ЗВІТ

про виконання лабораторної роботи №2

«Булеві функції»

з дисципліни «Програмування дискретних структур»

Студента групи КН-2327Б

Сухар Станіслав

Мета роботи: Поглибити і закріпити розуміння теоретичних положень розділу «Булеві функції», формувати практичні навички програмування алгоритмів розділу «Булеві функції», розвивати логічне мислення та алгоритмічну культуру.

## Завдання: Скласти програми із зазначеними вхідними даними та результатами.

## Побудувати таблицю істинності для функції ***fn***,заданої формулою (*n* – номер варіанту)

Варіант:



1. Текст програми:

main.cpp

#include <iostream>

using namespace std;

int main() {

cout << "p q r | (p or q) and (p or r)" << endl;

cout << "-----------------------------" << endl;

for (int p = 0; p <= 1; p++) {

for (int q = 0; q <= 1; q++) {

for (int r = 0; r <= 1; r++) {

bool result = (p || q) && (p || r);

cout << p << " " << q << " " << r << " | " << result << endl;

}

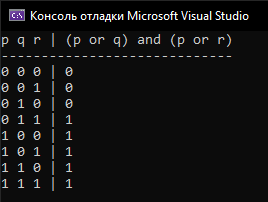
}

}

return 0;

}

1. Результат виконання коду:

****

1. UnitTest

#include "pch.h"

#include "CppUnitTest.h"

#include <bitset>

#include "../lab2/main.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

bool booleanFunction(int p, int q, int r) {

return (p || q) && (p || r);

}

TEST\_METHOD(TestBooleanFunction)

{

Assert::AreEqual(booleanFunction(0, 0, 0), false);

Assert::AreEqual(booleanFunction(0, 0, 1), false);

Assert::AreEqual(booleanFunction(0, 1, 0), false);

Assert::AreEqual(booleanFunction(0, 1, 1), true);

Assert::AreEqual(booleanFunction(1, 0, 0), true);

Assert::AreEqual(booleanFunction(1, 0, 1), true);

Assert::AreEqual(booleanFunction(1, 1, 0), true);

Assert::AreEqual(booleanFunction(1, 1, 1), true);

}

TEST\_METHOD(TestLogicalOperations)

{

bool p, q;

p = true;

q = false;

bool expectedAnd = p && q;

bool expectedOr = p || q;

bool expectedXor = p ^ q;

bool expectedImplication = !p || q;

bool expectedEquivalence = (p && q) || (!p && !q);

Assert::AreEqual(expectedAnd, false);

Assert::AreEqual(expectedOr, true);

Assert::AreEqual(expectedXor, true);

Assert::AreEqual(expectedImplication, false);

Assert::AreEqual(expectedEquivalence, false);

}

TEST\_METHOD(TestBitwiseOperations)

{

std::bitset<8> b1("11111000");

std::bitset<8> b2("10100110");

std::bitset<8> expectedOr = b1 | b2;

std::bitset<8> expectedAnd = b1 & b2;

std::bitset<8> expectedXor = b1 ^ b2;

Assert::AreEqual(expectedOr.to\_ulong(), std::bitset<8>("11111110").to\_ulong());

Assert::AreEqual(expectedAnd.to\_ulong(), std::bitset<8>("10100000").to\_ulong());

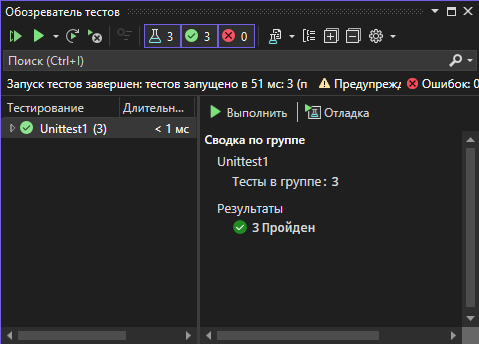
Assert::AreEqual(expectedXor.to\_ulong(), std::bitset<8>("01011110").to\_ulong());

}

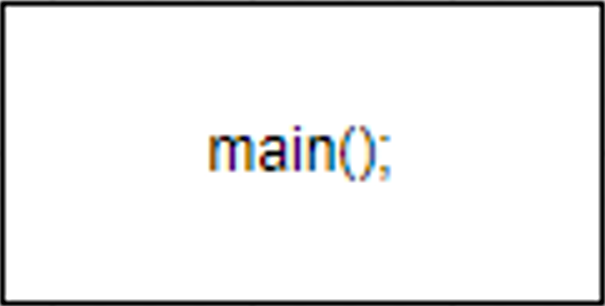
};

}

1. Результат UnitTest:



1. Побудова cтруктурної діаграми:



Посилання на Git-репозиторій:

Висновок:На даній лабораторній роботі, я поглибив і закріпив розуміння теоретичних положень розділу «Булеві функції» і сформував практичні навички програмування алгоритмів розділу «Булеві функції», розвив логічне мислення та алгоритмічну культуру.