ЗВІТ

про виконання лабораторної роботи №1

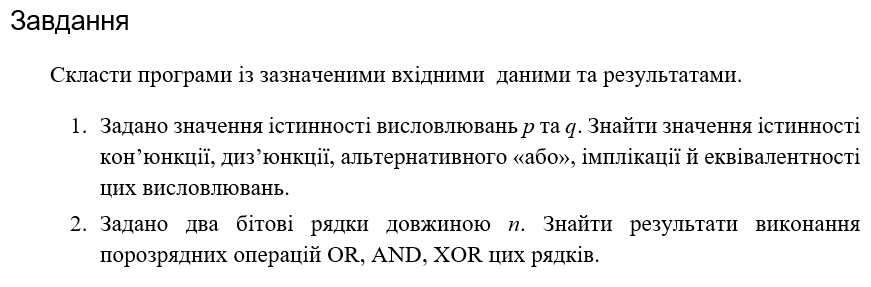
«Логіка»

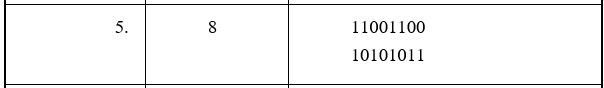
з дисципліни «Програмування дискретних структур»

Студента групи КН-2427Б

Соколовського Назарія

Поглибити і закріпити розуміння теоретичних положень логіки висловлювань, формувати навички складання алгоритмів та програм опрацювання логічних величин.





1. Текст програми:

#include <iostream>

#include <string>

using namespace std;

bool implication(bool p, bool q) {

return !p || q;

}

bool equivalence(bool p, bool q) {

return implication(p, q) && implication(q, p);

}

void bitwise\_operations(const string& a, const string& b,

string& or\_result, string& and\_result, string& xor\_result) {

int n = a.length();

or\_result = "";

and\_result = "";

xor\_result = "";

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

if (a[i] == '1' || b[i] == '1') {

or\_result += '1';

} else {

or\_result += '0';

}

if (a[i] == '1' && b[i] == '1') {

and\_result += '1';

} else {

and\_result += '0';

}

if (a[i] != b[i]) {

xor\_result += '1';

} else {

xor\_result += '0';

}

}

}

int main() {

bool p, q;

cout << "Enter the truth value for p (0 or 1): "; cin >> p;

cout << "Enter the truth value for q (0 or 1): "; cin >> q;

bool conjunction = p && q;

cout << "Conjunction (p and q): " << conjunction << endl;

bool disjunction = p || q;

cout << "Disjunction (p or q): " << disjunction << endl;

bool exclusiveOr = p != q;

cout << "Alternative \"OR\" (p xor q): " << exclusiveOr << endl;

bool impl\_pq = implication(p, q);

cout << "Implication (p -> q): " << impl\_pq << endl;

bool impl\_qp = implication(q, p);

cout << "Implication (q -> p): " << impl\_qp << endl;

bool equiv = equivalence(p, q);

cout << "Equivalence (p ~ q): " << equiv << endl;

string a, b;

cout << "\nEnter the 1 bit string: ";

cin >> a;

cout << "Enter the 2 bit string: ";

cin >> b;

if (a.length() != 8 || b.length() != 8) {

cout << "Error: Strings must be exactly 8 characters long"

<< endl;

return 1;

}

string or\_result, and\_result, xor\_result;

bitwise\_operations(a, b, or\_result, and\_result, xor\_result);

cout << "Result or: " << or\_result << endl;

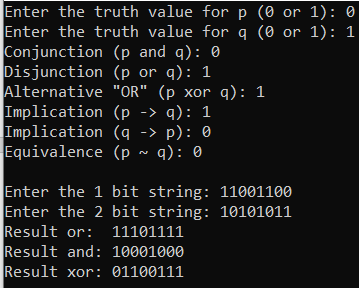
cout << "Result and: " << and\_result << endl;

cout << "Result xor: " << xor\_result << endl;

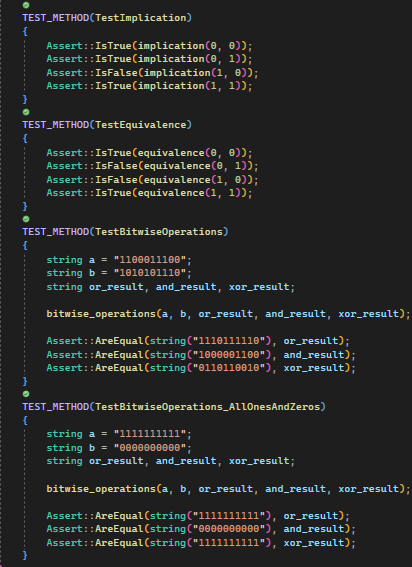
return 0;

}

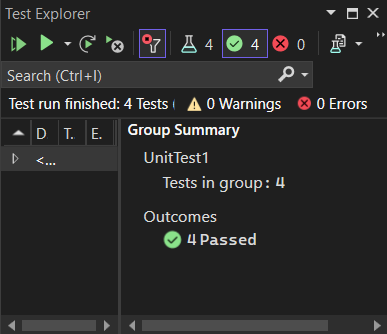
1. Результат виконання коду:



1. UnitTest завдання:
2. Код UnitTest:



1. Результат UnitTest:



1. Побудова Структурної діаграми:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Посилання на Git-репозиторій:

Висновок:На даній лабораторній роботі, я поглибив і закріпив розуміння теоретичних положень логіки висловлювань, сформував навички складання алгоритмів та програм опрацювання логічних величин.