Національний технічний університет України

«Київський політехнічний інститут»

Кафедра автоматизованих систем обробки інформації та управління

Звіт про виконання лабораторної роботи №4 з дисципліни

«Алгоритми та структури даних»

Перевірено: Ковалюк Т.В. Виконав ст. 1 курсу ФОІТ

Проскура С.Л. гр. ІС-52

Дорошенко А.В.

Київ 2016

## Лабораторна робота 4

# **Динамічні структури. Стеки**

Варіант № 9

*Мета:*

1. Вивчити особливості використання  динамічних структур типу стек.
2. Навчитися застосовувати динамічні структури типу стек в програмуванні

### Завдання

|  |  |
| --- | --- |
| В текстовому файлі задані арифметичні вирази. Визначити кількість операндів і операцій в арифметичних виразах, використовуючи для цього стек Вивести на екран значення елементів стеку. |  |

### Блок-схема алгоритму



Рис.1 (Блок-схема алгоритму вставки елемента до стеку)



Рис.2 (Блок-схема головного алгоритму)



Рис.3 (Блок-схема алгоритму визначення кількості операндів і операцій та виведення елементів стеку)



Рис.4 (Блок-схема алгоритму видалення елемента зі стеку і збереження його значення)

### Код програми

//Laboratory work

//written by student of the first curse

//of the group IC-52

//Anton Doroshenko

//2016.04.03

//==========================================================================

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include<iostream>

#include<conio.h>

using namespace std;

//Стек

struct Item

{

char val;

Item\* next;

};

char str[100];//допоміжний рядок

char sym;//змінна для зберігання значень символів

char\* ptrs = &sym;

char oper[] = "abcdefghijklmnopqrstuvwxyz";//операнди

char operatio[] = "+-\*/";//операції

int operand = 0;

int operations = 0;

Item\* head = NULL;

Item\* cur;

FILE\* f;

//додавання елемента до стеку

void push(char value)

{

cur = new(Item);

cur->val = value;

cur->next = head;

head = cur;

}

//видалення та зберігання елемента стеку

void pop(char\* value)

{

cur = head;

\*value = head->val;

head = cur->next;

delete cur;

}

//Визначення кількості операндів і операцій та виведення елементів стеку

void outElemOfStack()

{

while (head != NULL)

{

pop(ptrs);

cout << sym << " ";

if (strchr(oper, sym) != 0)

{

operand++;

}

else if (strchr(operatio, sym) != 0)

{

operations++;

}

}

}

int main()

{

cout << "laboratory work number 3 made by Anton Doroshenko, IS-52" << endl;

f = fopen("fff.txt", "r");

int k = 1;

while (!feof(f))

{

fscanf(f, "%s", str);

for (int i = 0; i < strlen(str); i++)

{

push(str[i]);

}

cout << "Elements of stack:" << endl;

outElemOfStack();

cout << endl;

cout << k << " expression has: " << operand << " operands and " << operations << " operations" << endl;

k++;

operand = 0;

operations = 0;

}

system("pause");

fclose(f);

