

## Оглавление

Оглавление .....	1
Задача 1 .....	2
Задача 2 .....	5
Задача 3 .....	8

## Вариант 9

### Задача 1

Разработать консольное приложение, выводящее совершенную дизъюнктивную нормальную форму и совершенную конъюнктивную нормальную форму вводимого в виде таблицы истинности выражения алгебры высказываний на переменных  $a, b, c, d, e$ . Программа должна содержать консольное меню, позволяющее пользователю изменить вводимое выражение, выполнить одно из названных действий или выйти из программы.

### Краткое описание используемых алгоритмов

При формировании массива для таблицы истинности размером  $32 \times 6$  использованы пять вложенных циклов с параметром (for), каждый из которых последовательно перебирает значения от 0 до 1. Шестой столбец отведен под выражение, которое вводит пользователь.

Используя цикл с предусловием (while), пользователю предлагается выбрать одно из четырех действий:

- 1 - ввести выражение,
- 2 - вывести совершенную дизъюнктивную нормальную форму,
- 3 - вывести совершенную конъюнктивную нормальную форму или
- 4 - выйти из программы.

Если пользователь при выборе действий вводит символ, отличный от предложенных, выводится предупреждение со звуковым сигналом.

Совершенная дизъюнктивная нормальная форма записана как дизъюнкция совершенных конъюнктивных одночленов, при которых функция равна 1, причем переменные, которые имеют значение 0, взяты с отрицанием.

Совершенная конъюнктивная нормальная форма записана как конъюнкция совершенных дизъюнктивных одночленов, при которых функция равна 0, причем переменные, которые имеют значение 1, взяты с отрицанием.

### Инструкция по работе с программой

При запуске программы выводится консольное меню, пользуясь которым необходимо выбрать одно из требуемых действий:

- 1 - ввести выражение
- 2 - вывести совершенную дизъюнктивную нормальную форму
- 3 - вывести совершенную конъюнктивную нормальную форму
- 4 - выйти из программы

Обозначения, использованные в программе:

- $\sim$  - отрицание;
- $\wedge$  - конъюнкция;
- $\vee$  — дизъюнкция.

## Текст созданных модулей

```
#include <iostream>
using namespace std;

int main()
{
    int i, j, a0, a1, a2, a3, a4, line;
    int a[32][6];
    char action = ' ';
    i = 0;
    //формирование массива для таблицы истинности
    for (a0 = 0; a0 < 2; a0++)
        for (a1 = 0; a1 < 2; a1++)
            for (a2 = 0; a2 < 2; a2++)
                for (a3 = 0; a3 < 2; a3++)
                    for (a4 = 0; a4 < 2; a4++)
                    {
                        a[i][0] = a0; a[i][1] = a1; a[i][2] = a2;
                        a[i][3] = a3; a[i][4] = a4;
                        i++;
                    }
    setlocale(LC_CTYPE, "rus");
    while (action != '4') //вывод консольного меню
    {
        cout << "Выберите одно из требуемых действий:" << endl;
        cout << "1 - ввести выражение" << endl;
        cout << "2 - вывести совершенную дизъюнктивную нормальную форму" << endl;
        cout << "3 - вывести совершенную конъюнктивную нормальную форму" << endl;
        cout << "4 - выйти из программы" << endl;
        cin >> action;
        if (action == '1') //ввод выражения
        {
            cout << "a b c d e F" << endl;
            for (i = 0; i < 32; i++)
            {
                for (j = 0; j < 5; j++)

                    cout << a[i][j] << " ";
                cin >> a[i][5];
            }
        }
        if (action == '2') //вывод совершенной дизъюнктивной нормальной формы
        {
            cout << "Совершенная дизъюнктивная нормальная форма:" << endl;
            line = 0;
            for (i = 0; i < 32; i++)
                if (a[i][5] == 1)
                {
                    if (line > 0) cout << "v";
                    if (a[i][0] == 0)
                        cout << "(~a^";
                    else cout << "(a^";
                    if (a[i][1] == 0)
                        cout << "~b^";
                    else cout << "b^";
                    if (a[i][2] == 0)
                        cout << "~c^";
                    else cout << "c^";
                    if (a[i][3] == 0)
                        cout << "~d^";
                    else cout << "d^";
                    if (a[i][4] == 0)
                        cout << "~e)";
                }
            line++;
        }
    }
}
```

```

        else cout << "e)";
        line++;
    }
    cout << endl << endl;
}
if (action == '3') //вывод совершенной конъюнктивной нормальной формы
{
    cout << "Совершенная конъюнктивная нормальная форма:" << endl;
    line = 0;
    for (i = 0; i < 32; i++)
        if (a[i][5] == 0)
        {
            if (line > 0) cout << "^";
            if (a[i][0] == 1)
                cout << "(~av";
            else cout << "(av";
            if (a[i][1] == 1)
                cout << "~bv";
            else cout << "bv";
            if (a[i][2] == 1)
                cout << "~cv";
            else cout << "cv";
            if (a[i][3] == 1)
                cout << "~dv";
            else cout << "dv";
            if (a[i][4] == 1)
                cout << "~e)";
            else cout << "e)";
            line++;
        }
    cout << endl << endl;
}
//требование корректного ввода
if ((action != '1') && (action != '2') && (action != '3') && (action != '4'))
    cout << "Введите корректное значение!\a" << endl;
}
}

