

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 ++"»

ХАІ.301.електроенергетика,електромеханіка і електротехніка  
319а,№23ЛР

Виконав студент гр. 319а

Дмитро Хара  
(П.І.Б.)

(підпис, дата)

Перевірив

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

## МЕТА РОБОТИ

низькорівневими рядками  
алгоритми пошуку в рядку, а  
Visual Studio.Завдання 1.

Вивчити теоретичний матеріал з основ роботи з  
на C++ і документацію до класу `string`, а також  
також реалізувати обробку рядків на C++ в середовищі

А. Вивчити по документації метод  
стандартного класу `string` відповідно `size_t find (const string& str, size_t pos = 0) const; .`  
В. Визначити функцію, що виконує ті ж дії, що і  
вивчений метод класу  
параметром (масив символів).  
функції обробки рядків зі

string. Вихідний рядок передати першим  
Для реалізації методу не використовувати

стандартних бібліотек.  
С. Викликати свій метод і метод `string`  
наведеними в дод.А. \*Перед викликом ввести  
зберегти в масиві символів і змінній типу  
`string`.

Для виконання цього завдання спочатку потрібно вивчити, як працює метод `find`  
класу `string` у стандартній бібліотеці C++. Документацію можна знайти на офіційному  
сайті C++ ([cppreference.com](http://cppreference.com) або [cplusplus.com](http://cplusplus.com)).

Метод `find` шукає підрядок `str` у рядку і повертає позицію першого входження цього  
підрядка. Якщо підрядок не знайдено, повертається `string::npos`.

Отже, ми можемо створити функцію, яка виконує ті ж дії, що і метод `find` класу `string`,  
але без використання стандартних функцій обробки рядків.

Ось приклад реалізації цієї функції:

```
include <iostream>
```

char

```
size_t custom_find(const char, const
```

```
if  
return
```

```
size_t 0  
size_t 0
```

```
while '\0'  
while '\0'
```

```
if  
return
```

```
for size_t 0
```

```
size_t
```

```
for 0  
if  
break
```

```
if  
return
```

```
return
```

```
int main  
char 100
```

```

        "Enter a string: "
        getline      100

        "Enter a substring to find:
"

        size_t      custom_find
        c_str
        string      find
        size_t

        "Custom find result: "

        "String find result: "

        return 0

```

Цей код виконує те ж саме, що й метод `find` класу `string`, але реалізований власноручно без використання стандартних функцій обробки рядків.

## Завдання 2.

А.Описати функцію, що обробляє рядок відповідно Дано рядок, що зображає арифметичний вираз виду «<цифра> ±

<цифра> ± ... ± <цифра>», де на місці знака операції «±» знаходиться

символ «+» або «-» (наприклад, «4 + 7 - 2 - 8»). Вивести значення

цього виразу (ціле число).. 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.

Д.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.

А. Функція для обробки арифметичного виразу у рядку та обчислення його значення:

```
include <iostream>
```

```
include <string>
```

```
include <sstream>
```

```
int evaluateExpression const
```

```
ss
```

```
int
```

```
char
```

```
while
```

```
int
```

```
if      '+'
```

```
else if  '-'
```

```
return
```

В. Функція для перевірки, чи задовольняє рядок умовам завдання:

```
include <string>
```

```
bool isValidExpression const
```

```
for char
```

```
if isdigit      '+'      '-'      ''
```

```
return false
```

```
return true
```

С. Створення текстового файлу з 10 різними рядками:

```
include <fstream>
```

```
void createInputFile
```

```
    file "input.txt"
```

```
if      is_open
```

```
    "4 + 7 - 2 - 8\n"
```

```
    "10 - 5 + 3\n"
```

```
    "6 + 9 - 1\n"
```

```
    "20 - 15 - 2\n"
```

```
    "3 + 5 + 7\n"
```

```
    "8 - 4 - 2\n"
```

```
    "9 + 2 - 1 - 5\n"
```

```
    "15 - 6 + 4\n"
```

```
    "20 - 8 + 9\n"
```

```
    "7 + 3 - 1\n"
```

```
close
```

D. Оброблення рядків файлу з використанням функцій 2.A та 2.B:

```
include <iostream>
```

```
include <fstream>
```

```
int evaluateExpression const
```

```
bool isValidExpression const
```

```
void processFile const
```

```
file
```



```

        output "output.txt"

    if      is_open      is_open

    while      getline

    if isValidExpression

    int      evaluateExpression

        "Expression: "      ", Result: "

    else

        "Invalid expression: "

    close

    close

    else

        "Unable to open file."

```

Е. Виклик функції processFile та введення додаткових даних з консолі:

```
int main
```

```
createInputFile
```

```
"Enter input filename: "
```

```
processFile
```

```
"Output has been written to output.txt."
```

```
return 0
```

Цей код створює вихідний файл з обробленими результатами виразів, зчитаних з вхідного файлу.