Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп’ютерної інженерії
Кафедра ПЗ

Лабораторна робота №6, варіант № 19
з дисципліни «Основи програмування»

Виконала: ст. 1ПІ-25Б Семенов В.О. .
Перевірив: доцент Решетнік О.О.

Вінниця – 2025

Тема: Робота з рядками

Мета: Опанувати на практиці операції з рядками та використання бібліотечних функцій для обробки рядків.

Завдання 19. Реалізувати функції: int stringCompareCS(char *str1, char *str2) – порівнює дві строки з урахуванням регістру літер і повертає 0 – якщо вони рівні, -1 – якщо str1 лексикографічно перша за str2, 1 – якщо str2 лексикографічно перша за str1. int stringCompareCIS(char *str1, char *str2) – порівнює дві строки без урахування регістру літер і повертає 0 – якщо вони рівні, -1 – якщо str1 лексикографічно перша за str2, 1 – якщо str2 лексикографічно перша за str1.

Декомпозиція завдання

1. Оголошення та підключення бібліотек

- Підключаються бібліотеки для вводу/виводу, роботи з символами та рядками.

2. Реалізація функцій порівняння рядків

- stringCompareCS – порівняння з урахуванням регістру.
- stringCompareCIS – порівняння без урахування регістру.

3. Функція введення рядка користувачем

- Забезпечує введення непорожніх рядків.
- Повторно запрошує введення, якщо рядок порожній.

4. Функція виведення результату порівняння

- Перетворює числовий результат (-1, 0, 1) у текстовий опис.

5. Демонстрація роботи з тестовими рядками

- Викликаються обидві функції порівняння для попередньо заданих рядків.
- Показуються всі варіанти результатів (-1, 0, 1).

6. Введення рядків користувачем і їх порівняння

- Користувач вводить два рядки.

- Виконується порівняння case-sensitive та case-insensitive.
- Виводиться результат у зрозумілому вигляді.

7. Відображення результатів

- Для кожного виклику порівняння показується текстовий висновок про відносини рядків.

Код програми показано на лістингу.

Приклади роботи програми можна побачити на рисунку 1

Лістинг 1 – Програма для знаходження рядка з найменшим добутком

```
#include <iostream>
#include <cctype>
#include <cstring>

int stringCompareCS(char* str1, char* str2) {
    while (*str1 && *str2) {
        if (*str1 < *str2) return -1;
        if (*str1 > *str2) return 1;
        str1++;
        str2++;
    }
    if (*str1 == *str2) return 0;
    return (*str1) ? 1 : -1;
}

int stringCompareCIS(char* str1, char* str2) {
    while (*str1 && *str2) {
        char c1 = std::tolower(*str1);
        char c2 = std::tolower(*str2);
        if (c1 < c2) return -1;
        if (c1 > c2) return 1;
        str1++;
        str2++;
    }
    if (*str1 == *str2) return 0;
    return (*str1) ? 1 : -1;
}

void inputString(char* str, int maxLength, const std::string&
prompt) {
    do {
        std::cout << prompt;
        std::cin.getline(str, maxLength);
        if (std::strlen(str) == 0) {
```

Продовження лістингу 1

```
            std::cout << "Invalid input. Please enter a non-
empty string.\n";
        }
    } while (std::strlen(str) == 0);
}

void printResult(int res) {
    if (res == 0) std::cout << "Result: 0 (strings are
equal)\n";
    else if (res == -1) std::cout << "Result:-1 (first string is
lexicographically less)\n";
    else std::cout << "Result:1 (first string is
lexicographically greater)\n";
}

int main() {
    const int MAX_LEN = 100;
    char user1[MAX_LEN];
    char user2[MAX_LEN];
    char test1[] = "Apple";
    char test2[] = "apple";
    char test3[] = "Banana";
    char test4[] = "Apple";
    char test5[] = "Apples";

    std::cout << "=== Case-sensitive comparisons with test
strings ===\n";
    printResult(stringCompareCS(test1, test2));
    printResult(stringCompareCS(test1, test4));
    printResult(stringCompareCS(test1, test3));
    printResult(stringCompareCS(test5, test4));

    std::cout << "\n=== Case-insensitive comparisons with test
strings ===\n";
    printResult(stringCompareCIS(test1, test2));
    printResult(stringCompareCIS(test1, test4));
    printResult(stringCompareCIS(test1, test3));
    printResult(stringCompareCIS(test5, test4));

    std::cout << "\n=== Now you can test your own strings
===\n";
    inputString(user1, MAX_LEN, "Enter first string: ");
    inputString(user2, MAX_LEN, "Enter second string: ");

    std::cout << "\nCase-sensitive comparison:\n";
    printResult(stringCompareCS(user1, user2));

    std::cout << "Case-insensitive comparison:\n";
    printResult(stringCompareCIS(user1, user2));
    return 0;
}
```

