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-1.

Базові конструкції»

«Рядки»

Варіант 31

Виконав студент ІП-11 Трикош Іван Володимирович

Перевірила Вітковська І. І.

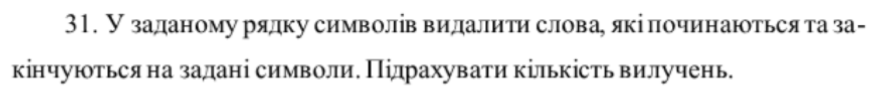
Київ 2021

**Лабораторна робота №9**

**Рядки**

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

Варіант 31.



**Постановка задачі.** Знаходимо позицію елемента, який увів користувач. Якщо він стоїть на початку слова або перед ним стоїть ' ', то переміщуємося у кінець слова і перевіряємо, чи там стоїть заданий символ. Якщо умова виконується, то видаляємо слово. В кінці виводимо результат.

**Код та результат програми на С++:**

#include <iostream>

#include <string>

using namespace std;

void Init(string\*, char\*); // Ввід рядка та символу

string Solution(string, char\*, int\*, int\*); // Обробка рядка

void Browse(string\*, int\*, int\*); // Вивід кінцевого рядка

int find(string, char, int); // Знаходження першого входження символу у рядок

string del(string, int, int, int); // Видалення слова

string delspace(string);

int main()

{

string str; // Оголошуємо рядок

char m; // Символ, який ввів користувач

int s = 0; // Кількість вилучень

int k = 2; // Для виводу результату

Init(&str, &m);

str = Solution(str, &m, &s, &k);

Browse(&str, &s, &k);

}

void Init(string\* str, char\* m)

{

cout << "Input a line:\n";

getline(cin, \*str); // Вводимо рядок

cout << "Input a symbol: ";

cin >> \*m; // Вводимо символ

}

int find(string str, char m, int length)

{

int pos = -1; //Позиція елемента

bool flag = true; //Для знаходження першого входження елемента

for (int i = 0; i < length; i++)

{

if ((str[i] == m) && flag) // Знаходимо індекс заданого елемента

{

pos = i;

flag = false;

}

}

return pos;

}

string del(string str, int q, int k, int len)

{

string l = "";

for (int i = 0; i < len; i++)

{

if (i < q) // Якщо індекс менший за позицію першого символу слова, то

{

l += str[i];

}

else if (i >= (k + q)) //Якщо індекс більший за позицію останнього символу слова, то

{

l += str[i];

}

}

return l;

}

string Solution(string str, char\* m, int\* s, int\* k)

{

int len, len1 = 0; // Оголошуємо змінну для довжини рядка і слова

int n = -1; // Позиція цього символу

int b = n; // Попередня позиція символу

char n1; // Поточний символ

bool flag = true; // Для визначення, чи потрібно видаляти слово

bool flag1 = true; // Для виходу з циклу

int i = 0; // Для знаходження елемента

len = str.length(); // Визначаємо довжину рядка

n = find(str, \*m, len); // Знаходимо позицію елемента, який ввів користувач

if (len == 0) // Перевіряємо, чи рядок пустий

{

\*k = 0;

}

else if (n == -1) // Перевіряємо, чи містить рядок задані символи

{

\*k = 1;

}

else

{

if (n == 0) // Якщо потрібно, видаляємо перше слово у рядку

{

for (int i = n; i < len - 1; i++)

{

n1 = str[i]; // Поточний елемент

if (((str[i + 1] == ' ') || (str[i + 1] == '\0')) && (n1 == \*m) && flag) // Якщо після нього стоїть " ", то видаляємо слово

{ // то видаляємо слово

len1 = i - n + 1; // Довжина слова

flag = false;

}

}

if (len1 != 0)

{

str = del(str, n, len1, len); // Видаляємо слово

\*s += 1;

len1 = 0;

}

}

flag = true;

n = -1;

cout << str << endl;

len = str.length();

n = find(str, \*m, len); // Знаходимо позицію елемента, який ввів користувач

while ((n != -1) && flag1)

{

b = n;

len1 = 0;

if (str[n - 1] == ' ') // Якщо перед символом стоїть ' ', то перевіряємо, чи потрібно видаляти слово

{

for (int j = n - 1; j < len; j++)

{

n1 = str[j]; // Поточний елемент

if (((str[j + 1] == ' ') || (str[j + 1] == '\0')) && (n1 == \*m) && flag) // Якщо після нього стоїть " ", або 0-ий символ,

{ // то видаляємо слово

for (int k = n; k <= j; k++)

{

if (str[k] == ' ') // Якщо у виразі є ' ', то пропускаємо його

{

flag = false;

}

}

if (flag)

{

len1 = j - n + 1; // Довжина слова

flag = false;

}

}

}

}

if (len1 != 0)

{

str = del(str, n, len1, len); // Видаляємо слово

\*s += 1;

