

Лабораторна робота №2. Обробка символьної інформації

Мета і задачі:

Навчитися створювати та відлагоджувати програми, у яких здійснюється обробка символьної інформації на мові програмування C.

Теоретичні відомості і методичні вказівки

Якщо в програмі треба працювати з рядками, їх оголошують як масиви символів. В такому масиві для кожного символу рядка (правильніше: коду символу) відводиться своя комірка, а останній елемент містить символ кінця рядка – „\0” (код нуля).

Для зберігання кодів символів найбільш добре підходить тип даних, який в літературі називають символьним: **char**.

Символьний масив оголошується та ініціалізується за схемою, загальною для масивів довільного типу, наприклад:

```
/* оголошення символьного масиву на 50 елементів */
char FirstName[50];

// Масив Greeting1 містить коди символів 'H', 'e', ...*/
char Greeting1[] = {"H", "e", "l", "l", "o"};

/* Масив Greeting2 ідентичний за вмістом масиву Greeting1
тому що код літери 'H' - 72, 'e' - 101, ... */
char Greeting2[] = {72, 101, 108, 108, 111};
```

Для зручності компілятори C/C++ дозволяють ініціалізувати символьні масиви не лише звичайним шляхом, продемонстрованим в попередньому прикладі, але й наступним чином:

```
char StreetName[] = "вул. Метробудівська 5-а";
```

Для масиву StreetName компілятор автоматично виділяє пам'ять у розмірі 24 байта (довжина ініціалізуючого рядка плюс один байт – символ завершення рядка).

При вводі рядка з клавіатури за допомогою функції scanf оператор отримання адреси – & – не потрібен, так як ідентифікатор масиву фактично являє собою адресу його початку:

```
char FirstName[20];
...
```

```
scanf ("%s", FirstName);
```

Обробка символічних масивів – рядків в цілому виконується аналогічно обробці масивів іншого типу, однак комплект засобів розробки C/C++ включає заготовочний модуль **string.h**, до складу якого входять функції, спеціалізовані для обробки рядків. Прототипи деяких, найбільш використовуваних функцій цього модуля представлені в таблиці 1.

Таблиця 1

Прототип функції	Опис
void *memcpy (void *dest, const void *src, int n);	Копіювання одного масиву в інший
char *strcpy (char *dest, const char *src)	Копіювання одного рядка в інший
char *strcat (char *dest, const char *src)	Конкатенація (зчеплення) двох рядків
char *strchr (const char *s, int c)	Пошук символу в рядку
int strcmp (const char *s1, const char *s2)	Порівняння рядків
int strlen (const char s)	Обчислення довжини рядка
char *strlwr (char *s)	Перетворення символів рядка у маленькі
char *strrev (char *s)	Інверсія рядка (перестановка всіх символів у зворотному порядку)
char *strset (char *s, int ch)	Заповнення рядка заданим символом
char strstr (const char *s1, const char *s2)	Пошук одного рядка в іншому
char *strupr (char *s)	Перетворення символів рядка у великі

Знак „*” (зірка) в таблиці означає, що аргументом або результатом роботи функції є адреса рядка (так званий покажчик), з яким можна працювати як з ідентифікатором відповідного масиву символів.

Порядок виконання і звітування

1. Створити програму на мові C згідно варіанту використавши середовище програмування Dev-C++ 4.0:
 - за допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів);
 - ім'я файлу повинне мати розширення dat.
2. Відкомпілювати та відлагодити програму.
3. Розробити набір тестів і перевірити роботу програми на них.
4. Відповісти на контрольні запитання.
5. Зробити висновки.

6. Звіт по лабораторній роботі має складатися з титульної сторінки, лістингів програм, висновків по роботі.

Варіанти завдань

Варіант 1.

Написати програму, яка визначає скільки разів у файлі зустрічаються символ введений з клавіатури.

Варіант 2.

Написати програму, яка знаходить слова з файлу, які містять цифрові символи. Вивести знайдені слова на екран монітору.

Варіант 3.

Написати програму, яка визначає кількість слів у файлі, які закінчуються на словосполучення, що задається з клавіатури. Вивести знайдені слова на екран монітору.

Варіант 4.

Написати програму, яка визначає кількість букв, кожного слова з файлу. Вивести на екран слово мінімальної довжини.

Варіант 5.

Написати програму, яка виводить на екран першу та останню букву кожного слова з файлу.

Варіант 6.

Написати програму, яка визначає, скільки разів у тексті з файлу зустрічається кожен з символів.

Варіант 7.

Написати програму, яка визначає кількість букв, кожного слова з файлу. Вивести на екран слово максимальної довжини.

Варіант 8.

Написати програму, яка виводить на екран слова з файлу у стовпчик.

Варіант 9.

Написати програму, яка у файлі всі входження *a* замінює на *b*. Словосполучення *a* та *b* задаються з клавіатури.

Варіант 10.

Написати програму, яка виводить на екран символ, який зустрічається у файлі максимальне число разів.

Підсумок

Навчитися створювати програми, у яких здійснюється обробка символічної інформації на мові програмування C.

Контрольні питання

1. Що виконує функція `islower` ?
2. Що виконує функція `atoll` ?
3. Що виконує функція `gets` ?
4. Що виконує функція `sprintf`?
5. Що виконує функція `strstr` ?
6. Що виконує функція `atoi`?
7. Що виконує функція `atof`?
8. Що виконує функція `putchar` ?
9. Що виконує функція `getchar` ?
10. Що виконує функція `puts`?
11. Що виконує функція `sscanf` ?
12. Що виконує функція `strcat` ?
13. Що виконує функція `strcmp`?

14. Що виконує функція memchr ?
15. Що виконує функція memmove ?
16. Що виконує функція memchr?
17. Що виконує функція memchr ?

Контрольні запитання

1. Скопіюйте стрічку, яка зберігається в масиві s2 в масив s1.
2. Введіть з клавіатури стрічку тексту масив s1 не використовуючи функцію scanf.
3. Визначте довжину стрічки в s1 і виведіть результат.

Джерела інформації

1. http://void.net.ua/The_C_Programming_Language.html.
2. http://publications.gbdirect.co.uk/c_book/.
3. <http://www.scribd.com/doc/16306895/Draft-ANSI-C-Rationale>
4. <http://www.cplusplus.com/doc/tutorial/>