```

## Выводы

На языке C++ разработано консольное приложение с консольным меню, позволяющее пользователю выводить совершенную дизъюнктивную нормальную форму и совершенную конъюнктивную нормальную форму вводимого в виде таблицы истинности выражения алгебры высказываний на пяти переменных.

## Задача 2

Разработать консольное приложение, определяющее принадлежность точки определенным образом задаваемому плоскому множеству точек.

Множество точек задано по вариантам выражением алгебры множеств из плоских фигур A, B, C, D, которые могут быть прямоугольниками или кругами. Прямоугольники задаются параметрами: координаты центра, ширина и высота; круги задаются параметрами: координаты центра, радиус.

Консольное меню должно позволять пользователю выбрать ввод фигур или выполнение проверки принадлежности вводимой точки одному из множеств, а также выход из программы.

$$((\overline{A \cap B}) \cup D) \cup C$$

### Краткое описание используемых алгоритмов

Посредством применения цикла с предусловием (while) пользователю предлагается выбрать одно из действий:

- 1 - ввести фигуры,
- 2 - выполнить проверку принадлежности или
- 3 - выйти из программы.

Если пользователь при выборе действий вводит символ, отличный от предложенных, выводится предупреждение со звуковым сигналом.

При вводе параметров фигур предлагается на выбор два варианта: прямоугольник или круг.

Проверка принадлежности точки с введенными координатами заданному плоскому множеству точек осуществляется последовательно через проверку принадлежности точки каждой из фигур с последующей обработкой результатов согласно определенным образом заданному плоскому множеству точек.

### Инструкция по работе с программой

При запуске программы выводится консольное меню, предлагающее на выбор одно из трех действий:

- 1 - ввести фигуры
- 2 - выполнить проверку принадлежности
- 3 - выйти из программы

При вводе параметров фигур предварительно требуется сделать выбор: 1 - если фигура прямоугольник, 2 - если круг.

Для проверки принадлежности точки заданному плоскому множеству точек необходимо ввести координаты точки, после чего программа выводит соответствующее сообщение.