}

cout << str << endl;

n = -1;

len = str.length();

n = find(str, \*m, len); // Знаходимо позицію елемента, який ввів користувач

if (n == b)

{

n = find(str, \*m, len); // Знаходимо позицію з n+1

b = n;

len = str.length(); // Визначаємо довжину рядка

if ((n+1) == len)

{

flag1 = false;

if (str[n - 1] == ' ')

{

str = del(str, n, len1, len);

}

}

}

flag = true;

i++;

}

str = delspace(str);

}

return str;

}

string delspace(string str)

{

string h = "";

for (int i = 0; i < str.length(); i++) // Видаляємо пробіли

{

if (str[i] != ' ')

{

h += str[i];

}

}

return h;

}

void Browse(string\* str, int\* s, int\* k)

{

if (\*k == 0)

{

cout << "The string is empty!";

}

else if (\*k == 1)

{

cout << "This element is not in the string!";

}

else

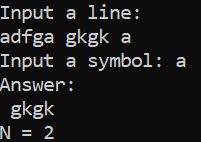
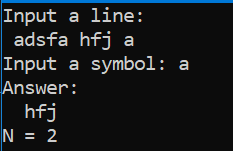
{

cout << "Answer:\n" << \*str << endl;

cout << "N = " << \*s;

}

}

**** ****

**Код та результат програми на Python:**

def Init():

s1 = input("Input the string:\n") # Вводимо рядок

m = input("Input the symbol: ") # Вводимо символ

return s1, m # Повертаємо введені значення

def list(str, length):

st = []

pos = 0 # Індекс першого символу, який не є пробілом

m = "" # Слово

j = True # Для знаходження pos

for i in range(length):

if j and str[i] != " ":

pos = i # Індекс першого символу, який не є пробілом

j = False

for i in range(length): # Створюємо список зі слів рядка

if str[i] != " ":

m += str[i] # Накопичуємо сиволи у слові

elif i >= pos:

st = st + [m] # Збільшуємо список

m = ""

if (i == length - 1) and (str[i] != " "):

st = st + [m] # Збільшуємо список

return st

def str1(l, length): # Перетворюємо список у рядок

s = ""

for i in range(length):

s += l[i] + " " # Збільшуємо рядок

return s

def find(str, m, length): # Знаходимо позицію заданого елемента

pos = -1

for i in range(length):

if str[i] == m:

pos = i

return pos

def delete(l, i, length): # Видаляємо елемент зі списку (пропускаємо його)

s = []

for j in range(i):

s = s + [l[j]]

for j in range(i+1, length):

s = s + [l[j]]

return s

def Solution(str, m):

k = 2 # Для виводу результату

length = len(str) # Довжина рядка

n = 0 # Кількість вилучень

pos = find(str, m, length) # Знаходимо позицію заданого елемента

if length == 0: # Якщо рядок пустий, то вовидимо відповідне повідомлення

k = 0

elif pos == -1: # Якщо рядок не містить заданого символу, то виводимо відповідне #повідомлення

k = 1

else:

l = list(str, length) # Розбиваємо рядок на список

i = 0 # Індекс елемента списку

while i < len(l):

s = l[i] # і-тий елемент списку

if (s[0] == m) and (s[-1] == m): # Перевіряємо, чи заданий символ є в кінці #чи на початку слова

l = delete(l, i, len(l)) # Видаляємо елемент зі списку

n += 1 # Збільшуємо кількість видалень

i -= 1 # Зменшуємо індекс, бо у списку стало на 1 елемент менше

i += 1 # Збільшуємо індекс

str = str1(l, len(l)) # Перетворюємо список у рядок

return str, n, k # Повертаємо рядок, кількість вилучень та k

def Browse(s2, s, k): # Виводимо відповідний результат

if k == 0:

print("The string is empty!")

elif k == 1:

print("This element is not in the string!")

else:

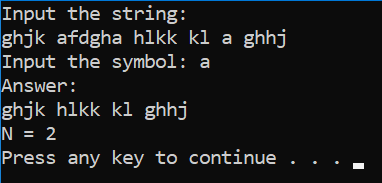
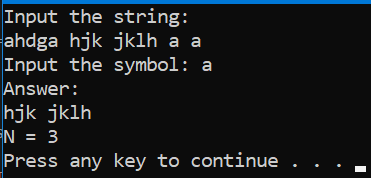
print("Answer:\n", s2, sep = "")

print("N =", s)

s1, m = Init() # Вводимо рядок та символ

s1, s, k = Solution(s1, m) # Видаляємо слова

Browse(s1, s, k) # Виводимо результат



**Висновок –** я ознайомився з особливостями реалізації текстових рядків, опанував технологію їх використання, навчився розробляти алгоритми та програми із застосуванням рядків, покращив навички програмування на мовах C++ та Python, удосконалив навички розробки алгоритмів.

У результаті роботи програми я одержав рядок, в якому видалені слова, які починаються і закінчуюються на заданий символ.