**Лабораторная работа N 1  
РАБОТА С СИМВОЛЬНИМИ СТРОКАМИ**

**1. Цель работы**

Закрепление практических навыков в работе с массивами и указателями языка C, обеспечении функциональной модульности

**2. Темы для предварительного изучения**

* Указатели в языке C.
* Представление строк.
* Функции и передача параметров.

**3. Постановка задачи**

По индивидуальному заданию создать функцию для обработки символьных строк. За образец брать библиотечные функции обработки строк языка C, но не применять их в своей функции. Предусмотреть обработку ошибок в задании параметров и особые случаи. Разработать два варианта заданной функции - используя традиционную обработку массивов и используя адресную арифметику.

**4. Индивидуальные задания**

|  |  |
| --- | --- |
| **№пк** | **Вариант** |
| 1 | Функция - Copies(s,s1,n). Назначение - копирование строки s в строку s1 n раз |
| 2 | Функция - Words(s). Назначение - подсчет слов в строке s |
| 3 | Функция - Concat(s1,s2). Назначение - конкатенация строк s1 и s2 (аналогичная библиотечная функция C - strcat) |
| 4 | Функция - Parse(s,t). Назначение - разделение строки s на две части: до первого вхождения символа t и после него |
| 5 | Функция - Center(s1,s2,l). Назначение - центрирование - размещение строки s1 в середине строки s2 длиной l |
| 6 | Функция - Delete(s,n,l). Назначение - удаление из строки s подстроки, начиная с позиции n, длиной l (аналогичная библиотечная Функция есть в Pascal). |
| 7 | Функция - Left(s,l). Назначение - выравнивание строки s по левому краю до длины l. |
| 8 | Функция - Right(s,l) Назначение - выравнивание строки s по правому краю до длины l. |
| 9 | Функция - Insert(s,s1,n). Назначение - вставка в строку s подстроки s1, начиная с позиции n (аналогичная библиотечная функция есть в Pascal). |
| 10 | Функция - Reverse(s). Назначение - изменение порядка символов в строке s на противоположный. |
| 11 | Функция - Pos(s,s1). Назначение - поиск первого вхождения подстроки s1 в строку s (аналогичная функция C - strstr). |
| 12 | Функция - LastPos(s,s1). Назначение - поиск последнего вхождения подстроки s1 в строку s. |
| 13 | Функция - WordIndex(s,n). Назначение - определение позиции начала в строке s слова с номером n. |
| 14 | Функция - WordLength(s,n). Назначение - определение длины слова с номером n в строке s. |
| 15 | Функция - SubWord(s,n,l). Назначение - выделение из строки s l слов, начиная со слова с номером n. |
| 16 | Функция - WordCmp(s1,s2). Назначение - сравнение строк (с игнорированием множественных пробелов). |
| 17 | Функция - StrSpn(s,s1). Назначение - определение длины той части строки s, которая содержит только символы из строки s1. |
| 18 | Функция - StrCSpn(s,s1). Назначение - определение длины той части строки s, которая не содержит символы из строки s1. |
| 19 | Функция - Overlay(s,s1,n). Назначение - перекрытие части строки s, начиная с позиции n, строкой s1. |
| 20 | Функция - Replace(s,s1,s2). Назначение - замена в строке s комбинации символов s1 на s2. |
| 21 | Функция - Compress(s,t). Назначение - замена в строке s множественных вхождений символа t на одно. |
| 22 | Функция - Trim(s). Назначение - удаление начальных и конечных пробелов в строке s. |
| 23 | Функция - StrSet(s,n,l,t). Назначение - установка l символов строки s, начиная с позиции n, в значение t. |
| 23 | Функция - Space(s,l). Назначение - доведение строки s до длины l путем вставки пробелов между словами. |
| 24 | Функция - Findwords(s,s1). Назначение - поиск вхождения в строку s заданной фразы (последовательности слов) s1. |
| 25 | Функция - StrType(s). Назначение - определение типа строки s (возможные типы - строка букв, десятичное число, 16-ричное число, двоичное число и т.д.). |
| 26 | Функция - Compul(s1,s2). Назначение - сравнение строк s1 и та s2 с игнорированием различий в регистрах. |
| 27 | Функция - Translate(s,s1,s2). Назначение - перевод в строке s символов, которые входят в алфавит s1, в символы, которые входят в алфавит s2. |
| 28 | Функция - Word(s). Назначение - выделение первого слова из строки s. |
| Примечание: под "словом" везде понимается последовательность символов, которая не содержит пробелов. | |

