|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования FPMI_ngtu_neti_rgb_polya«Новосибирский государственный технический университет» | | | |
|  | | | |
| Кафедра прикладной математики | | | |
| Практическое задание № 2 | | | |
| по дисциплине «Структуры данных и алгоритмы» | | | |
| **«Стек и очередь»** | | | |
|  | | | |
|  | Группа | ПМ-21 | |
| Бригада 2 | Егупов иван | |
|  | Щербакова Мария | |
| Сагыдыев Данияр | |
|  | |
|  | |
| Преподаватель | ХИЦЕНКО ВАЛЕНТИНА ПАВЛОВНА | |
| Дата | 31.03 | |
| Новосибирск,2023 | | | |

1. **Условие задачи**

В текстовом файле записано без ошибок логическое выражение (ЛВ) в следующей форме:

<ЛВ>::= t | f | (!<ЛВ>) | (<ЛВ>&<ЛВ>) | (<ЛВ>|<ЛВ>)

Здесь знаки !,&,| обозначают соотвественно отрицание, конъюнкцию и дизъюнкцию, а t – значение истина, f – значение ложь. Вычислить логическое значение этого выражения, используя статический стек.

1. **Анализ задачи**

Входные данные: Логическое выражение S ∈ ЛВ

<ЛВ>::= t | f | (!<ЛВ>) | (<ЛВ>&<ЛВ>) | (<ЛВ>|<ЛВ>)

Выходные данные: k ∈ {t;f}

Решение:

**при** x = 1-й символ S, G = пустой стек

**повторять** если x ≠ ‘(‘ и x ≠ ‘)’ , то положить x в стек, x = следующий символ S

иначе если x = ‘)’, то y = взять элемент из G, z = взять элемент из G

если z ≠ ‘!’, то p = взять элемент из G,

если z = ‘&’, то положить в G (y И p),x = след.

иначе положить в G (y ИЛИ p ), x = след.

иначе положить в G (НЕ y), x = след.

**пока** не конец S

Подзадачи:

result – Вычисление значения S

Входные данные: S

Выходные данные: t или f

check – перевод символа из S в логическое значение

Входные данные: символ из S

Выходные данные: логическое значение символа

solv – перевод логичесого значения в соответсвующий символ логического выражения

Входные данные: логическое значение

Выходные данные: символ ∈ ЛВ

Основые подзадачи работы со стеком:

create – Создание стека

Входные данные: информация о стеке

Isempty – проверка стека на пустоту

Входные данные: информация о стеке

Push – положить новый элемент в вершину стека

Входные данные: новый элемент и ифнормация о стеке

Pop – достать элемент из стека

Входные данные: информация о стеке

Выходные данные: элемент, взятый из вершины стека

Stkprint – вывести содержимое стека на экран

Входные данные: информация о стеке

Выходные данные: содержимое стека

1. **Структуры данных**

Внешнее представление   
Входные данные: Логическое выражение в заданной форме

Внутреннее представление: Статический стек

Top 0 K n - 1

Внутреннее представление элементов стека представлено в виде структуры

struct stack

{

int top;

char data[n];

};

1. **Алгоритм решения задачи**



Ввод S

x = 1-й символ S

G = пустой стек

Конец S?

**да**

нет

x = ‘)’ ?

x ≠ ‘(‘ и x ≠ ‘)’?

нет

да

да

k = взять из g

y = взять из G

z = взять из G

Положить x в G, x = следующий

x ≠ ‘!’ ?

Вывод k

Положить в G (НЕ x)  
x = следующий

нет

да

p = взять из G

z = ‘&’ ?

