

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Стеки и очереди**

**Студент гр. 7381**

**Кревчик А.Б.**

**Преподаватель**

**Фирсов М. А.**

**Санкт-Петербург**

**2018**

## **Цель работы.**

Ознакомиться с такими структурами данных, как стек и очередь, и научиться применять их на практике.

## **Задание.**

*Вариант 7:*

в заданном текстовом файле F записана формула вида

$$\begin{aligned} \langle \text{формула} \rangle ::= \langle \text{цифра} \rangle \mid M ( \langle \text{формула} \rangle , \langle \text{формула} \rangle ) \mid \\ m ( \langle \text{формула} \rangle , \langle \text{формула} \rangle ) \end{aligned}$$

$$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid \dots \mid 9$$

где M обозначает функцию max, а m – функцию min. Вычислить (как целое число) значение данной формулы. Например,  $M(5, m(6, 8)) = 6$ .

## **Основные теоретические положения.**

Стек - это структура данных, в которой хранятся элементы в виде последовательности, организованной по принципу LIFO (Last In — First Out). Такую структуру данных можно сравнить со стопкой тарелок или магазином автомата. Стек не предполагает прямого доступа к элементам и список основных операций ограничивается операциями помещения элемента в стек и извлечения элемента из стека. Их принято называть PUSH и POP соответственно. Также, обычно есть возможность посмотреть на верхний элемент стека не извлекая его (TOP) и несколько других функций, таких как проверка на пустоту стека и некоторые другие. Очередь - эта структура данных, в которой хранятся элементы в виде последовательности, организованной по принципу FIFO (First In — First Out). Эта структура данных более естественна - например, очередь в магазине. Также как и стек, очередь не предполагает прямого доступа к

элементам, а основные операции: добавление ENQ (enqueue) и извлечение DEQ(dequeue). Также обычно есть функции получения первого элемента без его извлечения, определения размера очереди, проверки на пустоту и некоторые другие.

### **Описание алгоритма.**

В программе происходит считывание строки. Если символ цифра, то она кладется в стек. Если символом является «М» или «m», то выполняется соответствующая операция. Параллельно строка проверяется на корректность. По окончании работы программы на экран выводится либо результат вычислений, либо сообщение о некорректности строки.

### **Описание функций.**

*Функция **int main()***

В главной функции производится считывание строки и создание объекта структуры типа *Array\_stack*.

*Функция **int F(char \*str,int \*i, Array\_stack \* S)***

***Рекурсивная функция***

*Аргументы:*

Указатель на строку, указатель на индекс рассматриваемого элемента строки и указатель на стек.

*Возвращаемое значение:*

0/1

*Описание:*

На вход функция получает указатель на строку и указатель на индекс элемента в этой строке. Далее выполняется проверка элемента. Если элемент цифра, то он кладется в стек. Функция возвращает значение 0.

Если символ «M», то, учитывая, что у параллельно строка проверяется на корректность, мы проверяем, является ли следующий символ открывающей скобкой. Если это так, то мы увеличиваем индекс на 2 и смотрим первый аргумент формулы. Для этого мы рекурсивно вызываем функцию *F*. Если функция вернула значение 0 (т.е. первый аргумент корректен), тогда мы проверяем, стоит ли после него запятая. Если это так, то мы увеличиваем индекс на 2 и проверяем второй аргумент формулы и если он также корректен, проверяем скобку, закрывающую формулу. Если формула верна, то мы достаем из стека два верхних элемента, вычисляем максимальный из них и кладем обратно в стек. Аналогичные действия если символ «m».

Если же на каком-то этапе проверки формула оказалась некорректна, функция возвращает -1.

Также на экран выводятся сообщения о том, что положено в стек, что достали из стека.

### **Тестирование программы.**

Файлы с тестовыми данными, находящиеся в директории **Tests** и имеющие названия вида **Tests.txt** ( $1 \leq n \leq 6$ ), проверяют функционал и работоспособность написанной программы. Корректность работы программы протестирована как на пустом списке, так и на списках различной глубины и содержания.

