МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №10**

**з дисципліни «Основи програмування»**

**на тему: *«*Лінійні динамічні структури.»**

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. Нежуміра О.І.

Дніпро, 2020

Тема. Лінійні динамічні структури.

Мета. Навчитися реалізовувати лінійні динамічні структури. Оволодіти навичками використання лінійних динамічних структур в розробці програм.

**1. Завдання**

Розробити програму для заданого індивідуального завдання. Для реалізації завдання обрати найзручніші динамічні лінійні структури.

Варіант 5

Дано символьний рядок, що містить правильний математичний вираз такого вигляду:

<формула> :: = <цифра> | M (<формула>, <формула>) | m (<формула>, <формула>)

M – операція обчислення max з двох виразів, m – операція обчислення min з двох виразів. Написати функцію обчислення значення цього виразу.

**2. Зовнішні специфікації**

Програма повинна реалізовувати такі дії:

* забезпечення вводу користувачем вхідних даних для обчислення;
* обчислення значення виразу;
* вивід результатів обчислення.

Вхідні дані: символьний рядок з правильним математичним виразом.

Вихідні дані: результат обчислення виразу.

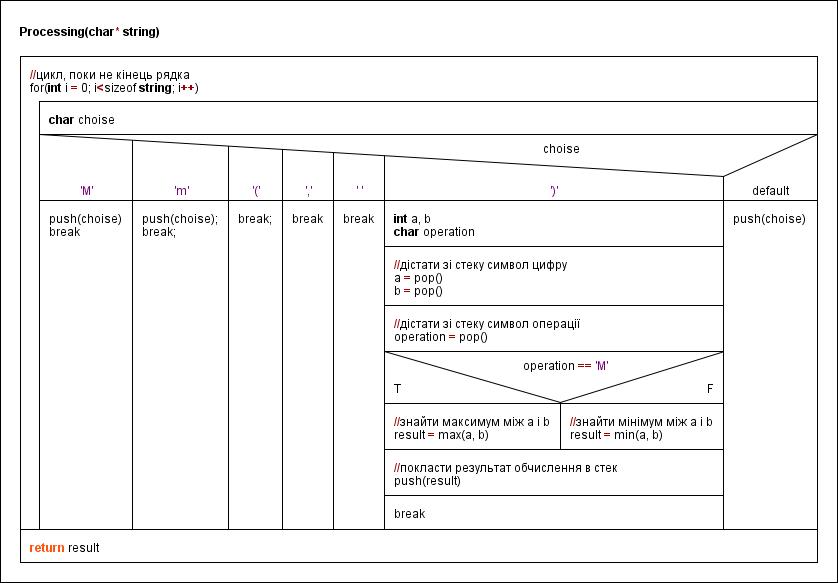
**3. Опис структури даних для реалізації завдання**

Для реалізації даного завдання використовувався стек. Він дозволяє працювати лише з його верхівкою і по-черзі класти та витягувати дані з нього.

**4. Тести**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва тесту | Вхідні дані | Очікувані результати |
| 1. | Одна операція | M(5, 1) | 5 |
| 2. | Вкладені операції | M(m(5, 2),M(9,8)) | 9 |
| 3. | Нулі в параметрах | M(0, 0) | 0 |

**5. Алгоритм**



**6. Текст програми**

Файл Source.cpp

//Дано символьний рядок, що містить правильний математичний вираз вигляду

//<формула> :: = <цифра> | M (<формула>, <формула>) | m (<формула>, <формула>)

//Вхід: рядок з виразом

//Вихід: результат обчислення

#include"func.h"

#define lenght 256

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Введите строку с правильным математическим выражением."<<endl;

cout << "Пример: M(m(4, 8), 9)"<<endl;

cout<<"'M' - поиск наибольшего среди двух цифр." << endl;

cout << "'m' - поиск наименьшего среди двух цифр." << endl;

char string[lenght];

gets\_s(string);

cout<<"Результат = "<<processing(string, strlen(string));

return 0;

}

Файл func.h

#pragma once

#include<iostream>

using namespace std;

void push(char x);

char pop();

int processing(char \*string, int lenght);

int min(int, int);

int max(int, int);

Файл func.cpp

#include"func.h"

struct Stack

{

char data;

Stack\* link;

};

struct Stack\* top;

/// <summary>

/// Пошук меншого числа

/// </summary>

/// <param name="a">число 1</param>

/// <param name="b">число 2</param>

/// <returns>менше число або 0, якщо числа рівні</returns>

int min(int a, int b)

