ЛАБОРАТОРНА РОБОТА

Технології та методології обробки текстових рядків

Мета роботи

- Вивчити можливості оголошення та ініціалізації рядків.
- Навчитись вводити рядки з клавіатури і виводити їх на дисплей.
- Дослідити основні функції роботи з рядками та реалізовувати найпростіші операції з текстовими рядками.
- Навчитись використовувати основні функції бібліотек мови C/C++ для роботи з рядками (масивами символів).

Теоретичні відомості

Основні функції роботи з символами та рядками

Рядок у мовах програмування – це скінчена послідовність символів, які сприймаються і обробляються як єдине ціле. Рядки можуть представлятися у вигляді масиву символів, але масив символів - не завжди рядок!

Рядковий тип (*тип рядка*) – це тип даних, значеннями якого є довільна послідовність символів алфавіту. Кожна змінна такого типу може бути представлена фіксованою кількістю байтів або мати довільну довжину.

Рядком в C/C++ вважається символьний масив, який завершується символом кінця рядка '\0' (NULL-символом).

В сучасних IDE, що підтримують мову C++, доступною для роботи є так звана *стандартна бібліотека шаблонів* STL (*Standard Template Library*), яка містить універсальні шаблонні класи і функції, які реалізують функціональність багатьох структур даних. Зокрема, в ній є клас string, що служить для організації роботи з рядками. Сам STL-рядок розглядається як контейнер для зберігання символів. Для використання класу string необхідно підключити заголовний файл <string>.

У C/C++ змінна-рядок, як масив символів, має наступний синтаксис оголошення:

char ім'я[загальна_довжина]

рядок, також, можна оголосити і як покажчик на тип char:

*char *ім'я*

Стандартна бібліотека мови C володіє багатим і різноманітним набором функцій для обробки символів (табл. 1) і рядків (табл. 2).

Для уведення-виведення рядків з консолі у мові C/C++ можуть використовуватися декілька альтернативних варіантів:

- функції форматованого введення-виведення *(scanf, printf);*
- спеціалізовані функції введення-виведення рядків *(gets, puts).*

Мова С++ розширює ці можливості шляхом використання потоків введення-виведення *cin і cout.*

Примітка: у С/С++ введення рядків за допомогою функцій форматованого і операцій потокового введення-виведення ***має важливе обмеження:*** такі рядки вводяться тільки до першого роздільного символу (символу пробілу, табуляції або нового рядка), всі інші символи рядка - ігноруються. Тому ці засоби можна застосовувати тільки для введення окремих слів.

Для введення рядків-речень можна використати спеціалізовану функцію введення рядків gets. Формат її виклику є таким:

```
gets(рядок);        // читання введеного з клавіатури рядка
```

При використанні функції gets, у випадку помилки введення, повертається нульовий покажчик NULL (описаний у бібліотеці stdio.h). Тому для перевірки правильності введення рядків доцільно икористовувати цикл:

```
while (gets(s)!= NULL)
{  …
}
```

У С++ для введення рядків-речень можна також скористатися функцією getline, вбудованою у потік введення cin. Функція getline має два параметри: перший аргумент – рядок, який вводиться, другий – кількість символів. Наприклад,

```
    char  s[36];
    cout<<"Enter row: ";
cin.getline( s, 30); // читання введеного з клавіатури рядка-речення
```

Рядки можуть оброблятися як цілісний об’єкт, а також поелементно (посимвольно). При *посимвольній* обробці доступ до конкретного символу рядка, як і до елементів масиву символів, у С/С++ здійснюється за індексом або за покажчиком. При посимвольній обробці до окремого символа рядка можна застосовувати ті ж операції, що і до змінної символьного типу.

Рядки, *як цілісні об’єкти*, можна присвоювати, порівнювати і обробляти різноманітним чином, застосовуючи вбудовані засоби (підпрограми) обробки рядків, які, зазвичай, містяться у стандартних бібліотеках відповідної системи програмування.

Дії по обробці рядків різняться в залежності від мови програмування. Однак, усі їх можна згрупувати наступним чином:

- ініціалізація рядка;
- визначення довжини рядка;
- копіювання рядка;
- об'єднання (конкатенація) рядків;
- порівняння рядків;
- аналіз символів рядка;
- пошук у рядку;
- перетворення рядка.