Положить в G (y И p)  
x = следующий

да

Положить в G (y ИЛИ p)  
x = следующий

нет

5.Структура программы

void create(stack\* s) назначение: инициализоровать поле top стека равным 0

stack \*s – указатель на стек

int isempty (stack\* stk) назначение: проверить стек на пустоту

stack \*s – указатель на стек

Возвращет 1, если стек пуст, иначе 0

void push(char c, stack \*stk) назначение:положить элемент в вершину стека

char c – элемент, который необходимо положить

stack \*stk – указатель на стек

char pop(stack \*stk) назначение: достать элемент из вершины стека

stack \*stk – указатель на стек

Возвращает элемент из вершины стека. Если стек пуст, то информирует об этом и возвращает 0

void stkPrint(stack\* stk) назначение: вывод содержимого стека на экран

stack \*stk – указатель на стек

bool check(char c) назначение: перевод символа в логическое значение

char c – элемент, логическое значение которого нужно определить

Возвращает Истину или Ложь

char solv(bool p) назначение: перевод логического значения в соответсвующий ему символ

bool p – логическое значение, необходимое перевести

Возвращает ‘t’, если логическое значение Истина, иначе ‘f’

char result(FILE \* file) назначение: Вычисление логического значения выражения, записанного в файле

FILE \*file – указатель на файл, содержащий логическое выражение

Возвращает ‘t’, если значение Истина, иначе ‘f’

6.Текст программы

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define n 200

struct stack

{

int stack\_size;

char data[n];

};

void create(stack\* s)

{

s->stack\_size = 0;

}

int isempty (stack\* stk) {

if (stk->stack\_size == 0) return 1;

else return 0;

}

void push(char ch, stack\* stk)

{

stk->data[stk->stack\_size] = ch;

stk->stack\_size++;

}

char top(stack\* stk)

{

return stk->data[stk->stack\_size - 1];

}

char pop(stack\* st)

{

char ch = top(st);

st->stack\_size--;

return ch;

}

void stkPrint(stack\* stk) {

int i;

i = stk->stack\_size;

if (isempty(stk) == 1) return;

do {

i--;

printf("%c\n", stk->data[i]);

} while (i > 0);

}

bool check(char c)

{

return c == 't' ? true : false;

}

char solv(bool p)

{

return p == true ? 't' : 'f';

}

char result(FILE\* file)

{

char res = 'f';

stack g;

create(&g);

bool bool\_res;

char x;

while ((x = fgetc(file)) != EOF)

{

if (x == ' ' || x == '(')

continue;

if (x == ')')

{

char operand2 = pop(&g);

char operatoR = pop(&g);

if (operatoR == '!')

{

bool\_res = !check(operand2);

}

else

{

char operand1 = pop(&g);

if (operatoR == '&')

bool\_res = check(operand1) && check(operand2);

else

bool\_res = check(operand1) || check(operand2);

}

push(solv(bool\_res), &g);

}

else

{

push(x, &g);

}

}

res = pop(&g);

return res;

}

int main()

{

FILE\* write,\*read;

fopen\_s(&read, "input.txt", "r");

char r = result(read);

fclose(read);

fopen\_s(&write, "output.txt", "w");

fprintf\_s(write, "%c",r);

fclose(write);

return 0;

}

7.Набор тестов

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | назначение |
| S = t | k = t | Простейший тест с вводом t |
| S = f | k = f | Простейший тест с вводом f |
| S = (!f) | k = t | Тест с отрицанием |
| S = (t&t) | k = t | На вход поступило выражение с конъюнкцией |
| S = (f|t) | k = t | На вход поступило выражение с дизъюнкцией |
| S = (!(f&f)) | k = t | Отрицание коъюнкции |
| S = (!(t|f)) | k = f | Отрицание дизъюнкции |
| S = ((f|t)&(t&t)) | k = t | Тест с вложенными оператоми |
| S = ((t|(!f))&(!f)) | k = t | Тест с вложенными отрицаниями |
| S = ((!(t&f))|(!(f|t))) | k = t | Тест в отрицанием результатов, вложенных операторов |
| S = (((t|f)&f)|((!(t|t))&(t|f))) | k = f | Тест с большим числом вложенных операторов |
| S = ((!((t|f)&f))|((!(t|t))&(t|f)))|f) | k = t | Отрицание выражения большой длины и последующая дизъюнкция |