**5. Пример решения задачи**

**5.1. Индивидуальное задание:**

* Функция - substr(s,n,l). Назначение - выделение из строки s подстроки, начиная с позиции n, длиной l.

**5.2. Описание метода решения**

**5.2.1.** Символьная строка в языке C представляется в памяти как массив символов, в конце которого находится байт с кодом 0 - признак конца строки. Строку, как и любой другой массив можно обрабатывать либо традиционным методом - как массив, с использованием операции индексации, либо через указатели, с использованием операций адресной арифметики. При работе со строкой как с массивом нужно иметь в виду, что длина строки заранее неизвестна, так что циклы должны быть организованы не со счетчиком, а до появления признака конца строки.

**5.2.2.** Функция должна реализовывать поставленную задачу - и ничего более. Это означает, что функцию можно будет, например, перенести без изменений в любую другую программу, если спецификации функции удовлетворяют условиям задачи. Это также означает, что при ошибочном задании параметров или при каких-то особых случаях в их значениях функция не должна аварийно завершать программу или выводить какие-то сообщения на экран, но должна возвращать какое-то прогнозируемое значение, по которому та функция, которая вызвала нашу, может сделать вывод об ошибке или об особом случае.

Определим состав параметров функции:

int substr (src, dest, num, len);

|  |  |  |
| --- | --- | --- |
| где | src - | строка, с которой выбираются символы; |
|  | dest - | строка, в которую записываются символы; |
|  | num - | номер первого символа в строке src, с которого начинается подстрока (нумерация символов ведется с 0); |
|  | len - | длина выходной строки. |

Возможные возвращаемые значения функции установим: 1 (задание параметров правильное) и 0 (задание не правильное). Эти значения при обращениях к функции можно будет интерпретировать как "истина" или "ложь".

Обозначим через Lsrc длину строки src. Тогда возможны такие варианты при задании параметров:

* num+len <= Lsrc - полностью правильное задание;
* num+len > Lsrc; num < Lsrc - правильное задание, но длина выходной строки буде меньше, чем len;
* num >= Lsrc - неправильное задание, выходная строка будет пустой;
* num < 0 или len <= 0 - неправильное задание, выходная строка будет пустой.

Заметим, что интерпретация конфигурации параметров как правильная / неправильная и выбор реакции на неправильное задание - дело исполнителя. Но исполнитель должен строго выполнять принятые правила. Возможен также случай, когда выходная строка выйдет большей длины, чем для нее отведено места в памяти. Однако, поскольку нашей функции неизвестен размер памяти, отведенный для строки, функция не может распознать и обработать этот случай - так же ведут себя и библиотечные функции языка C.

**5.3. Описание логической структуры**

**5.3.1.** Программа состоит из одного программного модуля - файл LAB1.C. В состав модуля входят три функции - main, substr\_mas и subs\_ptr. Общих переменных в программе нет. Макроконстантой N определена максимальная длина строки - 80.

**5.3.2.** Функция main является главной функцией программы, она предназначена для ввода исходных данных, вызова других функций и вывода результатов. В функции определены переменные:

* + ss и dd - входная и выходная строки соответственно;
  + n - номер символа, с которого должна начинаться выходная строка;
  + l - длина выходной строки.