}

1. **Screen Shot результатів**

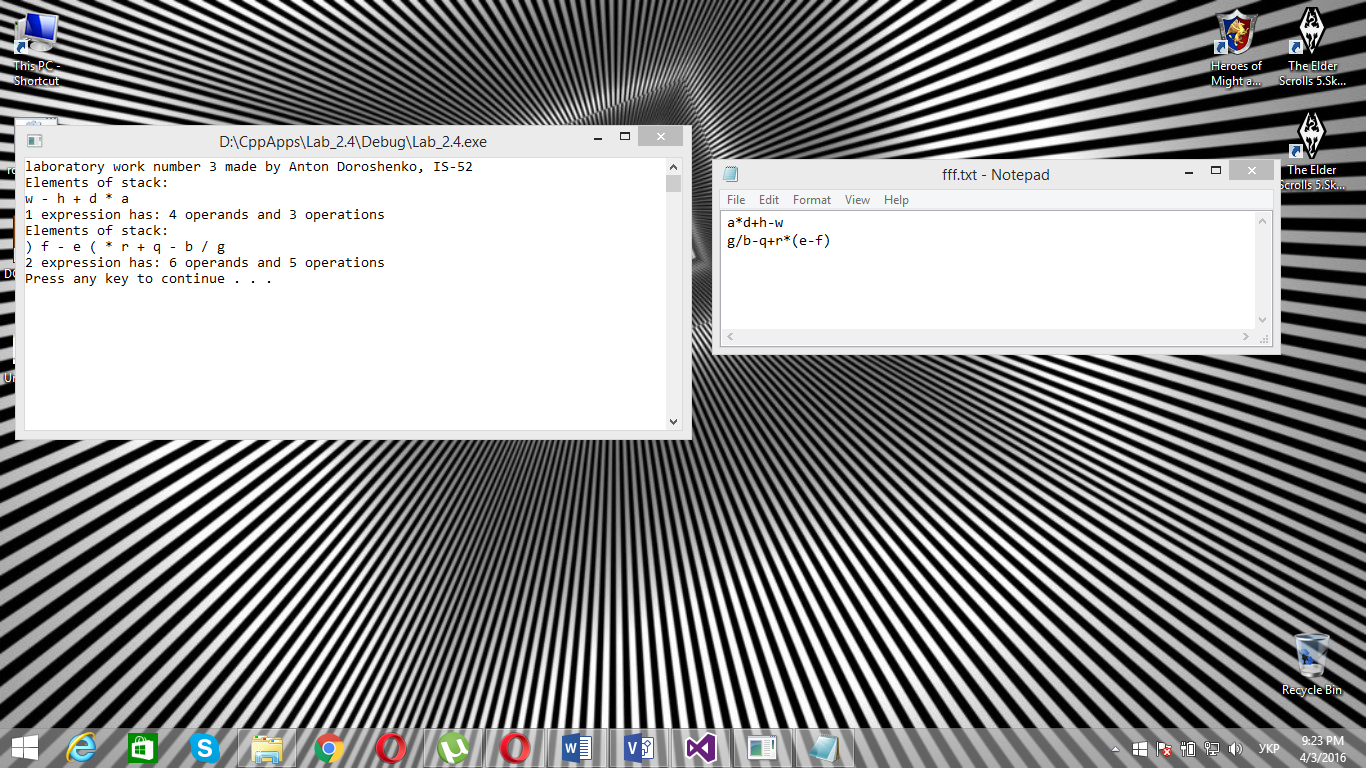


Рис.5 (Screen Shot вхідного файла)

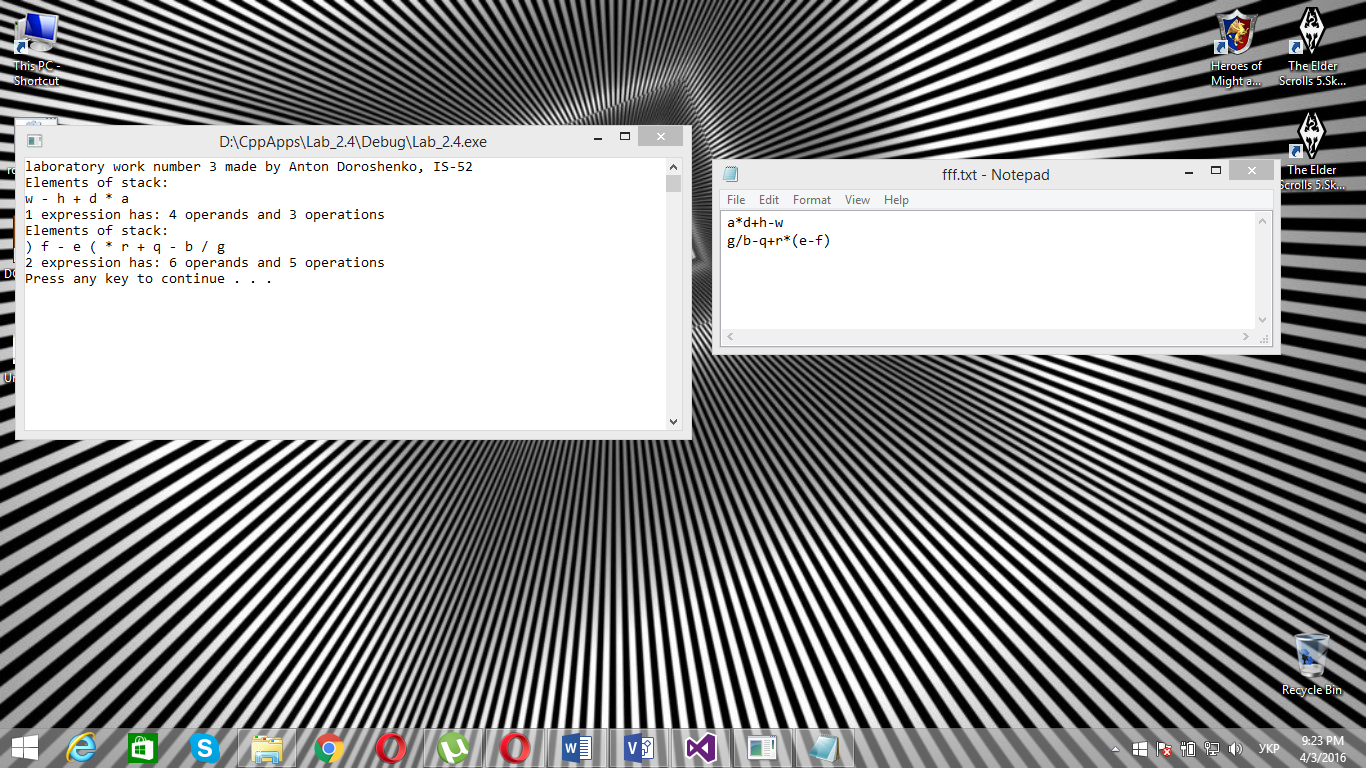


Рис.6 (Screen Shot результатів)

1. **Аналіз результатів**

Як ми бачимо зі ScreenShot’ів

Програма зчитує дані з файла, записує у стек та обробляє дані. Програма виводить в консоль кількість операндів і операцій та елементи стеку.

1. **Висновок**

Програма працює правильно, про що свідчить аналіз результатів та ScreenShot’и. Програма коректно виводить результати. Програма застосовує динамічну структуру даних стек.