

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 9
з дисципліни «Алгоритмізація та програмування»
на тему «Робота з рядками на C ++»

XAI.301.174.312.8

Виконав студент гр. 312

_____ Діхтяренко М.Г.
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ роботи з низькорівневими рядками на C++ і документацію до класу `string`, а також алгоритми пошуку в рядку, а також реалізувати обробку рядків на C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1.

А. Вивчити по документації метод стандартного класу `size_t find (const char* s, size_t pos, size_t n) const`;

В. Визначити функцію, що виконує ті ж дії, що і вивчений метод класу `string`. Вихідний рядок передати першим параметром (масив символів). Для реалізації методу не використовувати функції обробки рядків зі стандартних бібліотек.

С. Викликати свій метод і метод `string size_t find (const char* s, size_t pos, size_t n) const`;

Завдання 2.

А. Описати функцію, що обробляє рядок. Дано рядки `S` і `S0`. Знайти кількість входжень рядка `S0` в рядок `S`. Для реалізації можна використовувати функції обробки рядків зі стандартних бібліотек.

В. Описати функцію, яка перевіряє, чи задовольняє рядок умовам завдання.

С.* Створити вихідний текстовий файл, що містить не менше 10 різних рядків. D. Використовуючи функції 2.А і 2.В, обробити рядок / * текстовий файл рядок за рядком. Додаткові дані ввести з консолі.

Е. Отриманий результат записати у вихідний файл.

ВИКОНАННЯ РОБОТИ

Завдання 1.

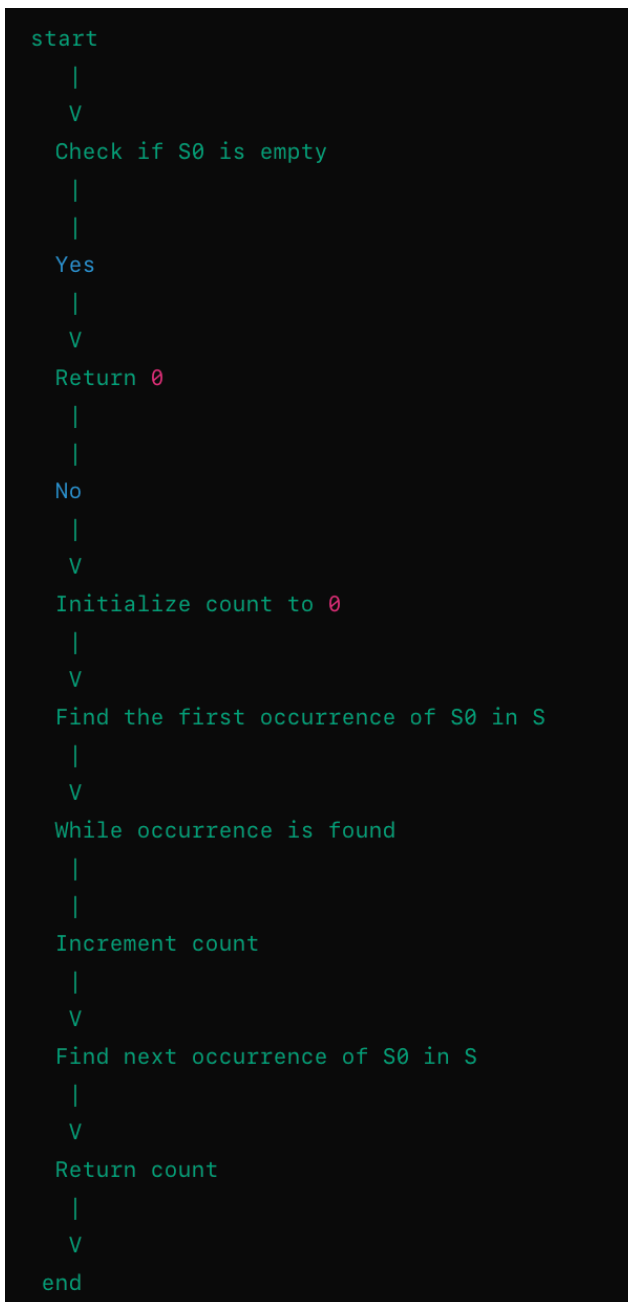
А) Метод `find` в стандартному класі `string` приймає рядок `s`, початкову позицію `pos` та кількість символів `n`, і повертає позицію першого входження підрядка `s` у рядок, починаючи з позиції `pos` та не перевищуючи `n` символів. Якщо підрядок не знайдено, повертається `string::npos`.