Функция запрашивает и вводит значение входной строки, номера символа и длины. Далее функция вызывает функцию substr\_mas, передавая ей как параметры введенные значения. Если функция substr\_mas возвращает 1, выводится на экран входная и выходная строки, если 0 - выводится сообщение об ошибке и входная строка. Потом входная строка делается пустой и то же самое выполняется для функции substr\_ptr.

**5.3.3.** Функция substr\_mas выполняет поставленное задание методом массивов. Ее параметры: - src и dest - входная и выходная строки соответственно, представленные в виде массивов неопределенного размера; num и len. Внутренние переменные i и j используются как индексы в массивах.

Функция проверяет значения параметров в соответствии со случаем 4, если условия этого случая обнаружены, в первый элемент массива dest записывается признак конца строки и функция возвращает 0.

Если случай 4 не выявлен, функция просматривает num первых символов входной строки. Если при этом будет найден признак конца строки, это - случай 3, при этом в первый элемент массива dest записывается признак конца строки и функция возвращает 0.

Если признак конца в первых num символах не найден, выполняется цикл, в котором индекс входного массива начинает меняться от 1, а индекс выходного - от 0. В каждой итерации этого цикла один элемент входного массива пересылается в выходной. Если пересланный элемент является признаком конца строки (случай 2), то функция немедленно заканчивается, возвращая 1. Если в цикле не встретится конец строки, цикл завершится после len итераций. В этом случае в конец выходной строки записывается признак конца и Функция возвращает 1.

**5.3.4.** Функция substr\_ptr выполняет поставленное задание методом указателей. Ее параметры: - src и dest - входная и выходная строки соответственно, представленные в виде указателей на начала строк; num и len.

Функция проверяет значения параметров в соответствии со случаем 4, если условия этого случая выявлены, по адресу, который задает dest, записывается признак конца строки и функция возвращает 0, эти действия выполняются одним оператором.

Если случай 4 не обнаружен, функция пропускает num первых символов входной строки. Это сделано циклом while, условием выхода из которого является уменьшение счетчика num до 0 или появление признака конца входной строки. Важно четко представлять порядок операций, которые выполняются в этом цикле:

* + выбирается счетчик num;
  + счетчик num уменьшается на 1;
  + если выбранное значение счетчика было 0 - цикл завершается;
  + если выбранное значение было не 0 - выбирается символ, на который указывает указатель src;
  + указатель src увеличивается на 1;
  + если выбранное значение символа было 0, т.е., признак конца строки, цикл завершается, иначе - повторяется.

После выхода из цикла проверяется значение счетчика num: если оно не 0, это означает, что выход из цикла произошел по признаку конца строки (случай 3), по адресу, который задает dest, записывается признак конца строки и функция возвращает 0.

Если признак конца не найден, выполняется цикл, подобный первому циклу while, но по счетчику len. В каждой итерации этого цикла символ, на который показывает src переписывается по адресу, задаваемому dest, после чего оба указателя увеличиваются на 1. Цикл закончится, когда будет переписано len символов или встретится признак конца строки. В любом варианте завершения цикла по текущему адресу, который содержится в указателе dest, записывается признак конца строки и функция завершается, возвращая 1.

**5.4. Данные для тестирования**

Тестирование должно обеспечить проверку работоспособности функций для всех вариантов входных данных. Входные данные, на которых должно проводиться тестирование, сведены в таблицу:

|  |
| --- |
| **вариант src num len dest** |
|  |
| 1 012345 2 2 23  012345 0 1 0  012345 0 6 012345 |
| 2 012345 5 3 5  012345 2 6 2345  012345 0 7 012345 |
| 3 012345 8 2 пусто |
| 4 012345 -1 2 пусто  012345 5 0 пусто  012345 5 -1 пусто |

### 

### Содержание отчета

Отчет оформляется по каждой лабораторной работе и состоит из следующих разделов.

4.1. Тема лабораторной работы.

4.2. Цель работы.

4.3. Индивидуальное задание.

4.4. Описание структур данных и алгоритмов

4.5. Результаты работы программы.

4.6. Интерпретация результатов.