Основи програмування – 2. Алгоритми та структури даних

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 № 5 з дисципліни

«Алгоритми та структури даних-2.

Структури даних»

«Спискові структури даних»

Варіант 5

Виконав студент ІП-15, Буяло Дмитро Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив Соколовський Владислав Володимирович

(прізвище, ім'я, по батькові)

Київ 2022

Основи програмування – 2. Алгоритми та структури даних

**Лабораторна робота 5**

**Спискові структури даних**

**Мета** – вивчити основні підходи формалізації евристичних алгоритмів і вирішення типових задач з їх допомогою.

**Індивідуальне завдання**

**Варіант 5**

**Завдання**

Заданий текст, що містить декілька рядків слів, розділених пробілами. Використовуючи стек, елементами якого є слова, надрукувати текст, в якому слова кожного рядка містяться у зворотному порядку.

Основи програмування – 2. Алгоритми та структури даних

**Виконання**

1. Псевдокод алгоритмів

|  |
| --- |
| **Початок** |
| **Повторити** для і від 0 до довжини s |
| **Якщо** s[i] - пробіл |
| **то** |
| push(word) |
| word = “” |
| **Інакше** |
| word += s[i] |
| **Все повторити** |
| **Повторити** |
| **Поки** top |
| res += pop() + ‘ ‘ |
| **Все повторити** |
| res += ‘\n‘ |
| **Кінець** |

1. Програмна реалізація
   1. Вихідний код

#include <iostream>

#include <string>

using namespace std;

struct Node {

string data;

Node\* next;

};

class myStack {

Node\* top;

public:

myStack() :top(NULL) {}

~myStack();

void push(string);

string pop();

void reverse(string, string&);

};

myStack::~myStack() {

while (top) {

Node\* p = top;

top = top->next;

delete p;

}

}

void myStack::push(string word) {

Node\* temp = new Node;

temp->data = word;

temp->next = top;

top = temp;

}

string myStack::pop() {

string word;

Node\* temp = top;

if (!temp) {

cout << "Stack is empty" << endl;

return 0;

}

word = temp->data;

top = temp->next;

delete temp;

return word;

}

void myStack::reverse(string s, string& res) {

string word;

for (int i = 0; i < s.length(); i++) {

if (isspace(s[i])) {

push(word);

word = "";

}

else {

word += s[i];

}

}

while (top) {

res += pop() + ' ';

}

res += '\n';

}

void clear(string&);

int main()

{

string res = "";

myStack st;

int ascii = 27; // Ctrl + [

string line = " ";

cout << "Press Enter to go to the next line\nPress Ctrl + [ to end writing\n\nEnter your text:\n";

while (int(line[0]) != ascii) {

getline(cin, line);

if (line == "") line = " ";

clear(line);

st.reverse(line, res);

}

cout << "Your reversed text:" << endl;

res.erase(res.length() - 3, 3); // убираем \n (-4, 4 - если вообще без абзацов)

cout << res;

}

void clear(string& s) {

for (int i = 1; i < s.length(); i++) {

if (isspace(s[i - 1]) && isspace(s[i])) {

s.erase(i, 1);

i--;

}

}

if (s.length() > 1 && isspace(s[0])) { // видаляє зайвий пробіл на початку речення

s.erase(0, 1);

}

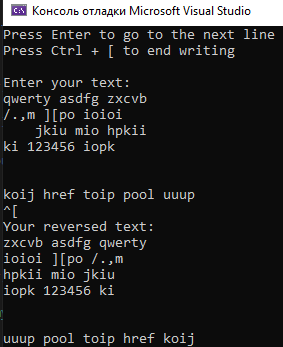
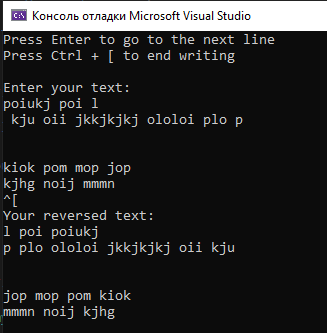
if (!isspace(s[s.length() - 1])) {

s += ' ';

}

* 1. Приклади роботи

На рисунках 2.1 та 2.2 показані приклади роботи програми



|  |  |
| --- | --- |
| Рисунок 2.1 | Рисунок 2.2 |

**ВИСНОВОК**

При виконанні п’ятої лабораторної роботи, було створено стек, за допомогою якого було виведено слова кожного рядку в зворотному порядку. Стек для цього завдання підходить найбільше, через його особливість – Last Input First Output.