Лабораторна робота № 6.

Тема: "Рядки"

Мета: Вивчення символьних і рядкових змінних і способів їхньої обробки в мові С.

1. Короткі теоретичні відомості

Рядок є масивом символів, останнім з яких є символ з кодом 0 (‘\0’). Рядкова константа зображається у вигляді послідовності будь-яких символів, вставлених у лапки, наприклад:

"*** Це рядок символів у мові С",

причому символ з кодом 0 явно в рядок не включається. Він вноситься у процесі компіляції.

Оскільки зворотня коса риска зображається двома косими, то рядок символів з повним шляхом до файлу myfile.txt має вигляд *"C:\\dir\\subdir\\myfile.txt*".

Дві рядкові константи, між яким немає інших розділювачів, крім узагальнених прогалинних символів (прогалина, табуляція, кінець рядка і т.д.), сприймаються, як одна рядкова константа. Таким чином,

"Думи мої, " "думи мої"

"Лихо мені з вами"

сприймається як одна константа:

" Думи мої, думи мої Лихо мені з вами "

Опис змінної, що містить рядок символів, в С-програмах може бути виконаний в одній із наступних форм:

char ім'я рядка [максимальний розмір] = “початкове значення”;

char ім'я рядка [ ] = “початкове значення”;

char * ім'я рядка = "початкове значення".

У першій формі присвоєння початкового значення не є обов'язковим, а у двох інших воно необхідне.

При використанні першої форми слід передбачити позицію для нульового символу. Наприклад, опис *char str* [5] = "Мама" є вірним, а опис *char str* [4] = "Тато" містить помилку.

Основні функції обробки рядків символів

Для роботи з рядками в мові передбачений ряд функцій, більшість з яких визначені у файлах включення *string.h, stdio.h і stdlib.h.*

int strlen(s) – повернення фактичної довжини рядка;

gets(s) – ввід рядка s з клавіатури;

puts(s) – вивід рядка s на екран;

strcat(s1,s2) – приєднання рядка *s2* до рядка *s1;*

strchr(s, ch) – визначення адреси першого входження символу *ch* в рядок *s;*

strstr(s1,s2) – визначення адреси першого входження підрядка *s2* в рядок *s1;*

strcmp(s1,s2) – порівняння рядків *s1* і *s2* (при збігу повертає 0);

strcpy(s1,s2) – копіювання рядка *s2* в рядок *s1;*

atoi(s), – перетворення рядка *s* в ціле число.

atof(s) – перетворення рядка *s* в дійсне число.

При використанні рядка в якості параметра функції в останню досить передати тільки адресу початку рядка, оскільки його довжина однозначно визначається положенням кінцевого нульового символу.

При маніпулюванні з рядками система не контролює вихід фактичної довжини рядка за межі максимально допустимого розміру, що задається при описі. У разі помилки додаткові символи будуть записані поверх інформації, яка розташовується услід за полем, відведеним для рядка. Останнє може стати причиною порушення нормальної роботи програми, тому рекомендується стежити за виходом рядка за передбачені межі, вводячи достатні розміри рядків при їх описі.

Приклад.

```
//Для рядка знайти кількість слів, що закінчуються і починаються на ту ж букву
#include <stdio.h>
#include <string.h>
int Kilslov( char str[])
{
   int k = 0;
   char *token; // тимчасовий буфер

   // розбиваємо стрічку на підстрічки функцією strtok
    token = strtok(str, " ");
```

```
   if(token[0] == token[strlen(token) - 1]) { // якщо перший і останній символи в слові однакові
k++
      k++;
   }

   // поки не розібрали цілу стрічку
   while (token != NULL) {
      // розбиваємо стрічку на підстрічки функцією strtok
      token = strtok(NULL, " ");
      if(token != NULL && token[0] == token[strlen(token) - 1]) { // якщо перший і останній
символи в слові однакові cout++
      k++;
        }
   }
   return k;
}

int main()
{

   int k=0;
   char s[256] = "anapa sos gfsd gdhs ss hgf fgf"; // текстова стрічка
   k = Kilslov(s);
   printf("Kilkist slov: %i", k);
   return 0;
}
```

2. Постановка завдання

Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

3. Варіанти

1. Надрукувати найдовше й найкоротше слово в цьому рядку.
2. Надрукувати всі слова, які не містять голосних букв.
3. Надрукувати всі слова, які містять по одній цифрі.
4. Надрукувати всі слова, які співпадають з її першим словом.
5. Перетворити рядок таким чином, щоб спочатку в ньому були надруковані тільки букви, а потім тільки цифри, не міняючи порядку проходження символів у рядку.

6. Перетворити рядок так, щоб всі букви в ньому були відсортовані за зростанням.
7. Перетворити рядок так, щоб всі цифри в ньому були відсортовані за спаданням.
8. Перетворити рядок так, щоб всі слова в ньому стали ідентифікаторами, слова які складаються тільки із цифр - знищити.
9. Надрукувати всі слова-паліндроми, які є в цьому рядку(див 1 варіант).
10. Перетворити рядок таким чином, щоб на його початку були записані слова, що містять тільки цифри, потім слова, що містять тільки букви, а потім слова, які містять і букви і цифри.
11. Перетворити рядок таким чином, щоб всі слова в ньому були надруковані навпаки.
12. Перетворити рядок таким чином, щоб букви кожного слова в ньому були відсортовані за зростанням.
13. Перетворити рядок таким чином, щоб цифри кожного слова в ньому були відсортовані за спаданням.
14. Перетворити рядок таким чином, щоб у ньому залишилися тільки слова, що містять букви й цифри, інші слова знищити.
15. Визначити яке слово зустрічається в рядку найчастіше.
16. Визначити які слова зустрічаються в рядку по одному разі.
17. Всі слова рядка, які починаються з букви, відсортувати за абеткою.
18. Всі слова рядка, які починаються із цифри відсортувати за спаданням.
19. Знищити з рядка всі слова, які не є ідентифікаторами.
20. Знайти для рядка довжину найдовшого слова.
21. Знищити всі парні слова у речені.
22. Для рядка знайти кількість слів, що починаються на букву «а».
23. Для рядка знайти довжину найкоротшого слова.
24. Для рядка знайти кількість слів у ньому.

4. Методичні вказівки

При програмній реалізації можна скористатись наступним фрагментом програми.

Завдання.

Перевірити чи є рядок паліндромом. (Паліндром - це вираз, що читається однаково зліва направо і зправа наліво).

```
#include <stdio.h>
#define n 100
_Bool checkpolindrom( char s[n])
{
int m=strlen(s);
int k=m/2;
int i;
_Bool p=1;
for(i=0; i<k; i++)
   if(s[i]!=s[m-i-1])
   p*=0;
return p;
}
int main()
{
   char s1[n];
   puts("Input text: ");
   gets(s1);
   puts("Output text: ");
   puts(s1);
   printf("%i\n", strlen(s1));
   int m=strlen(s1);
   int p;
   p=checkpolindrom(s1);
      if(p==0)
         printf("The text isn't polindrom\n");
      else
         printf("The text is polindrom\n");
   printf("Hello world!\n");
   return 0;}
```

5. Зміст звіту

1. Постановка завдання для конкретного варіанту.

2. Початкові дані.

3. Текст програми.

4. Результати виконання програми.