1 ​|t|<tα/2（99）

#include<iostream>

#include<string>

#include<vector>

#include<stack>

#include<algorithm>

using namespace std;

void show()

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*逻辑运算\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " 1——逻辑与运算" << endl;

cout << " 2——逻辑或运算" << endl;

cout << " 3——逻辑非运算" << endl;

cout << " 4——逻辑异或运算" << endl;

cout << " 5——逻辑单条件运算" << endl;

cout << " 6——逻辑双条件运算" << endl;

cout << " 7——逻辑常元表达式运算" << endl;

cout << " 8——逻辑变元表达式运算" << endl;

cout << " 0——程序退出" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl;

}

void menu1()

{

cout << "----------------使用说明----------------" << endl;

cout << "========================================" << endl;

cout << "== true用'1' false用'0' ==" << endl;

cout << "== 命题变元字母不能重复且数量小于8 ==" << endl;

cout << "== 逻辑变元用单个字母表示 ==" << endl;

cout << "== 与逻辑用'\*' 或逻辑用'+' ==" << endl;

cout << "== 非逻辑用'!' 异或逻辑用'^' ==" << endl;

cout << "== 单条件用'>' 双条件用'=' ==" << endl;

cout << "=======================================" << endl;

}

void menu2()

{

cout << "----------------使用说明----------------" << endl;

cout << "========================================" << endl;

cout << "== 命题变元所用字母不能重复 ==" << endl;

cout << "== 且变元个数不能超过8个 ==" << endl;

cout << "== 逻辑变元用单个字母表示 ==" << endl;

cout << "== 与逻辑用'\*' 或逻辑用'+' ==" << endl;

cout << "== 非逻辑用'!' 异或逻辑用'^' ==" << endl;

cout << "== 单条件用'>' 双条件用'=' ==" << endl;

cout << "========================================" << endl;

}

stack<bool>opr;

stack<char>opr1;

stack<char>cal;

bool calculate()

{

bool a = 0, b = 0; char c = '0';

c = cal.top(); cal.pop();

if (c != '!')

{

b = opr.top(); opr.pop();

}

a = opr.top(); opr.pop();

switch (c)

{

case '!':

return !a;

case'\*':

return a && b;

case'+':

return a || b;

case'>':

{

if (a == 1 && b == 0)

return 0;

else return 1;

}

case'=':

return a == b;

}

}

int priority(const char ch)//符号优先级判断

{

//¬, ∧, ∨, →, ↔

switch (ch)

{

case'(':

return 0;

case'^':

return 1;

case'=':

return 2;

case'>':

return 3;

case'+':

return 4;

case'\*':

return 5;

case'!':

return 6;

case')':

return 7;

}

}

bool pack(string str1)

{

char ch = '0'; int lv1 = 0, lv2 = 0; bool r = 0, result = 1;

int i = 0, len = str1.size() - 1;//读到最后一个

while (i < len)//扫描式子

{

if (str1[i] == '0' || str1[i] == '1')

{

if (str1[i] == '0')

{

r = 0;

opr.push(r);

}

else

{

r = 1;

opr.push(r);

}

i++;

}

else

{

if (cal.empty())

{

cal.push(str1[i]);

i++;

continue;

}

else

{

ch = cal.top();

lv1 = priority(ch);

lv2 = priority(str1[i]);

if (lv2 == 0)

{

cal.push(str1[i]);

i++;

}

else if (lv2 == 7)

{

ch = cal.top();

while (ch != '(')

{

result = calculate();

opr.push(result);

ch = cal.top();

}

cal.pop();//弹出左括号

i++;

}

else

{

while (!cal.empty() && lv2 <= lv1)

{

result = calculate();

opr.push(result);

if (!cal.empty())

{

ch = cal.top();

lv1 = priority(ch);

}

}

cal.push(str1[i]);

i++;

}

}

}

}

while (!cal.empty())//当把式子读完且操作符栈仍非空时

{

result = calculate();//计算

opr.push(result);

}

if (cal.empty())

return result;

}

void case1(bool P, bool Q, bool result)

{

cout << "请输入真值value1和value2：";

cin >> P >> Q;

result = P && Q;

cout << "逻辑与结果为：" << result << endl;

cout << endl;

}

void case2(bool P, bool Q, bool result)

{

cout << "请输入真值value1和value2：";

cin >> P >> Q;

result = P || Q;

cout << "逻辑或结果为：" << result << endl;

cout << endl;

}

void case3(bool P, bool result)

{

cout << "请输入真值value1：";

cin >> P;

result = !P;

cout << "逻辑非结果为：" << result << endl;

cout << endl;

