

Лабораторная работа №9

Строки

Задания для самостоятельного выполнения

Напишите программу, решающую задачу по номеру своего варианта.

Требования и ограничения

Необходимые действия реализовать в виде функций.

Указания по выполнению работы

Для использования строковых функций необходимо включить в программу заголовочный файл `<string.h>`. Ниже перечислены строковые функции, которые могут быть полезны при выполнении данной работы. Для более глубокого понимания работы со строками обратитесь к курсу лекций и дополнительной литературе.

| Прототип функции | Описание |
|---|--|
| <code>size_t strlen(const char *str)</code> | возвращает длину <code>str</code> |
| <code>char *strcat(char *str1, char *str2)</code> | присоединяет к <code>str1</code> копию <code>str2</code> и завершает <code>str1</code> нулевым символом. Возвращает <code>str1</code> |
| <code>char *strchr(char *str, int ch)</code> | возвращает указатель на первое вхождение младшего байта <code>ch</code> в <code>str</code> или <code>NULL</code> , если символ не найден |
| <code>char *strrchr(char *str, int ch)</code> | возвращает указатель на последнее вхождение младшего байта <code>ch</code> в <code>str</code> или <code>NULL</code> , если символ не найден |
| <code>char *strstr(char *str1, char *str2)</code> | возвращает указатель на первое вхождение <code>str2</code> в <code>str1</code> или <code>NULL</code> , если вхождения нет |
| <code>int strcmp(char *str1, char *str2)</code> | сравнивает в лексикографическом порядке <code>str1</code> и <code>str2</code> и возвращает значение: меньше нуля, если <code>str1 < str2</code> ; нуль, если <code>str1 == str2</code> ; больше нуля, если <code>str1 > str2</code> |
| <code>char *strcpy(char *str1, char *str2)</code> | копирует содержимое <code>str2</code> в <code>str1</code> и возвращает <code>str1</code> |
| <code>size_t strspn(char *str1, char *str2)</code> | возвращает индекс первого символа в <code>str1</code> , который не совпадает ни с одним из символов в <code>str2</code> |
| <code>size_t strcspn(char *str1, char *str2)</code> | возвращает индекс первого символа в <code>str1</code> , который совпадает с любым из символов в <code>str2</code> |
| <code>char *strpbrk(char *str1, char *str2)</code> | возвращает указатель на первый символ в <code>str1</code> , который совпадает с любым символом в <code>str2</code> |
| <code>char *strtok(char *str1, char *str2)</code> | позволяет разделить <code>str1</code> на лексемы, <code>str2</code> - разделители лексем |

Пример использования функции `strtok()`:

```

1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str[] = "I'm going off the rails on a crazy train";
6      char delim[] = ", ";
7      char *p = strtok(str, delim);
8      printf(p);
9      while(p = strtok(NULL, delim))
10         printf("|%s", p);
11     return 0;
12 }
```

Результаты работы программы:

`I'm|going|off|the|rails|on|a|crazy|train`

Варианты заданий:

1. Сформировать из исходной строки новую строку, удалив из неё все последовательности одинаковых символов, оставив число повторов и один символ. Например: при вводе `"aaabcc"` результат работы программы будет `"3ab2c"`.
2. Обратить порядок букв в каждом слове исходной строки. Например: при вводе `"exit light enter night"` результат работы программы будет `"tixe thgil retne thgin"`.
3. Вывести на экран строку, составленную из последних букв каждого слова исходной строки.
4. Выполнить частотный анализ введённой строки и удалить из неё слово (или слова, если их несколько) с максимальной суммой частот символов.
5. Вывести на экран длиннейшую подстроку исходной строки, содержащую только уникальные символы. Например: при вводе `"theskyisturningredreturntopower"` результат работы программы будет `"kyisturn"`.
6. Найти самое длинное слово-палиндром в строке.
7. Определить, есть ли среди слов введённой строки целые числа и просуммировать их.
8. Вывести на экран слова исходной строки, встречающиеся в ней только один раз. Например: при вводе `"When hunter meets hunter this hunter eats hunter"` результат работы программы будет `"When meets this eats"`.
9. Разместить в новой строке слова исходной строки в порядке убывания их длины.
10. Составить из исходной строки новую, удалив все слова наиболее часто встречающейся длины. Например: при вводе `"Vicariously I live while the whole world"` результат работы программы будет `"Vicariously I live the"`.
11. Определить, есть ли среди слов введённой строки вещественные числа и найти среднее.
12. Удалить из слов исходной строки все повторные вхождения первого символа.