

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт
з лабораторної роботи №9 з дисципліни
«Основи програмування»
«Рядки»
Варіант 34

Виконав студент ІП-1134 Шамков Іван Дмитрович
(прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2021
Лабораторна робота №9
Рядки

Лабораторна робота 9

Рядки

Мета – ознайомитися з особливостями реалізації текстових рядків, опанувати технологію їх використання, навчитися розробляти алгоритми та програми із застосуванням рядків.

Варіант: 34

Умова задачі:

34. Заданий рядок, що містить розділені пробілами слова. Відсортувати слова за першим символом кожного слова.

Математична модель:

Змінна	Тип	Ім'я	Призначення
Набір слів, поділених пробілами	Символьний	line	Початкове дане
Масив слів із заданого рядка	Символьний	A	Результат
Розмір масиву	Цілий	sizeA	Проміжне значення
Лічильник, що вказує на поточну кількість слів у масиві	Цілий	counter	Проміжне значення
Змінна для запису поточного слова	Символьний	word	Проміжне значення
Лічильник	Цілий	i	Проміжне значення
Лічильник	Цілий	j	Проміжне значення
Лічильник	Цілий	k	Проміжне значення

Змінна для перевірки умови, чи не є символи слова однаковими і у слові ще є символи	Логічний	Bool	Проміжне значення
Показчик на задане користувачем речення	Символьний	ptr	Проміжне значення
Буфер обміну поточного та наступного елементів масиву місцями	Символьний	tmp	Проміжне значення

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб отримати від користувача речення(масив символів), який потім ми розрізаємо на окремі слова. Із цих слів створюємо окремий масив, який потім сортуємо в алфавітному порядку. У кінці виводимо відсортований масив.

Текст:

```
#include <iostream>
```

```
#include <string>
```

```
#include <Windows.h>
```

```
#include <locale>
```

```
using namespace std;
```

```
string CreateLine();//Занум у користувача речення
```

```
int SplitWords(string*, char*);//Розділення речення на слова та заповнення  
одновимірного масиву
```

```
void Sort(string*, int);//Сортування масиву слів в алфавітному порядку
```

```
void CoutArray(string*, int);//Виведення масиву
```

```
int main() {
```

```
    //Підключаємо можливість використання Кирилиці
```

```
    SetConsoleCP(1251);
```

```
    SetConsoleOutputCP(1251);
```

```
    string line;
```

```
    string A[100];
```

```
    line=CreateLine();
```

```
    cout << "\n\nYour sentence is: " << line << endl;
```

```
    int sizeA=SplitWords(A, &line[0]);
```

```
    Sort(A, sizeA);
```

```
    cout << "\n\nSorted sentence is: ";
```

```
    CoutArray(A, sizeA);
```

```
}
```

```
string CreateLine() {
```

```
    string line;
```

```

        cout << "Enter some words: ";
        getline(cin, line);

        return line;
    }

int SplitWords(string* A, char* ptr) {
    string word="";
    int counter = 0;
    for (int i = 0; (ptr[i] != '\0') ;++i) {
        if (ptr[i] != ' ') word += ptr[i];

        if ( ptr[i]== ' ' || (ptr[i + 1] == '\0')) {
            A[counter] = word;
            counter++;
            word = "";
        }
    }

    return counter;
}

```

```

void Sort(string* A, int sizeA) {
    //Використання сортування обміном
    bool Bool;
    for (int i = 0; i < sizeA; i++) {

```

```

    for (int j = 0; j < sizeA - 1; j++) {
        Bool = true;
        for (int k = 0; A[j][k] != '\0' && A[j + 1][k] != '\0' && Bool;
k++) {//Третій цикл для проходу по символам слова

            if (A[j][k] > A[j + 1][k]) {//Перевірка умови, чи
зустрічається символ поточного слова в алфавіті, ніж поточного слова

                string tmp = A[j + 1];
                A[j + 1] = A[j];
                A[j] = tmp;
                Bool = false;
            }

            if (A[j][k] < A[j + 1][k]) {

                Bool = false;
            }

            if (A[j][k] == A[j + 1][k] && (A[j + 1][k + 1] == '\0'))
//Додаткова умова для врахування останнього слова та його додавання

                string tmp = A[j + 1];
                A[j + 1] = A[j];
                A[j] = tmp;
                Bool = false;
            }

        }

    }
}