}

void case4(bool P, bool Q, bool result)

{

cout << "请输入真值value1和value2：";

cin >> P >> Q;

result = P ^ Q;

cout << "逻辑异或结果为：" << result << endl;

cout << endl;

}

void case5(bool P, bool Q, bool result)

{

cout << "请输入真值value1和value2：";

cin >> P >> Q;

if (P == 1 && Q == 0)

result = 0;

cout << "逻辑单条件结果为：" << result << endl;

cout << endl;

}

void case6(bool P, bool Q, bool result)

{

cout << "请输入真值value1和value2：";

cin >> P >> Q;

if (P != Q)

result = 0;

cout << "逻辑双条件结果为：" << result << endl;

cout << endl;

}

void case7()

{

string str1;

cout << "请输入表达式<以#键结束>：";

cin >> str1;

cout << "运算结果为：" << pack(str1) << endl;

cout << endl;

}

void case8()

{

string s;

cout << "请输入表达式<以#键结束>：";

cin >> s;

s.pop\_back();

vector<char> letter, disjunctive, conjunctive;

//letter存放字母

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

{

if ((s[i] > 64 && s[i] < 91) || (s[i] > 96 && s[i] < 123))

{//大小写26个字母

letter.push\_back(s[i]);

}

}

//真值表

cout << "真值表为：" << endl;

cout << "————————————————————" << endl;

for (int i = 0; i < letter.size(); i++)

cout << letter[i] << " ";

cout << s << endl;

cout << "————————————————————" << endl;

vector<char> letters = letter;

int d;

stack<int> stk;

for (int i = 0; i < pow(2, letter.size()); i++)

{

int t = i;

string s1 = s;

for (int j = 0; j < letter.size(); j++)

{

d = t % 2;

t = t / 2;

stk.push(d);

}

for (int k = 0; k < letter.size(); k++)

{

int top = stk.top();

cout << top << " ";

s1[s1.find(letter[k])] = top + 48;//转成char型的0和1

stk.pop();

}

s1.push\_back('#');

bool c = pack(s1);

cout << c << endl;

//主析取范式

if (c == true)

{

int num = 0;

disjunctive.push\_back('(');

for (int i = 0; i < s1.size(); i++)

{

if (s1[i] == 48) //真值为0，加!

{

disjunctive.push\_back('!');

disjunctive.push\_back(letters[num]);

disjunctive.push\_back('\*');

num++;

}

else if (s1[i] == 49) //真值为1

{

disjunctive.push\_back(letters[num]);

disjunctive.push\_back('\*');

num++;

}

else if (s1[i] == '#')

break;

else

continue;

}

disjunctive.pop\_back();

disjunctive.push\_back(')');

disjunctive.push\_back('+');

}

//主合取范式

else

{

int num = 0;

conjunctive.push\_back('(');

for (int i = 0; i < s1.size(); i++)

{

if (s1[i] == 49)

{//真值为1，加!

conjunctive.push\_back('!');

conjunctive.push\_back(letters[num]);

conjunctive.push\_back('+');

num++;

}

else if (s1[i] == 48)

{//真值为0

conjunctive.push\_back(letters[num]);

conjunctive.push\_back('+');

num++;

}

else if (s1[i] == '#')

break;

else

continue;

}

conjunctive.pop\_back();

conjunctive.push\_back(')');

conjunctive.push\_back('\*');

}

}

disjunctive.pop\_back();

conjunctive.pop\_back();

cout << "————————————————————" << endl;

cout << "主析取范式为：";

for (int i = 0; i < disjunctive.size(); i++)

{

cout << disjunctive[i];

}

cout << endl << "主合取范式为：";

for (int i = 0; i < conjunctive.size(); i++)

{

cout << conjunctive[i];

}

cout << endl; cout << endl;

}

int main()

{

int choice = 0; bool p = 0, q = 0, r = 0, result = 1;

show();

while (true)

{

cout << "请输入您要进行逻辑运算的符号：";

cin >> choice;

switch (choice)

{

case 1:

menu1();

case1(p, q, result);

break;

case 2:

menu1();

case2(p, q, result);

break;

case 3:

menu1();

case3(p, result);

break;

case 4:

menu1();

case4(p, q, result);

break;

case 5:

menu1();

case5(p, q, result);

break;

case 6:

menu1();

case6(p, q, result);

break;

case 7:

case7();

break;

case 8:

menu2();

case8();

break;

case 0:

exit(0);

break;

default:

cout << "输入错误请重新输入!" << endl;

break;

}

}

return 0;

}

2