Таблиця 1 - Функції для роботи з символами рядка (<ctype.h>)

Прототип функції	Опис
`int` **`isalnum`**`(int c);`	Перевірка, чи є символ літерою або цифрою
`int` **`isalpha`**`(int c);`	Перевірка, чи є символ літерою
`int` **`iscntrl`**`(int c);`	Перевірка, чи є символ керуючим
`int` **`isdigit`**`(int c);`	Перевірка, чи є символ десятковою цифрою
`int` **`isgraph`**`(int c);`	Перевірка, чи є символ видимим
`int` **`isprint`**`(int c);`	Перевірка, чи є символ видимим, включаючи пробіл
`int` **`islower`**`(int c);`	Перевірка, чи є символ літерою нижнього регістру
`int` **`ispunct`**`(int c);`	Перевірка, чи є символ знаком пунктуації
`int` **`isspace`**`(int c);`	Перевірка, чи є символ пробільним
`int` **`isupper`**`(int c);`	Перевірка, чи є символ літерою верхнього регістру
`int` **`isxdigit`**`(int c);`	Перевірка, чи є символ шістнадцятковою цифрою
`int` **`tolower`**`(int c);`	Перетворення символу в нижній регістр
`int` **`toupper`**`(int c);`	Перетворення символу у верхній регістр

Таблиця 2 - Функції бібліотеки введення/виведення *(<stdio.h>)*

Прототип функції	Опис
`int` **`getchar`** `(void)`	Вводить наступний символ зі стандартного при - строю введення і повертає його в форматі цілого.
`char` ***`gets`**`(char *s)`	Уводить символи зі стандартного пристрою введення в масив s до тих пір, поки не зустріне символ кінця рядка або індикатор кінця файлу.
`int` **`putchar`**`(int c)`	Виводить символ, який зберігається в с.
int puts(const char *s)	Виводить рядок s з переходом на наступний рядок
int: sprintf (char *buf, const char *format [,arg1, ...])	Виконує форматоване виведення у рядок buf. Параметр format задає спосіб відображення значень змінних argl, ... Дія функції sprintf аналогічна дії функції printf, але виведення виконується в рядок - буфер, а не на екран.
int sscanf (const char *s, const char *format [, addressl, ...])	Виконує дії, еквівалентні scanf за винятком того, що уведення здійснюється з масиву s, а не з клавіатури.

Рядкові функції працюють з масивами символів (рядками), що закінчуються символом кінця рядка. У мові C/C++ для роботи з рядковими функціями використовується заголовок <*string.h*>. У ньому визначено тип *size_t* - тип результату, який утворюється після застосування оператора *sizeof()* і являє собою різновид цілого без знака. У таблиці 3 наведено найбільш популярні і підтримувані більшістю компіляторів функції

Таблиця 3 - Функції роботи з рядками *(<string.h>)*.

Прототип функції	Опис
`char *strcpy (char *s1, const char *s2)`	Копіює рядок s2 в масив sl. Повертає значення sl.
`char *strncpy(char *s1, const char *s2, size t n)`	Копіює не більше, ніж n символів рядка s2 в масив sl. Повертає значення sl.
`char *strcat(char *s1, const char *s2)`	Об'єднує рядок s2 з рядком масива sl. Перший символ рядка s2 переписує символ NULL рядка sl. Повертає значення sl.
`char *strncat(char *s1, const char *s2, size_t n)`	Об'єднує не більше, ніж n символів рядка s2 з рядком s l . Перший символ рядка s2 переписує символ NULL рядка sl. Повертає значення sl.
`int strcmp (const char *s1, const char *s2)`	Порівнює рядок sl з рядком s2. Функція повертає 0, значення менше 0 або більше 0, якщо sl рівна, менше або більше, ніж s2.
`int strncmp (const char *s1, const char *s2, size_t n)`	Порівнює n символів рядків s1 і s2. Функція повертає 0 , з начення, менше 0 або більше 0, якщо sl відповідно рівний, менший або більший, ніж s2.
`char *strchr(const char *s, int c)`	Знаходить позицію першого входження символа с в рядок s. Якщо с знайдено, функція повертає покажчик на с в рядку s. Інакше повертається покажчик NULL.
`char *strrchr(const char *s, int c)`	Знаходить позицію останнього входження символу с в рядок s. Якщо с знайдено, то повертається покажчик на с в рядку s. Інакше повертається покажчик NULL.
`size t strcspn(const char *s1, const char *s2)`	Повертає довжину початкового сегмента рядка sl, що містить тільки ті символи, що не входять в s2.
`size t strspn(const char *s1, const char *s2)`	Визначає і повертає довжину початкового сегмента рядка sl, що містить тільки ті символи, що входять в s2.
char *strstr(const char *s1, const char *s2)	Знаходить позицію першого входження рядка s2 в рядок sl. Якщо знайдено, повертається покажчик підрядка в рядку sl. Інакше повертається покажчик NULL.

