# PROIECT NR.1:TABELE DE JUSTIȚIE

A elaborat: Dmitri COJOCARI, gr. IS11Z

A verificat: Vitalie ȚÎCĂU, lector universitar

## 1. Formularea problemei

1. De trasat tabelele operațiilor:

1. negația;
2. disjuncția;
3. conjuncția;
4. diferența;
5. diferența simetrică.

## 2. Studiu preliminar

Tabelele justeții au fost făcute în C++. Fiecărei operații din punctul anterior îi corespunde un subprogram ce îi trasează tabelul de adevăr. Întrucât nu avem operatori expliciți pentru fiecare dintre operațiile de mai sus, am utilizat operatorii din C++ (!- negația, && - conjuncția, || - disjuncția, ^ - xor (diferența simetrică)). C++ nu are un operator explicit pentru diferență, de aceea am utilizat identitatea: .

## 3. Programul

#include <iostream>

#include <stdio.h>

using namespace std;

void LinieTabel(int nr\_col, char left, char lin, char mid, char right);

void Negatie();

void Disjunctie();

void Conjunctie();

void Diferenta();

void DiferentaSimetrica();

int main()

{

cout << "Proiect nr.1, elaborat de Cojocari Dmitri" << endl;

cout << "Tema: Tabelul justetei" << endl;

Negatie();

Disjunctie();

Conjunctie();

Diferenta();

DiferentaSimetrica();

return 0;

}

void LinieTabel(int nr\_col, char left, char lin, char mid, char right)

{

int i, j;

printf("\n%c", left);

for (j = 1; j < nr\_col; j++)

{

for (i = 1; i <= 3; i++)

printf("%c", lin);

printf("%c", mid);

}

for (i = 1; i <= 9; i++)

printf("%c", lin);

printf("%c", right);

}

void Negatie()

{

cout << "\n\tNegatia (complementara):" << endl;

LinieTabel(2, 201, 205, 209, 187);

printf("\n%c a %c%c not a %2c", 186, 179, 32, 186);

LinieTabel(2, 199, 196, 197, 182);

for (int a = 0; a <= 1; a++)

{

printf("\n%c %d %c %4d %4c", 186, a, 179, !a, 186);

}

LinieTabel(2, 200, 205, 207, 188);

}

void Disjunctie()

{

cout << "\n\tDisjunctia (adunarea):\n";

LinieTabel(3, 201, 205, 209, 187);

printf("\n%c a %c b %c a or b %2c", 186, 179, 179, 186);

LinieTabel(3, 199, 196, 197, 182);

for (int a = 0; a <= 1; a++)

for (int b = 0; b <= 1; b++)

{

printf("\n%c %d %c %d %c %4d %4c", 186, a, 179, b, 179, a || b, 186);

}

LinieTabel(3, 200, 205, 207, 188);

}

void Conjunctie()

{

cout << "\n\tConjunctia (inmultirea):\n";

LinieTabel(3, 201, 205, 209, 187);

printf("\n%c a %c b %c a and b%c%c", 186, 179, 179, 32, 186);

LinieTabel(3, 199, 196, 197, 182);

for (int a = 0; a <= 1; a++)

for (int b = 0; b <= 1; b++)

{

printf("\n%c %d %c %d %c %4d %4c", 186, a, 179, b, 179, a && b, 186);

}

LinieTabel(3, 200, 205, 207, 188);

}

void Diferenta()

{

cout << "\n\tDiferenta :\n";

LinieTabel(3, 201, 205, 209, 187);

printf("\n%c a %c b %c a - b%c%2c", 186, 179, 179, 32, 186);

LinieTabel(3, 199, 196, 197, 182);

for (int a = 0; a <= 1; a++)

for (int b = 0; b <= 1; b++)

{

printf("\n%c %d %c %d %c %4d %4c", 186, a, 179, b, 179, a && !b, 186);

}

LinieTabel(3, 200, 205, 207, 188);

}

void DiferentaSimetrica()

{

cout << "\n\tXor (diferenta simetrica):\n";

LinieTabel(3, 201, 205, 209, 187);

printf("\n%c a %c b %c a xor b%c%c", 186, 179, 179, 32, 186);

LinieTabel(3, 199, 196, 197, 182);

for (int a = 0; a <= 1; a++)

for (int b = 0; b <= 1; b++)

{

printf("\n%c %d %c %d %c %4d %4c", 186, a, 179, b, 179, a ^ b, 186);

}

LinieTabel(3, 200, 205, 207, 188);

}

## 4. Rezultatele si analiza lor

Proiect nr.1, elaborat de Cojocari Dmitri

Tema: Tabelul justetei

Negatia (complementara):

|  |  |
| --- | --- |
| a | not a |
| 0 | 1 |
| 1 | 0 |

Disjunctia (adunarea):

|  |  |  |
| --- | --- | --- |
| a | b | a or b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Conjunctia (inmultirea):

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Diferenta :

|  |  |  |
| --- | --- | --- |
| a | b | a - b |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Xor (diferenta simetrica):

|  |  |  |
| --- | --- | --- |
| a | b | a xor b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## 5. Concluzii

În urma realizării prezentei lucrări, am observat că:

* Operatorul negație este unar,
* Disjuncția are valoarea True, dacă cel puțin un operand este True,
* Conjuncția are valoarea True, dacă și numai dacă toți operanzii au valoarea True,
* Diferența are valoarea True, dacă și numai dacă primul operand are valoarea True și restul operanzilor au valoarea False,
* Diferența simetrică are valoarea True, dacă și numai dacă operanzii au valori diferite.