МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Алгоритмы дискретной математики

Лабораторная работа №3

по дисциплине «Алгоритмы дискретной математики»

Выполнил студент Пшеничный Д.О.

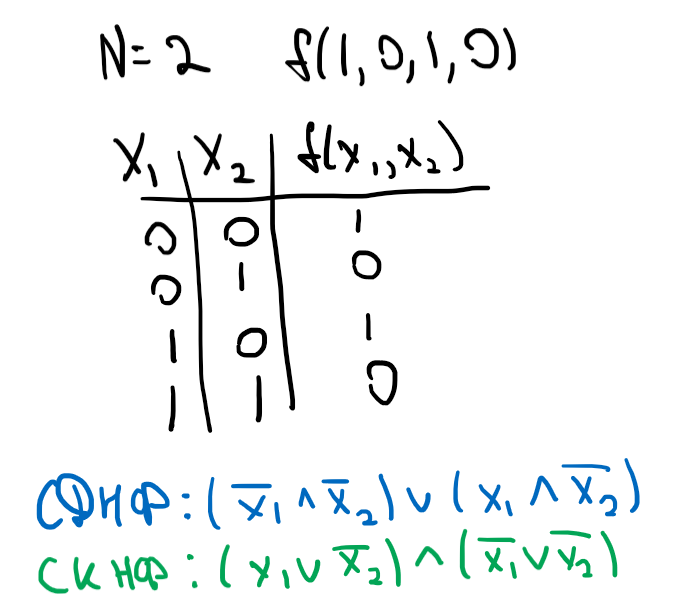
Факультет, группа ФКФН, ПО(аб)-81

Руководитель работы  **Резак Е.В.**

Хабаровск – 2020г.

**Исходная информация**: Логическая функция 𝑓(𝑥1, 𝑥2, … , 𝑥𝑛) задана таблицей истинности. Требуется построить СДНФ и СКНФ функции f. Исходные данные: n – число переменных, от которых зависит функция; f – массив значений функции на различных наборах, упорядоченных по возрастанию.

**Ручной расчет**



**Листинг**

using System;

using System.Collections.Generic;

namespace Lab3

{

class Program

{

static void Main(string[] args)

{

int N;

List<int> funcResult = new List<int>();

string sdnf = "СДНФ ";

string sknf = "СКНФ ";

Console.Write("Введите число переменных: ");

N = int.Parse(Console.ReadLine());

for(int i = 0; i < Math.Pow(2, N); i++)

{

Console.Write("Значение " + (i + 1).ToString() + " строки таблицы истинности\nВведите 1 или 0: ");

while(true)

{

int value = int.Parse(Console.ReadLine());

if(value == 1 || value == 0)

{

funcResult.Add(value);

break;

}

else

{

Console.Write("Введите 1 или 0: ");

}

}

}

PrintLogTable(N, funcResult);

for(int i = 0; i < Math.Pow(2, N); i++)

{

string func = ConvertToBinary(i).ToString();

while(func.Length != N)

{

func = func.Insert(0, "0");

}

if(funcResult[i] == 1)

{

sdnf += " (";

for(int k = 0; k < N; k++)

{

if(func[k] == '1')

{

sdnf += "x" + (k + 1).ToString() + "^";

}

else if (func[k] == '0')

{

sdnf += "┐x" + (k + 1).ToString() + "^";

}

if(k == (N - 1))

{

sdnf = sdnf.Remove(sdnf.Length - 1, 1);

}

}

sdnf += ") v";

/\*if((i + 1) == Math.Pow(2, N))

{

sdnf = sdnf.Remove(sdnf.Length - 1, 1);

}\*/

}

else

{

sknf += " (";

for(int k = 0; k < N; k++)

{

if(func[k] == '0')

{

sknf += "x" + (k + 1).ToString() + "v";

}

else if(func[k] == '1')

{

sknf += "┐x" + (k + 1).ToString() + "v";

}

if(k == N - 1)

{

sknf = sknf.Remove(sknf.Length - 1, 1);

}

}

sknf += ") ^";

/\*if (i + 1 == Math.Pow(2, N))

{

sknf = sknf.Remove(sknf.Length - 1, 1);

}\*/

}

}

sdnf = sdnf.Remove(sdnf.Length - 1, 1);

sknf = sknf.Remove(sknf.Length - 1, 1);

Console.WriteLine(sdnf + "\n" + sknf);

}

public static void PrintLogTable(int n, List<int> func)

{

string result = "\nТаблица истинности для данной функции:\n";

for(int i = 1; i <= n; i++)

{

result += " x" + i.ToString() + "|";

}

result += " f|\n";

for(int i = 0; i < Math.Pow(2, n); i++)

{

string binI = ConvertToBinary(i).ToString();

while (binI.Length != n)

{

binI = binI.Insert(0, "0");

}

for (int j = 0; j < n; j++)

{

result += " " + binI[j] + " |";

}

result += " " + func[i].ToString() + "|\n";

}

Console.WriteLine(result);

}

public static int ConvertToBinary(int decNum)

{

int result = 0;

int k = 1;

while(decNum != 0)

{

result += (decNum % 2) \* k;

k \*= 10;

decNum /= 2;

}

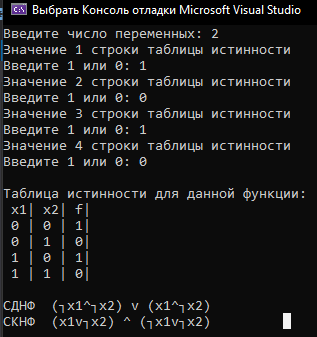
return result;

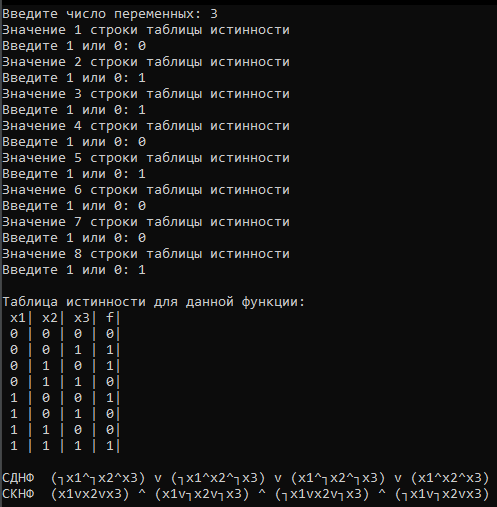
}

}

}

**Результат работы программы**

****

****