char *strtok (const char *s1, const char *s2)	Послідовні виклики функції виконують розбиття рядка sl на лексеми (слова), розділені символами, які містяться в s2. При першому виклику функція отримує в якості аргументу рядок sl, а при наступних викликах, щоб продовжити розбиття того ж рядка, першим аргументом передається NULL. При кожному виклику повертається покажчик на поточну лексему рядка sl. Якщо лексем в рядку не залишилось, то повертається NULL.
size_t strlen(const char *s)	Визначає довжину рядка s (до символу NULL).

Оскільки в C/C++ не передбачений автоматичний контроль порушення меж масивів, вся відповідальність за їх переповнення лягає на програміста.

Детально приклади обробки рядків розглянуті у відповідній лекції у розділі «Обробка рядків».

Приклад розробки програми для роботи з рядками і символами

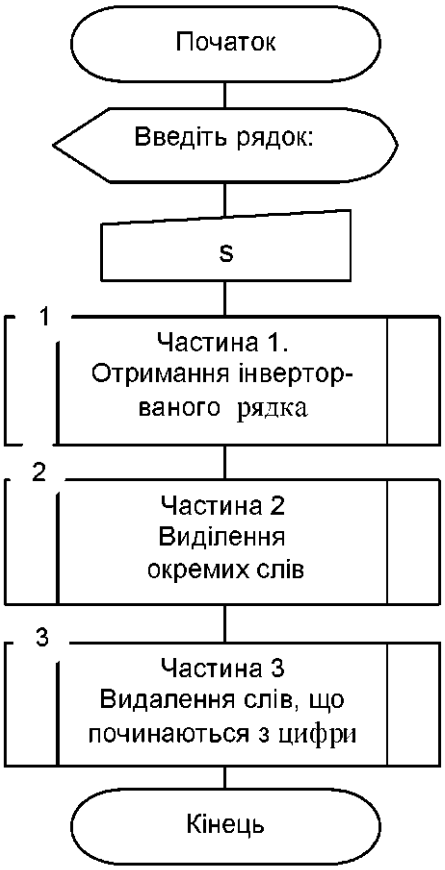
Задача. З клавіатури вводиться текстовий рядок. Скласти програму, яка:
а) інвертує рядок, подаючи його у зворотному вигляді;
б) підраховує кількість чисел у тексті;
в) видаляє всі слова, що починаються з цифри.

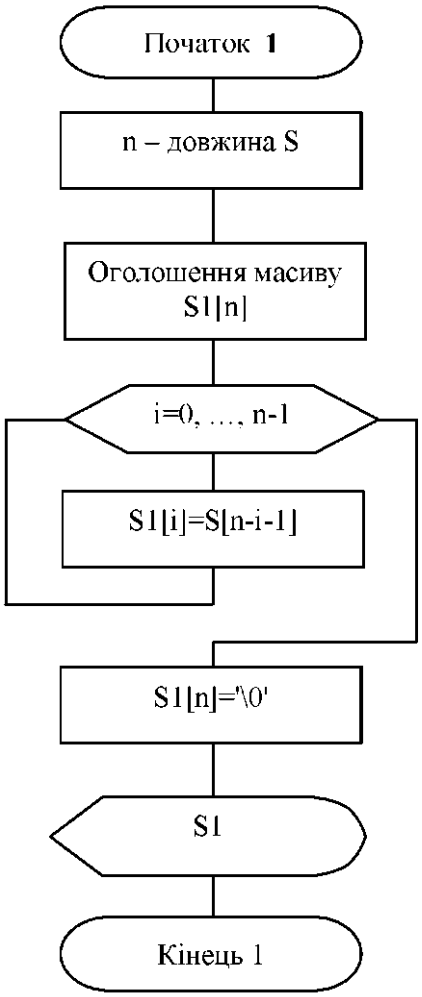
Формалізація задачі
1. Вхідний рядок S представимо у вигляді масиву символів.
2. Підрахуємо n - кількість символів у рядку.
3. Далі програму розділимо на три послідовних частини.
   a) Отримання інвертованого рядка:
   - підготуємо рядок S l такого самого розміру n;
   - здійснимо в циклі (i=0, ..., n-l) присвоєння:
   - $S1_i = S_{n-i-1}$ (тут враховуємо те, що останній символ рядка S дорівнює '\0');
   - додаємо в массив S1 символ закінчення (щоб обірвати рядок).
   b) Виділення окремих слів.
   - запам'ятовуємо S в S1 (щоб не зіпсувати початковий рядок);
   - переглядаємо рядок S по словах (виділяємо частину рядка до пробілу). Це доцільно зробити в циклі.
   c) Видалення слів, що починаються з цифри:
   - відновлюємо S з S1;
   - готуємо рядок S2, в якому будемо зберігати ті слова, що не починаються з цифри;
   - переглядаємо рядок S по словах (виділяємо частину рядка до пробілу). Якщо слово починається з цифри, не додаємо його до S2, в іншому випадку - додаємо;
   - додаємо в массив S2 символ закінчення (щоб обірвати рядок).

