ЗВІТ

про виконання лабораторної роботи № 1

«Логіка»

з дисципліни

«Програмування дискретних структур»

Студента групи КН-203Б

Ячечака Масима Ігоровича

Умова завдання:

Скласти програми із зазначеними вхідними  даними та результатами.

1. Задано значення істинності висловлювань p та q. Знайти значення істинності кон’юнкції, диз’юнкції, альтернативного «або», імплікації й еквівалентності цих висловлювань.

2. Задано два бітові рядки довжиною n. Знайти результати виконання порозрядних операцій OR, AND, XOR цих рядків.

|  |  |  |
| --- | --- | --- |
| 12. | 8 | 11001110  01100111 |

#include <iostream>

#include <string>

using namespace std;

void true\_false(bool p, bool q) {

bool result = false;

bool result\_xnor\_1 = false;

bool result\_xnor\_2 = false;

bool p\_implication = false;

bool q\_implication = false;

//-------AND(Conjunction)(∧)-------

if (p == q && p == true){

result = true;

}

else {

result = false;

}

cout << "AND(Conjunction)(∧): ";

if (result) {

cout << "True" << endl;

}

else {

cout << "False" << endl;

}

//-------OR(Disjunction)(∨)-------

if (p == q && p == false) {

result = false;

}

else {

result = true;

}

cout << "OR(Disjunction)(∨): ";

if (result) {

cout << "True" << endl;

}

else {

cout << "False" << endl;

}

//-------XOR(Strict Disjunction)(⊕)-------

if (p != q) {

result = true;

}

else {

result = false;

}

cout << "XOR(Strict Disjunction)(⊕): ";

if (result) {

cout << "True" << endl;

}

else {

cout << "False" << endl;

}

//-------NOTOR(Implication)(→)-------

if (p == false) {

p\_implication = true;

}

else {

p\_implication = false;

}

if (p\_implication == q && p\_implication == false) {

result = false;

}

else {

result = true;

}

cout << "NOTOR(Implication)(→): ";

if (result) {

cout << "True" << endl;

}

else {

cout << "False" << endl;

}

//-------NOTOR(Reverse Implication)(←)-------

if (q == false) {

q\_implication = true;

}

else {

q\_implication = false;

}

if (p == q\_implication && p == false) {

result = false;

}

else {

result = true;

}

cout << "NOTOR(Reverse Implication)(←): ";

if (result) {

cout << "True" << endl;

}

else {

cout << "False" << endl;

}

//-------XNOR(Equivalence)(⟷)(~)-------

//NOTOR(Implication)(→)

if (p == false) {

p\_implication = true;

}

else {

p\_implication = false;

}

if (p\_implication == q && p\_implication == false) {

result\_xnor\_1 = false;

}

else {

result\_xnor\_1 = true;

}

//NOTOR(Reverse Implication)(←)

if (q == false) {

q\_implication = true;

}

else {

q\_implication = false;

}

if (p == q\_implication && p == false) {

result\_xnor\_2 = false;

}

else {

result\_xnor\_2 = true;

}

//AND(Conjunction)(∧)

if (result\_xnor\_1 == result\_xnor\_2 && result\_xnor\_1 == true) {

result = true;

}

else {

result = false;

}

cout << "XNOR(Equivalence)(⟷)(~): ";

if (result) {

cout << "True" << endl << endl;

} else {

cout << "False" << endl << endl;

}

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

}

void OR\_AND\_XOR(int arr[], int arr1[])

{

int n = 8;

int result\_or = 0;

int result\_and = 0;

int result\_xor = 0;

cout << "One---Two---OR---AND---XOR" << endl;

for (int i = 0; i < n; i++) {

//-------OR-------

if (arr[i] == arr1[i] && arr[i] == 0) {

result\_or = 0;

}

else {

result\_or = 1;

}

//-------AND-------

if (arr[i] == arr1[i] && arr[i] == 1) {

result\_and = 1;

}

else {

result\_and = 0;

}

//-------XOR-------

if (arr[i] != arr1[i]) {

result\_xor = 1;

}

else {

result\_xor = 0;

}

cout << " " << arr[i] << " | " << arr1[i] << " | " << result\_or << " | " << result\_and << " | " << result\_xor << endl;

}

}

int main()

