

Лабораторна робота № 9

Використання вказівників для роботи із символьними рядками в програмах на С.

Мета роботи: оволодіння практичними навичками роботи з даними символного типу та рядкового типу, в тому числі зі стандартними та функціями для рядкового типу даних.

Короткі теоретичні відомості

Заголовковий файл `string.h` або `cstring` містить оголошення функцій опрацювання стрічок. При використанні цих функцій можуть виникати попередження щодо використання функцій, тому доречно звертатися до відповідної директиви препроцесора: `#pragma warning(disable :4996)`

`int getch()`; - повертає значення символу (якщо він є), який користувач набрав на клавіатурі, символ на екрані не відображається.

Заголовки - `stdio.h`:

`int getchar ()`; - Повертає значення символу (якщо він є), який користувач набрав на клавіатурі. Після введення символу потрібно натиснути клавішу `Enter`.

`int putchar (int c)`; - Виводить символ `c` на екран. У разі успіху повертає сам символ `c`, в іншому випадку - `EOF`.

`char * gets (char * s)`; - Читає символи, включаючи пробіли та табуляції, до тих пір, поки не зустрінеться символ нового рядка, який замінюється нульовим символом., н.п. `char in [20]`;

`cout << "введіть стрічку";`

`setlocale (0, "C")`;

`char * phar`;

`char = gets (in)`;

`int puts (const char * s)`; - Виводить рядок, задану аргументом `const char * s`.

Ще один спосіб введення стрічки

`char in [70]`;

`cout << "введіть стрічку";`

`cin.getline (in, 70)`;

найпростіше:

`char in [70]`;

`cout << "введіть стрічку";`

`setlocale (0, "C")`;

`char * phar = in`;

`phar = gets (in)`;

`cin >> in`;

Заголовок - `string.h`, функції:

`char * strcat (char * dest, const char * scr)`; - Об'єднує вихідний рядок `scr` і результуючий рядок `dest`, дописуючи `str` до `dest`. Повертає `dest`.

`char * strncat (char * dest, const char * scr, int maxlen)`; - Об'єднує `maxlen` символів початкового рядка `scr` і результуючий рядок `dest`, приєднуючи частину першої до останньої. Повертає `dest`.

char * strchr (const char * s, int c); - Шукає в рядку s перше входження символу c, починаючи з початку рядка. У разі успіху повертає покажчик на знайдений символ, інакше повертає нуль.

char * strrchr (const char * s, int c); - Аналогічно попередньому, тільки пошук здійснюється з кінця рядка.

int strcmp (const char * s1, const char * s2); - Порівнює два рядки. Повертає від'ємне значення, якщо $s1 < s2$; нуль, якщо $s1 == s2$; позитивне значення, якщо $s1 > s2$. Параметри - покажчики на порівнювані рядки.

int stricmp (const char * s1, const char * s2); - Аналогічно попередньому, тільки порівняння здійснюється без урахування регістру символів.

int strncmp (const char * s1, const char * s2, int maxlen); - Аналогічно попередньому, тільки порівнюються перші maxlen символів.

int strnicmp (const char * s1, const char * s2, int maxlen); - Аналогічно попередньому, тільки порівнюються перші maxlen символів без урахування регістру.

int strcspn (const char * s1, const char * s2); - Повертає довжину максимальної початкової підрядка рядка s1, яка не містить символів з другого рядка s2.

int strlen (const char * s); - Повертає довжину рядка s - кількість символів, що передують нульового символу.

char * strlwr (char * s); - Перетворює всі прописні (великі) букви в рядкові (малі) в рядку s.

char *strupr (char * s); - Перетворює всі рядкові (малі) літери в прописні (великі) в рядку s.

char * strnset (char * s, int c, int n); - Заповнює рядок s символами c. Параметр n задає кількість розміщуваних символів в рядку.

char * strpbrk (const char * s1, const char * s2); - Шукає в рядку s1 перше входження будь-якого символу з рядка s2. Повертає покажчик на перший знайдений символ або нуль - якщо символ не знайдено.

char * strrev (char * s); - Чи змінює порядок проходження символів в рядку на зворотний (крім завершального нульового символу). Функція повертає рядок s.

char * strset (char * s, int c); - Замінює всі символи рядка s заданим символом c.

int strspn (const char * s1, const char * s2); - Обчислює довжину максимальної початкової підрядка рядка s1, що містить тільки символи з рядка s2.

char * strstr (const char * s1, const char * s2); - Шукає в рядку s1 рядок s2. Повертає адреса першого символу входження рядка s2. Якщо рядок відсутній - повертає нуль.

char * strtok (char * s1, const char * s2); - Ділить вихідну рядок s1 на лексеми (підрядка), розділені одним або декількома символами з рядка s2.