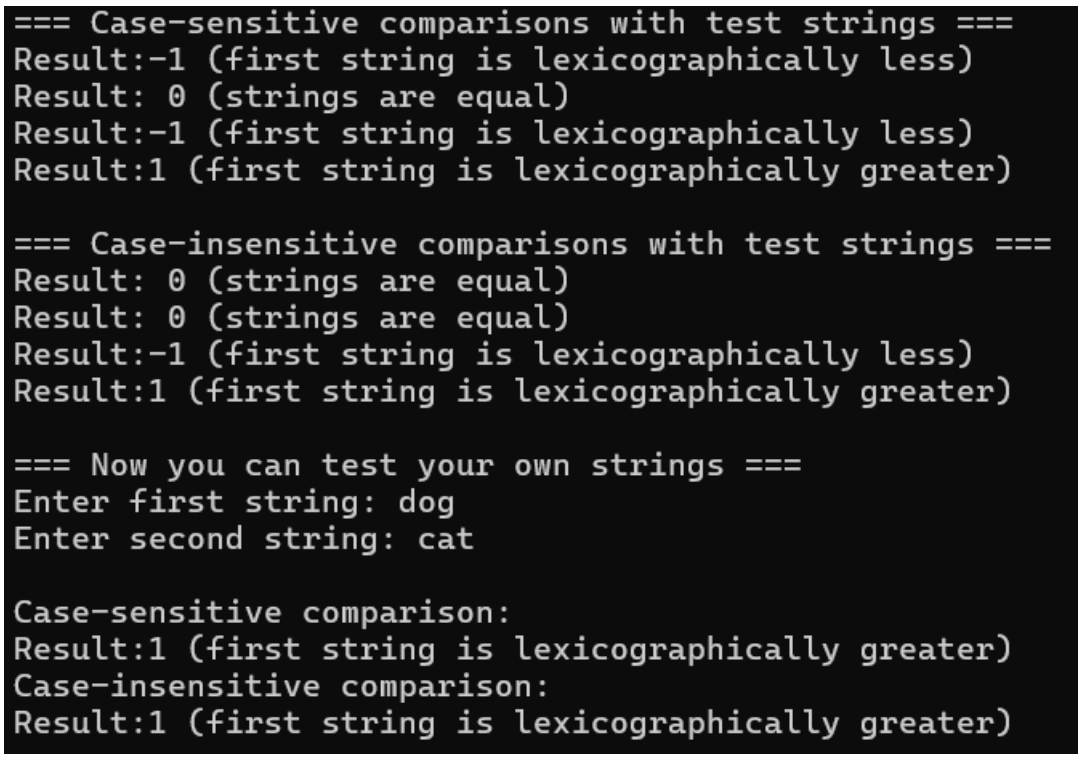

Рисунок 1 – робота програми при правильному введені

Контрольні запитання:

19. Операція взяття адреси

- Повертає адресу змінної в пам’яті (оператор &).

20. Константний покажчик

- Покажчик, який завжди вказує на один і той же об’єкт, але значення об’єкта можна змінювати.

21. Арифметичні дії з покажчиками

- Додавання/віднімання чисел – пересування по масиву.
- Віднімання одного покажчика від іншого – обчислення відстані між елементами.
- Порівняння покажчиків – перевірка порядку або рівності адрес.
- Використовуються для роботи з масивами, динамічною пам’яттю та обходу елементів.

Висновок: У процесі роботи з рядками ми навчились порівнювати рядки з урахуванням регістру і без нього, використовувати бібліотечні функції для обробки символів, а також забезпечувати коректний ввід рядків

користувачем. Це дає базу для ефективної обробки текстових даних у програмах.