{

int arr[8] = { 1,1,0,0,1,1,1,0 };

int arr1[8] = { 0,1,1,0,0,1,1,1 };

bool p = false;

bool q = false;

string p\_checker = "";

string q\_checker = "";

cout << "p: (True|False) (true|false) (1|0) : " << endl;

cin >> p\_checker;

if (p\_checker == "true" || p\_checker == "True" || p\_checker == "1") {

p = true;

}

else if (p\_checker == "false" || p\_checker == "False" || p\_checker == "0") {

p = false;

}

cout << "q: (True|False) (true|false) (1|0) : " << endl;

cin >> q\_checker;

if (q\_checker == "true" || q\_checker == "True" || q\_checker == "1") {

q = true;

}

else if (q\_checker == "false" || q\_checker == "False" || q\_checker == "0") {

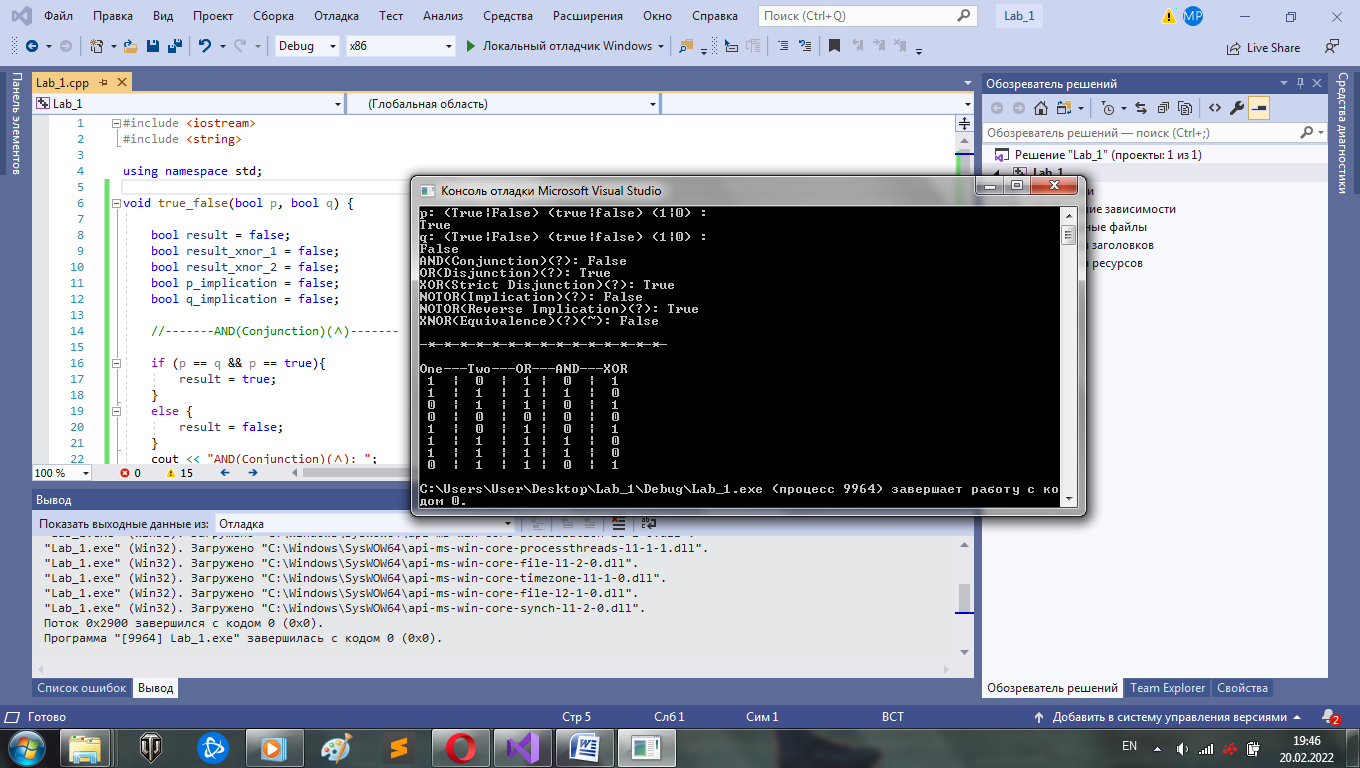
q = false;

}

true\_false(p, q);

OR\_AND\_XOR(arr, arr1);

}



Висновки

Я поглибив і закріпив розуміння теоретичних положень логіки висловлювань, формував навички складання алгоритмів та програм опрацювання логічних величин.

https://github.com/MISTIKMAKS/lab5\_1