Заголовок - math.h

double atof (const char * s); - Перетворює рядок s в число з плаваючою точкою типу double.

Заголовок - stdlib.h

int atoi (const char * s); - Перетворює рядок s в число типу int. Повертає значення або нуль, якщо рядок перетворити не можна.

long atol (const char * s); - Перетворює рядок s в число типу long.

char * itoa (int value, char * s, int radix); - Перетворює значення цілого типу value в рядок s. Повертає покажчик на результуючий рядок. Значення radix - основа системи числення, що використовується при перетворенні (від 2 до 36).

Завдання

Скласти програму, яка опрацює введений користувачем текст. Для зберігання тексту використати масиви типу char[] або char*. Виконати завдання згідно варіанту,

використовуючи бібліотечні функції для обробки символьних рядків. Додатково порахувати кількість слів у результуючому тексті. Опрацьований текст та кількість слів в ньому вивести на екран.

Варіанти завдань

1. Користувач вводить рядок, що містить арифметичний вираз, у якому використовуються круглі дужки. Перевірити, чи правильно в ньому стоять дужки.
2. Знайти символ, який входить до введеного рядка тексту S найбільшу кількість разів.
3. Користувач задає число n із проміжку від 0 до 109. Надрукувати його українськими словами (напр., 234 = двісті тридцять чотири).
4. Скласти програму, яка всі цифри, що містяться у введеному рядку тексту, замінює їх квадратами.
5. Скласти програму перетворення введеного рядка A, видаливши в ньому кожний символ '*' і подвоївши кожний символ, відмінний від '*'.
6. Скласти програму видалення з введеного рядка всіх входжень заданої групи (послідовності) символів.
7. Скласти програму, яка за рядком A та символом s буде новий рядок, отриманий заміною кожного символу, що йде за s, заданим символом c.
8. Скласти програму підрахунку найбільшої кількості цифр, що йдуть поспіль у введеному рядку A.
9. Скласти програму підрахунку кількості входжень до рядка A заданої послідовності символів, перелічених у рядку V.
10. Виділити з введеного рядка A найбільший підрядок, перший та останній символи якого збігаються.
11. Замінити всі пари однакових символів введеного рядка, які йдуть поспіль, одним символом. Наприклад, рядок "aabcb" має перетворитися на рядок "abc".
12. Побудувати рядок S із двох введених рядків S1, S2 так, щоб до S входили усі символи S1, які не входять до S2, і всі символи S2, які не входять до S1.
13. Скласти програму виведення на друк усіх цифр, які входять до заданого рядка, та окремо – решту символів, зберігаючи при цьому взаємне розташування символів у кожній із цих двох груп.
14. Перевірити, чи є введений рядок ідентифікатором, натуральним числом, чи ні тим, ні іншим.
15. Скласти підпрограму для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа в системі числення за основою b ($2 < b < 16$).

Приклад

16. Скласти програму, що утворює із заданого тексту масив слів і впорядковує слова в алфавітному порядку

```
#include<iostream>
#include<string.h>
using namespace std;

void main()
{
    //для прикладу стрічка start[] ініціалізована в кодї програми
    //для виконання роботи тут слід запрограмувати ввід тексту
    char start[1000] = "Пристрасті – це вітри, що надимають вітрила
корабля; ";
    strcat(start, "вітер, правда, інколи топить корабель, ");
    strcat(start, "але без нього корабель не міг би пливти.");
    char ** s = new char*[100];
    int n = 0;
    setlocale(0,"ukr");
    const char separator[]=" .,"; //Символи-розділювачі
    char *token=NULL; //Показчик для ф-ції strtok
    token=strtok(start, " .,");

    while (token)
    {
        s[n] = new char(strlen(token)+1);
        strcpy(s[n++],token);
        cout<<token<<"\n";
        token=strtok(0, " .,");
    }

    cout<<"\n\n";
    for(int i = 0; i<n-1;i++)
        for(int j = i+1; j<n;j++)
            if(stricmp(s[i],s[j])>0)
            {
                char tmp[100];
                strcpy(tmp,s[i]);
                strcpy(s[i],s[j]);
                strcpy(s[j],tmp);
            }

    for(int i = 0; i<n;i++)
        cout<<s[i]<<" " <<'\n';
}
```