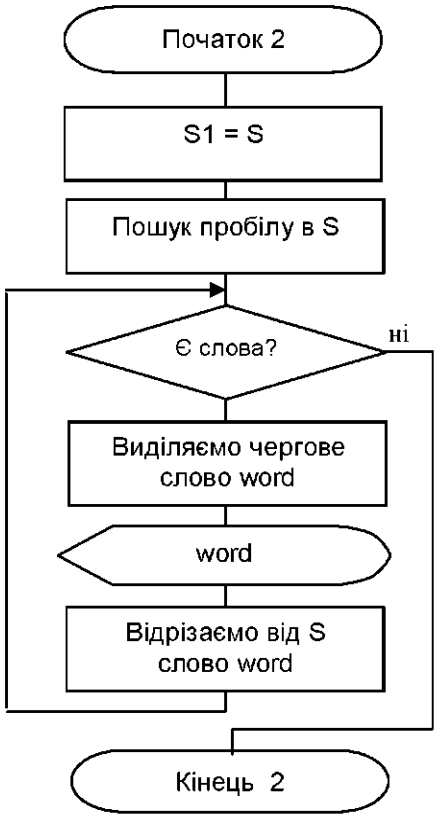
Схема роботи програми

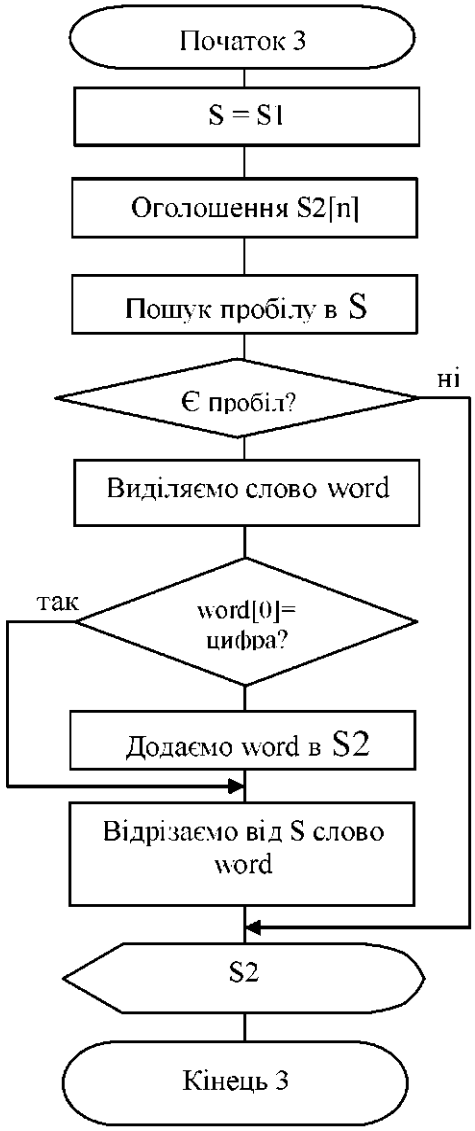
Розробимо загальну схему роботи програми.