В) Функція `size_t find (const char* s, size_t pos, size_t n) const` приймає три параметри:

1. s - це вказівник на масив символів типу `const char*`, який представляє рядок, в якому ми шукаємо підрядок.
2. pos - це беззнакове ціле число типу `size_t`, яке вказує на позицію, з якої починається пошук у рядку s .
3. n - це беззнакове ціле число типу `size_t`, яке обмежує максимальну кількість символів, які можна врахувати при пошуку.

Результатом роботи цієї функції є беззнакове ціле число типу `size_t`, яке вказує на позицію першого входження підрядка у рядок s , починаючи з позиції pos та не перевищуючи n символів. Якщо підрядок не знайдено, повертається константа `string::npos`, яка зазвичай має значення $(size_t)-1$.

Діаграма алгоритму:



Завдання 2.

А) Функція `countMatches` обробляє рядок `S` та підрядок `S0`, знаходячи кількість входжень підрядка `S0` у рядок `S`.

В) Дана функція перевіряє, чи відповідає рядок умовам завдання. У цьому випадку, умова полягає в тому, щоб знайти кількість входжень підрядка `S0` у рядок `S`.

Лістинг коду вирішення задачі 1,2 наведено в дод. А (стор. 6).

Екран роботи програми наведено в дод. Б (стор. 9).

ВИСНОВКИ

В ході виконання лабораторної роботи 8 було вивчено теоретичний матеріал з основ роботи з низькорівневими рядками на C++ і документацію до класу `string`, а також алгоритми пошуку в рядку, а також реалізувати обробку рядків на C++ в середовищі Visual Studio, а також відпрацьовані навички написання коду.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <cstring>

using namespace std;

void task26();
size_t find(const char* s, const char* point, size_t pos, size_t n);

void sstring32();
int countMatches(const string& S, const string& S0);

int main() {

    int choice = 0;

    while (choice != 3) {

        cout << "Choose task: "
              "\n1.Task26"      // size_t find (const char* s, size_t pos,
size_t n) const;
              "\n2.String32"
              "\n3.Exit" << endl;

        cin >> choice;

        switch (choice) {
        case 1: {
            task26();          //Task task26
            break;
        }
        case 2: {
            sstring32();        //Task string32
            break;
        }
        case 3: {
            cout << "Program is end!";
            break;
        }
        default: {
            cout << "Wrong one, choose again!\n";
        }
    }
}

void task26() {

```

```

    const char* s = "Hello, i'm a student Dikhtiarenko Mikhail, studying in
    Kharkov Aerospace Institute";
    const char* target = "studying";
    size_t pos = 0;
    size_t n = 120;

    size_t result = find(s, target, pos, n);

    if (result != string::npos) {
        cout << "Found at index: " << result << endl;
    }
    else {
        cout << "Not found" << endl;
    }
}

```

```

size_t find(const char* s, const char* point, size_t pos, size_t n) {
    size_t len_s = 0;
    size_t len_target = 0;

    while (s[len_s] != '\0') {
        len_s++;
    }

    while (point[len_target] != '\0') {
        len_target++;
    }

    if (len_target == 0) {
        return pos;
    }

    if (pos >= len_s || len_target > n) {
        return string::npos;
    }

    for (size_t i = pos; i <= len_s - len_target && i < pos + n; ++i) {
        size_t j = 0;
        while (j < len_target && s[i + j] == point[j]) {
            j++;
        }
        if (j == len_target) {
            return i;
        }
    }

    return string::npos;
}

```

```

void sstring32() {

```

```
string S, S0;

cout << "Enter S: ";
cin >> S;

cout << "Enter S0: ";
cin >> S0;

int result = countMatches(S, S0);
cout << "Count matches: " << result << endl;
}

int countMatches(const string& S, const string& S0) {
    if (S0.empty()) {
        return 0;
    }

    int count = 0;
    size_t pos = S.find(S0);

    while (pos != string::npos) {
        count++;
        pos = S.find(S0, pos + S0.length());
    }

    return count;
}
```


ДОДАТОК Б
Скрін-шот вікна виконання програми

```
Choose task:
1.Task26
2.String32
3.Exit
1
Found at index: 43
Choose task:
1.Task26
2.String32
3.Exit
2
Enter S: adadadadadad
Enter S0: ad
Count matches: 6
Choose task:
1.Task26
2.String32
3.Exit
```

Рисунок Б - Екран виконання програми для вирішення завдання 1,2