#include <iostream>

using namespace std;

class BooleanOperations

{

bool x, y;

public:

BooleanOperations(bool a, bool b)

{

x = a;

y = b;

}

bool conjunction()

{

return (x && y);

}

bool disjunction()

{

return (x && y);

}

bool XOR()

{

return (x ^ y);

}

bool conditional()

{

return (x || (!y));

}

bool biconditional()

{

return !(x ^ y);

}

bool XNOR()

{

return !(x ^ y);

}

bool negation()

{

return (!x);

}

bool NAND()

{

return !(x && y);

}

bool NOR()

{

return !(x || y);

}

};

int main()

{

BooleanOperations o1(0, 0);

BooleanOperations o2(0, 1);

BooleanOperations o3(1, 0);

BooleanOperations o4(1, 1);

int choice;

cout << "Truth table Menu" << endl;

cout << "1. conjunction" << endl;

cout << "2. disjunction" << endl;

cout << "3. XOR" << endl;

cout << "4. conditional" << endl;

cout << "5. biconditional" << endl;

cout << "6. XNOR" << endl;

cout << "7. negation" << endl;

cout << "8. NAND" << endl;

cout << "9. NOR" << endl;

cout << "Enter choice: ";

cin >> choice;

switch (choice)

{

case 1:

cout << "Conjunction Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.conjunction() << endl;

cout << "0 1 " << o2.conjunction() << endl;

cout << "1 0 " << o3.conjunction() << endl;

cout << "1 1 " << o4.conjunction() << endl;

break;

case 2:

cout << "Disconjunction Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.disjunction() << endl;

cout << "0 1 " << o2.disjunction() << endl;

cout << "1 0 " << o3.disjunction() << endl;

cout << "1 1 " << o4.disjunction() << endl;

break;

case 3:

cout << "XOR Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.XOR() << endl;

cout << "0 1 " << o2.XOR() << endl;

cout << "1 0 " << o3.XOR() << endl;

cout << "1 1 " << o4.XOR() << endl;

break;

case 4:

cout << "Conditional Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.conditional() << endl;

cout << "0 1 " << o2.conditional() << endl;

cout << "1 0 " << o3.conditional() << endl;

cout << "1 1 " << o4.conditional() << endl;

break;

case 5:

cout << "BiConditional Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.biconditional() << endl;

cout << "0 1 " << o2.biconditional() << endl;

cout << "1 0 " << o3.biconditional() << endl;

cout << "1 1 " << o4.biconditional() << endl;

break;

case 6:

cout << "XNOR Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.XNOR() << endl;

cout << "0 1 " << o2.XNOR() << endl;

cout << "1 0 " << o3.XNOR() << endl;

cout << "1 1 " << o4.XNOR() << endl;

break;

case 7:

cout << "Negation Truth Table" << endl;

cout << "x res" << endl;

cout << "0 " << o1.negation() << endl;

cout << "1 " << o4.negation() << endl;

break;

case 8:

cout << "NAND Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.NAND() << endl;

cout << "0 1 " << o2.NAND() << endl;

cout << "1 0 " << o3.NAND() << endl;

cout << "1 1 " << o4.NAND() << endl;

break;

case 9:

cout << "NOR Truth Table" << endl;

cout << "x y res" << endl;

cout << "0 0 " << o1.NOR() << endl;

cout << "0 1 " << o2.NOR() << endl;

cout << "1 0 " << o3.NOR() << endl;

cout << "1 1 " << o4.NOR() << endl;

break;

default:

cout << "Invalid choice" << endl;

break;

}

return 0;

}