```

```

}

void CoutArray(string* A, int sizeA) {
    for (int k = 0; k < sizeA; k++) {
        cout << A[k] << " ";
    }
    cout << "\n\n";
}
}

```

Текст файла проекту:

C++:

```

1  #include <iostream>
2  #include <string>
3  #include <Windows.h>
4  #include <locale>
5
6  using namespace std;
7
8  string Createline(); //Запит у користувача речення
9  int SplitWords(string*, char*); //Розділення речення на слова та заповнення одновимірного масиву
10 void Sort(string*, int); //Сортування масиву слів в алфавітному порядку
11 void CoutArray(string*, int); //Виведення масиву
12
13 int main() {
14     //Підключаємо можливість використання Кирилиці
15     SetConsoleCP(1251);
16     SetConsoleOutputCP(1251);
17
18     string line;
19     string A[100];
20     line=Createline();
21
22     cout << "\n\nYour sentence is: " << line << endl;
23
24     int sizeA=SplitWords(A, &line[0]);
25     Sort(A, sizeA);
26
27     cout << "\n\nSorted sentence is: ";
28     CoutArray(A, sizeA);
29 }
30
31 string Createline() {
32     string line;
33     cout << "Enter some words: ";
34     getline(cin, line);
35
36     return line;
37 }
38

```

```

41 int SplitWords(string* A, char* ptr) {
42     string word="";
43     int counter = 0;
44     for (int i = 0; (ptr[i] != '\0') ;++i) {
45         if (ptr[i] != ' ') word += ptr[i];
46
47         if ( ptr[i]==' ' || (ptr[i + 1] == '\0')) {
48             A[counter] = word;
49             counter++;
50             word = "";
51         }
52     }
53
54     return counter;
55 }
56

```

```

57 void Sort(string* A, int sizeA) {
58     //Використання сортування обміном
59     bool Bool;
60     for (int i = 0; i < sizeA; i++) {
61         for (int j = 0; j < sizeA - 1; j++) {
62             Bool = true;
63             for (int k = 0; A[j][k] != '\0' && A[j + 1][k] != '\0' && Bool; k++) { //Третій цикл для проходу по символам слова
64
65                 if (A[j][k] > A[j + 1][k]) { //Перевірка умови, чи зустрічається символ поточного слова в алфавіті, ніж поточного слова
66                     string tmp = A[j + 1];
67                     A[j + 1] = A[j];
68                     A[j] = tmp;
69                     Bool = false;
70                 }
71
72                 if (A[j][k] < A[j + 1][k]) {
73                     Bool = false;
74                 }
75
76                 if (A[j][k] == A[j + 1][k] && (A[j + 1][k + 1] == '\0')) { //Додаткова умова для врахування останнього слова та його додавання
77                     string tmp = A[j + 1];
78                     A[j + 1] = A[j];
79                     A[j] = tmp;
80                     Bool = false;
81                 }
82             }
83         }
84     }
85 }
86
87

```

```

88 void CoutArray(string* A, int sizeA) {
89     for (int k = 0; k < sizeA; k++) {
90         cout << A[k] << " ";
91     }
92     cout << "\n\n";
93 }
94

```


Копії екранних форм:

C++:

```
Microsoft Visual Studio Debug Console
Enter some words: you can't escape the fate of the chosen one
Your sentence is: you can't escape the fate of the chosen one
Sorted sentence is: can't chosen escape fate of one the the you
```

```
Microsoft Visual Studio Debug Console
Enter some words: bb ba br by bz cz at pr p n
Your sentence is: bb ba br by bz cz at pr p n
Sorted sentence is: at ba bb br by bz cz n p pr
```

```
Microsoft Visual Studio Debug Console
Enter some words: play pla pl p
Your sentence is: play pla pl p
Sorted sentence is: p pl pla play
```

Висновок

Отже, виконавши цю лабораторну роботу, ми навчилися працювати з текстовими рядками. Проекти, на мою думку, розроблені коректно, адже заплановані елементи працюють, а саме: отримання даних від користувача, виклик функцій поділу речення на слова, переміщення їх у масив, сортування цього масива слів в алфавітному порядку, а також виведення усіх цих значень на екран. Ділимо наше речення на слова через підпрограму `SplitWords(string* A, char* ptr)`, а потім сортуємо через `Sort(string* A, int sizeA)`. Виводимо відсортований масив через підпрограму `CoutArray(string* A, int sizeA)`.