Входные данные	Результат работы программы
*пустая строка*	Строка некорректна!
M(M(7,8),m(6,4))	pushed:7

	<p>pushed:8</p> <p>poped:8</p> <p>poped:7</p> <p>pushed:8</p> <p>pushed:6</p> <p>pushed:4</p> <p>poped:4</p> <p>poped:6</p> <p>pushed:4</p> <p>poped:4</p> <p>poped:8</p> <p>pushed:8</p> <p>Результат: 8</p>
M(7, 9)	<p>pushed:7</p> <p>Строка некорректна!</p>
M(y,9)	Строка некорректна!
M(8,89)	<p>pushed:8</p> <p>pushed:8</p> <p>Строка некорректна!</p>
M(m(8,M(8,4)),M(4,3))	<p>pushed:8</p> <p>pushed:8</p> <p>pushed:4</p> <p>poped:4</p> <p>poped:8</p> <p>pushed:8</p> <p>poped:8</p> <p>poped:8</p> <p>pushed:8</p>

	<div>pushed:4</div> <div>pushed:3</div> <div>popped:3</div> <div>popped:4</div> <div>pushed:4</div> <div>popped:4</div> <div>popped:8</div> <div>pushed:8</div> <div>Результат: 8</div>
--	---

### **Выводы.**

В ходе выполнения работы была изучена новая структура данных: стек.

**ПРИЛОЖЕНИЕ А**  
**ИСХОДНЫЙ КОД ПРОГРАММЫ**

```
#include <iostream>
#include <fstream>
#include <cstdlib>
//#include <windows.h>
#include "stack.h"
#include <cstring>
using namespace std;
//M(8,m(8,5))

int F(char *str,int *i, Array_stack * S){
    int a,b,res;

    if ((str[*i] >= '0') && (str[*i] <= '9') ) {
        //если цифра от 0 до 9
        S->push(str[*i]);
        cout << "pushed:" << str[*i] << endl;
        //*i=*(i)+2;
        return 0;
    }

    if ((str[*i] == 'M') && (str[*i+1] == '(') ) {
        *i=(*i)+2;                //если М и за ней (
        if((F(str, i, S) == 0) && (str[*i+1] == ',')) {
            //ищем первый аргумент
            *i=(*i)+2;
            if((F(str, i, S) == 0) && (str[*i+1] == ' ')){
                a = S->pop();
                cout << "poped:" << a - 48 << endl;
                b = S->pop();
                cout << "poped:" << b - 48 << endl;
                res = max(a,b);
            }
        }
    }
}
```

```

        S->push(res);
        cout << "pushed:" << res - 48 << endl;
        *i=(*i)+1;
        //S->push(max(S->pop(), S->pop()));
        return 0;
    }
}

if ((str[*i] == 'm') && (str[*i+1] == '(') ) {
    *i=(*i)+2;
    //если m
и за ней (
    if((F(str, i, S) == 0) && (str[*i+1] == ',')) {
        //ищем первый аргумент
        *i=(*i)+2;
        if((F(str, i, S) == 0) && (str[*i+1] == ' ')){
            a = S->pop();
            cout << "poped:" << a - 48 << endl;
            b = S->pop();
            cout << "poped:" << b - 48 << endl;
            res = min(a,b);
            S->push(res);
            cout << "pushed:" << res - 48 << endl;
            *i=(*i)+1;
            //S->push(min(S->pop(), S->pop()));
            return 0;
        }
    }
}

return (-1);
}

int main(){
    //SetConsoleCP(1251);

```



```

//SetConsoleOutputCP(1251);

char ch;
//char str[30];
int i = 0;
int size = 10;
char *str = new char [size];
    cout << "Ведите формулу вида " << endl;
    cout << "<формула> ::= <цифра> | M(<формула>,<формула>) |
m(<формула>,<формула>) " << endl;
    cout << "<цифра> ::= 0|1|...|9 " << endl;

Array_stack S;

while(1){
    std::cin >> std::noskipws >> ch;

    if(i+1 == size){
        str[i]='\0';
        char *copy = new char [size];
        strncpy(copy,str,size);
        delete []str;
        size*=2;
        str = new char [size];
        strncpy(str,copy,size/2);
        delete []copy;
    }
    if(ch == '\n'){
        str[i]='\0';
        break;
    }
    str[i]=ch;
    i++;
}

```

```
}
```

```
i=0;
```

```
if(F(str, &i, &S) == 0)
```

```
    cout << "Результат: " << S.pop() << endl;
```

```
else
```

```
    cout << "Строка некорректна! ";
```

```
delete []str;
```

```
return 0;
```

```
}
```