Для кожної з частин доцільно розробити окрему схему


Початок
Введіть рядок:
S
1
Частина 1.
Отримання інвертор-
ваного рядка
2
Частина 2
Виділення
окремих слів
3
Частина 3
Видалення слів, що
починаються з цифри
Кінець


Початок 1
n – довжина S
Оголошення масиву
S1[n]
i=0, …, n-1
S1[i]=S[n-i-1]
S1[n]='\0'
S1
Кінець 1


Початок 2
S1 = S
Пошук пробілу в S
Є слова?
ні
Виділяємо чергове
слово word
word
Відрізаємо від S
слово word
Кінець 2


Початок 3
S = S1
Оголошення S2[n]
Пошук пробілу в S
Є пробіл?
ні
Виділяємо слово word
так
word[0]=
цифра?
Додаємо word в S2
Відрізаємо від S слово
word
S2
Кінець 3

Лістинг програми

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
  cout << "\n Input String:  ";
  char s[80];
  gets(s);
  int n=strlen(s);// довжина рядка
  char *s1 = new char[n];
  for (int i=0; i<n; i++)// перевертаємо рядок
       s1[i]=s[strlen(s)-i-1];
  s1[n]='\0';
  cout << "\n Reverse String:  " << s1;
            // виділяємо слова у реченні
  strcpy(s1,s);   // запамятовуємо початкове речення
  cout <<"\n\n\n Words in string:  \n";
  char *word;
  word = strtok(s, " " );
  while (word != NULL)
  {
       printf("\n\t%s",word);
       word=strtok(NULL, " ");
  }
  // ще раз виділяємо слова, але не виводимо ті, що починаються з цифри
  strcpy(s,s1);   // відновлюємо початкове речення
  cout <<"\n\n\n String without first digits:  \n";
  char *s2 = new char[n];
  strcpy(s2,"");
  word = strtok(s, " " );
  while (word != NULL)
  {
       if (strpbrk(word,"0123456789")==NULL)
            strcat(s2,word);
       strcat(s2," ");
       word=strtok(NULL, " ");
  }
  strcat(s2,"\0");
  printf("\n\t%s",s2);
  delete[] s1;
  delete[] s2;
  _getch();
  return 0;
}
```

Результат роботи програми

```
Input String:  HELO: aaaaaa 2bbbbbbb ccccccc 3ddddddd yy!
Reverse String:  !yy ddddddd3 ccccccc bbbbbbb2 aaaaaa :OLEH

Words in string:

        HELO:
        aaaaaa
        2bbbbbbb
        ccccccc
        3ddddddd
        yy!

String without first digits:

        HELO: aaaaaa  ccccccc  yy!
```

Порядок виконання роботи

1. Ознайомитись з теоретичним матеріалом, з функціями стандартних бібліотек для роботи з рядками і символами.
2. Дослідити процес реалізації завдань прикладів, відлагодити наведену програму на своєму комп'ютері.
3. Розробити власну програму, яка реалізує індивідуальне завдання.
4. Підготувати звіт, який включатиме:
   - варіант і текст індивідуального завдання;
   - формалізацію завдання (включаючи математичну модель задачі);
   - схему роботи програми;
   - лістинг програми;
   - роздруківку трьох контрольних прикладів (вигляд екрану з результатами - різні значення вхідних даних);
   - висновки.

Варіанти індивідуальних завдань

Задача 1. З клавіатури вводиться текстовий рядок. Розробити програму, яка реалізує вказані дії.

1	а) підраховує кількість слів, які мають непарну довжину; б) виводить на екран частоту входження кожної літери; в) видаляє текст, що розміщено в круглих дужках.
2	а) перевіряє, чи співпадає кількість відкритих і закритих дужок у введеному рядку (перевірити для круглих та квадратних дужок); б) виводить на екран найдовше слово; в) видаляє всі слова, що складаються тільки з латинських літер.
3	а) підраховує кількість різних слів, що входять до заданого тексту; б) виводить на екран кількість використаних символів; в) видаляє всі слова, що мають подвоєні літери.
4	а) підраховує кількість слів у тексті; б) виводить на екран слово, що містить найбільшу кількість голосних літер; в) видаляє з тексту всі непотрібні пробіли.
5	а) підраховує кількість розділових знаків у тексті; б) виводить всі слова, що мають парну кількість літер; в) міняє місцями першу і останню літери кожного слова.
6	а) підраховує кількість великих літер у тексті; б) виводить на екран слова, що мають найменшу кількість літер; в) видаляє всі слова, що починаються з малої літери.
7	а) підраховує кількість чисел у тексті (не цифр, а саме чисел); б) виводить на екран всі слова, що складаються тільки з латинських літер; в) видаляє кожне друге слово.

8	а) підраховує кількість цифр у тексті; б) виводить на екран слова, що починаються з приголосних літер; в) знищує всі слова, які починаються і закінчуються за одну й ту ж літеру.
9	а) підраховує кількість слів у тексті, які закінчуються на голосну літеру; б) виводить на екран всі слова, довжина яких менша п'яти символів; в) видаляє всі слова, які містять хоча б одну латинську літеру.
10	а) підраховує кількість слів у тексті, які починаються з голосної літери; б) виводить на екран всі слова, що мають непарну кількість приголосних літер; в) видаляє всі числа з тексту.
11	а) замінює всі великі літери, що входять до тексту на відповідні малі; б) виводить на екран найдовше слово; в) видаляє всі слова, що містять непарну кількість приголосних літер.
12	а) кількість слів, які містять однакову кількість голосних і приголосних літер; б) виводить на екран найдовше слово; в) видаляє з тексту всі слова-паліндроми.
13	а) виводить всі символи, які розташовані після першого символу б) підраховує кількість речень, що містять непарну кількість слів; в) видаляє з тексту всі слова, які розташовані після ком.
14	а) підраховує кількість слів у кожному реченні; б) виводить на екран найдовше речення; в) видаляє всі слова, передостання літера яких голосна.
15	а) інвертує рядок, подаючи його у зворотному вигляді; б) підраховує кількість чисел у тексті; в) видаляє всі слова, що починаються з голосних літер.

Додаткове самостійне завдання для

**** Задача 2.*** Зловмисник не викраде інформацію, про існування якої він не підозрює - ось гасло стеганографії. Карл - геть не зловмисник, однак він найближчими днями святкуватиме свій день народження, тому Аліса і Боб повинні обговорити подарунок, який вони вирішили разом презентувати Карлу.

На шляху їх благородних бажань стала одна перешкода - Боба поселили в гуртожитку в одній кімнаті з Карлом. Переймаючись тим, що Карл може ненароком побачити, про що переписуються Боб з Алісою, Боб вирішив скористатись здобутками стеганографії, приховуючи свої справжні повідомлення, керуючись:

- кожне *k-те* повторення голосної та кожне l-те повторення приголосної літери кирилиці становлять літери прихованого повідомлення;

- кожна зустріч латинської *літери* l, вставленої в тексті замість аналогічної літери кирилиці, позначає, що до приблизної вартості подарунку необхідно додати десять гривень, а кожна латинська *літера* 2 -одну гривню.

Напишіть програму, яка допоможе Алісі читати приховані повідомлення Боба.

Варіант	*l*	*k*	*літера* 1	*літера* 2
1	4	6	a	
2	3	8	x	
3	5	7	i	
4	2	5	y	
5	4	4	a	
6	3	6	x	
7	2	8	i	
8	2	7	y	i

Варіант	*l*	*k*	*літера* 1	*літера* 2
9	4	5	a	e
10	3	4	x	a
11	5	6	i	x
12	2	8	y	o
13	4	7	a	e
14	3	5	x	i
15	2	4	i	a

Контрольні питання

1. Який заголовний файл використовується у мові С для роботи з символами?
2. Наведіть приклади використання функцій обробки символів.
3. Які функції існують для введення і виведення символів?
4. Наведіть основі функції уведення\виведення рядків і їх призначення. Який заголовний файл містить опис цих функцій?
5. Чим відрізняються різні функції уведення\виведення рядків? Наведіть приклади їх використання.
6. Який заголовний файл використовується у мові С для обробки рядків?
7. Назвіть основні функції роботи з рядками.
8. Як можна оголосити рядок? Наведіть різні способи оголошення та ініціалізації рядків.
9. Для чого у рядках використовується завершуючий символ?
10. Як отримати довжину рядка?
11. Як компілятор мови С контролює вихід за межі масиву символів?
12. Як можна запрограмувати виділення слів у введенному реченні?
13. Яким чином можна об'єднати рядки символів та виділити підрядки за певними ознаками? Наведіть приклади.