{

if (a > b)

return b;

else if (b > a)

return a;

else

return 0;

}

/// <summary>

/// Пошук більшого числа

/// </summary>

/// <param name="a">число 1</param>

/// <param name="b">число 2</param>

/// <returns>більше число або 0, якщо числа рівні</returns>

int max(int a, int b)

{

if (a > b)

return a;

else if (b > a)

return b;

else

return 0;

}

/// <summary>

/// Помістити у стек

/// </summary>

/// <param name="x">Символ, який треба помістити у стек</param>

void push(char x)

{

Stack\* NewPointer = new Stack;

if (!NewPointer)

{

cout << "Ошибка, стек переполнен!" << endl;

exit(1);

}

NewPointer->data = x;

NewPointer->link = top;

top = NewPointer;

}

/// <summary>

/// Витягування зі стеку

/// </summary>

/// <returns></returns>

char pop()

{

Stack\* temp;

if (top == NULL)

{

cout << "Ошибка, стек пуст!" << endl;

exit(1);

}

char x;

temp = top;

top = top->link;

x = temp->data;

temp->link = NULL;

free(temp);

return x;

}

/// <summary>

/// Обробка приклада в рядку

/// </summary>

/// <param name="string">рядок</param>

/// <param name="lenght">довжина рядка</param>

/// <returns>Результат підрахунку</returns>

int processing(char \*string, int lenght)

{

int result = 0;

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

{

char choise = string[i];

switch (choise)

{

case 'M':

{

push(choise);

break;

}

case 'm':

{

push(choise);

break;

}

case '(':

{

break;

}

case ',':

{

break;

}

case ' ':

{

break;

}

case ')':

{

int a, b;

char operation;

a = pop() - '0';

b = pop() - '0';

operation = pop();

if (operation == 'M')

result = max(a, b);

else

result = min(a, b);

push(result + '0');

break;

}

default:

{

push(choise);

}

}

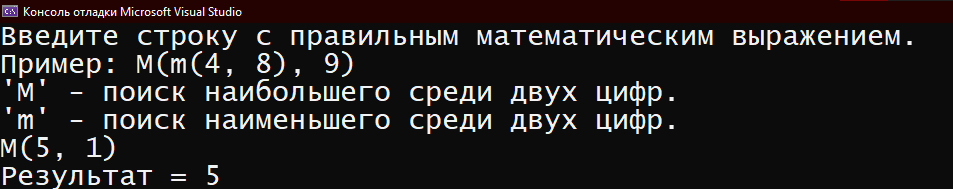
}

return result;

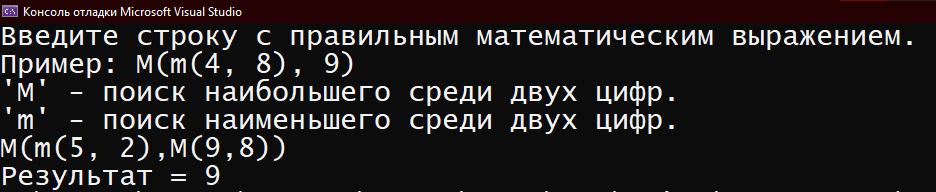
}

**7. Результати тестувань**

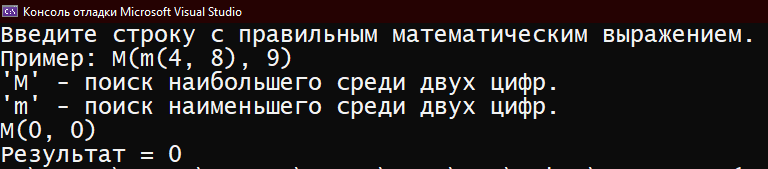
7.1 Одна операція



7.2 Вкладені операції



7.3 Нулі в параметрах



Програма налагоджена і працює вірно.

**8. Висновок**

Під час виконання лабораторної лабораторної роботи познайомився з лінійними динамічними структурами. Це структури, які мають інформаційні поля та поле з адресом наступного елементу. Динамічні структури доцільно використовувати, коли ви не знаєте скільки пам'яті треба виділяти під зберігання даних. Вони не потребують неперервної області пам'яті як масиви, проте якщо даних багато то доступ до них буде відбуватися повільніше. Але навідміну від масивів в структурі можна добавляти та видаляти елементи з середини списку. Динамічні структури бувають таких типів як однозв'язний та двозв'язний список, стек, черга та дек.