.

Задача 9.3. Слова в тексте разделены пробелами. Посчитать количество слов.

```
#include <iostream>
using namespace std;
void main()
{
    char str[100];
    cout<<"InputLine: ";
    cin.getline(str, 100, '\n');
    int counter = 0, i;
    for(i=1; str[i]!='\0'; ++i)
        if(str[i-1]!=' ' && str[i]==' ') ++counter;
    if(str[i-1]!=' ') ++counter;
    cout<<counter<<" words";
}
```

Задача 9.4. Дана строка, состоящая из слов, разделенных пробелами (одним или несколькими). Записать все слова в массив.

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    char str[100];
    cout<<"InputLine: ";
    cin.getline(str, 100, '\n');
    int k = 0;
    int len = strlen(str); //длина введенного текста
    char words[50][30]; //массив для хранения слов
    int i = 0;
    int begin = 0, end = 0;
    int count = 0;
    while (i < len)
    {
        while (str[i] == ' ' && i < len)
            ++i; //пропустить пробелы
        begin = i; //номер первого символа слова
        while (str[i] != ' ' && i < len)
            ++i; //пропустить все символы слова
        end = i; //номер символа, следующего за последним символом слова

        strncpy_s(words[count], &str[begin], end-begin); //записать слово в массив
        count++;
    }
}
```

```

        words[count++][end-begin] = '\0';           // добавить символ конца строки
    }
    for(i = 0; i < count; ++i)
        cout<<words[i]<<"\n";
    return 0;
}

```

Задача 9.5. Задана строка символов, в которой слова отделяются друг от друга одним или несколькими пробелами. Вывести на экран все слова этой строки.

```

#include <string.h>
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
    char st[100], sl[100];
    int k = 0, i;
    cout<<"Vvedite stroku";
    cin.getline(st,100,'\n');
    strcat_s(st, " ");
    cout<<"Slova: ";
    int n = strlen(st);
    sl[0] = '\0';
    for (i=0; i<n; i++)
        if (st[i] != ' ')
        {
            sl[k] = st[i];
            sl[k+1] = '\0';
            k++;
        }
        else
        {
            if (strlen (sl) > 0) cout<<sl<<"\n";
            sl[0] = '\0';
            k = 0;
        }
    system("pause");
    return 0;
}

```

Аудиторные и домашние задания

1. Ввести строку символов и выполнить над ней следующие действия:
 - а) подсчитать количество строчных букв латинского алфавита;
 - б) вставить после каждой цифры по 2 пробела;
 - в) заменить каждую цифру в строке на следующую (0 на 1, 1 на 2,..., 9 на 0);
 - г) получить новую строку, в которую входят все символы исходной строки без повторов;
 - д) удалить в строке все знаки +, -, *, /;
 - е) вывести вначале те буквы латинского алфавита, которые встретились в строке, а потом те буквы, которых нет в тексте. Не различать прописные и строчные буквы;

- ж) определить символ с максимальным кодом.
2. Удалить начало строки до первого появления двух одинаковых символов, следующих подряд.
 3. Включить в строку заданную последовательность символов после появления первой цифры.
 4. Выбрать из строки последовательности символов, ограниченные с обеих сторон заданным символом.
 5. Ввести два целых числа X,Y (типа long). Получить строку, отображающую результат суммирования этих двух чисел (например, при $X=35000, Y=65234$ должны получить строку "35000+65234=100234").
 6. В строке символов слова разделены пробелами. Выполнить следующие действия:
 - а) получить столько строк, сколько слов в исходной строке, вывести полученные строки;
 - б) вывести все слова строки в обратном порядке;
 - в) вывести те слова, перед которыми находится более короткое слово, а после него – более длинное;
 - г) удалить в строке одинаковые слова;
 - д) вывести те слова, которые совпадают с начальным отрезком латинского алфавита (a, ab, abc и т.д.);
 - е) во всех словах перенести последний символ в начало слова;
 - ж) в каждом слове нечетной длины удалить его средний символ.
 7. Дана строка, состоящая из слов, разделенных пробелами (одним или несколькими). Найти порядковый номер слова максимальной длины.
 8. Дана строка. Определить, сколько слов строки начинается и заканчивается на одну и ту же букву. Вывести эти слова и их количество.
 9. Дана строка и два слова. Определить, есть ли эти слова в строке и сколько раз каждое из них встречается в строке. Если их в строке нет, дать соответствующее сообщение.
 10. Дана строка и подстрока из двух (трех) символов. Найти в строке заданную подстроку и удалить ее из всех слов строки. Оставшийся текст сжать. Определить количество удаленных подстрок.
 11. Дана строка и символ. Определить и вывести слова, в которых встречается этот символ, и номер позиции, в которой он размещен в слове в первый раз.
 12. Дана строка. Вывести самое короткое слово строки и его длину.
 13. Выделить из исходной строки части строки в круглых скобках вместе со скобками; вложенных скобок нет. Определить количество таких частей текста и в каждом из них количество латинских букв.
 14. Дана строка. Вывести те слова строки, которые отличны от последнего и симметричны.
 15. Дана строка. Вывести те слова строки, которые отличны от последнего и первая буква слова входит в него еще раз.
 16. Дана строка. Вывести те слова строки, которые отличны от первого и буквы слова упорядочены по алфавиту.
 17. Дана строка. Вывести те слова строки, которые отличны от первого, удалив из них первую и последнюю буквы.
 18. Дана строка. Вывести те слова строки, которые отличны от последнего, поменяв в них местами первую и последнюю буквы.
 19. Слова в строке разделены знаками препинания. Отсортировать слова в алфавитном порядке.
 20. Строка состоит из слов, разделенных пробелами. Выбрать вначале те из них, которые являются записью числа типа long, затем - double, а затем те, которые нельзя преобразовать в число.

Контрольные вопросы

1. Каковы особенности работы со строками?
2. Как описываются строки?
3. Чем всегда должна заканчиваться строка?
4. Какие функции определены для ввода и вывода символьных данных в библиотеке языка C?
5. Какие стандартные процедуры и функции работы со строками вы знаете?