## Текст созданных модулей

```
#include <iostream>
using namespace std;

int main()
{
    double x0[4], y0[4], w[4], h[4], r[4];
    double x, y;
    char action = ' ';
    int fig[4], i;
    bool belong[4];
    setlocale(LC_CTYPE, "rus");
    while (action != '3')
    {
        cout << "Выберите одно из требуемых действий:" << endl;
        cout << "1 - ввести фигуры" << endl;
        cout << "2 - выполнить проверку принадлежности" << endl;
        cout << "3 - выйти из программы" << endl;
        cin >> action;
        if (action == '1') //ввод параметров фигуры
        {
            for (i = 0; i < 4; i++)
            {
                cout << "Введите параметры " << i + 1 << " фигуры:" << endl;
                cout << "введите 1 - если прямоугольник, 2 - если круг: ";
                cin >> fig[i];
                if (fig[i] == 1)
                {
                    cout << "координата x0 = "; cin >> x0[i];
                    cout << "координата y0 = "; cin >> y0[i];
                    cout << "ширина w = "; cin >> w[i];
                    cout << "высота h = "; cin >> h[i];
                }
                if (fig[i] == 2) //ввод координат точки
                {
                    cout << "координата x0 = "; cin >> x0[i];
                    cout << "координата y0 = "; cin >> y0[i];
                    cout << "радиус r = "; cin >> r[i];
                }
            }
            cout << endl;
        }
        if (action == '2') //проверка принадлежности
        {
            cout << "Введите координаты точки: " << endl;
            cout << "координата x = "; cin >> x;
            cout << "координата y = "; cin >> y;
            for (i = 0; i < 4; i++)
            {
                if (fig[i] == 1)
                {
                    if ((x >= (x0[i] - w[i] / 2)) && (x <= (x0[i] + w[i] / 2))
                        && (y >= (y0[i] - h[i] / 2)) && (y <= (y0[i] + h[i] / 2)))
                        belong[i] = true;
                    else belong[i] = false;
                }
                if (fig[i] == 2)
                {
                    if (sqrt(pow((x - x0[i]), 2) + pow((y - y0[i]), 2)) <=
                        r[i])
                        belong[i] = true;
                    else belong[i] = false;
                }
            }
        }
    }
}
```

```

        }
    }
    cout << endl;
    if (!(belong[0] && belong[1]) || belong[2]) || belong[3])
        cout << "Точка принадлежит множеству";
    else cout << "Точка не принадлежит множеству";
    cout << endl << endl;
}
//требование корректного ввода
if ((action != '1')&&(action != '2')&&(action != '3'))
    cout << "Введите корректное значение!\a" << endl;
}
}

```

## Выводы

На языке C++ разработано консольное приложение, определяющее принадлежность точки плоскому множеству точек, заданному выражением алгебры множеств и состоящему из плоских фигур, которые могут быть прямоугольниками или кругами. Прямоугольники при этом задаются через консольное меню координатами центра, шириной и высотой, а круги – координатами центра и радиусом.

### **Задача 3**

Разработать консольное приложение, позволяющее вводить граф в виде матрицы смежности с клавиатуры и из файла, осуществлять представление графа в других формах: матрица инцидентий, списки смежности, списки инцидентий.

Консольное меню должно позволять пользователю выбрать ввод графа с клавиатуры или из файла или выполнить требуемые способы представления.

#### **Краткое описание используемых алгоритмов**

При написании программы использован цикл с предусловием (while). Пользователю предлагается выбор в основном меню:

- 1 - ввести граф с клавиатуры,
- 2 - ввести граф из файла,
- 3 - вывести граф в другой форме или
- 4 - выйти из программы

и во вспомогательном для выбора вывода графа в виде:

- 1 - матрицы инцидентий,
- 2 - списков смежности,
- 3 - списков инцидентий или
- 0 - выход в основное меню.

Для вводимой с клавиатуры или из файла матрицы смежности и выводимой матрицы инцидентий созданы два двумерных массива. В случае возникновения ошибок при открытии файла предусмотрен вывод сообщения об ошибке.

#### **Инструкция по работе с программой**

При запуске программы выводится консольное меню:

Выберите требуемое действие:

- 1 - ввести граф с клавиатуры
- 2 - ввести граф из файла
- 3 - вывести граф в другой форме
- 4 - выйти из программы

В случае вывода сообщения об ошибке необходимо проверить нахождение файла по указанному пути.

Выбор пункта «3 - вывести граф в другой форме» открывает вспомогательное меню:

Вывод графа в виде:

- 1 - матрицы инцидентий
- 2 - списков смежности
- 3 - списков инцидентий
- 0 - выход в основное меню.



## Текст созданных модулей

```
#include <string>
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    char action = ' ';
    int i, j, n, k, l, m;
    string line;
    int a[20][20], b[20][20];
    setlocale(LC_CTYPE, "rus");
    while (action != '4')
    {
        cout << "Выберите требуемое действие:\n";
        cout << "1 - ввести граф с клавиатуры\n";
        cout << "2 - ввести граф из файла\n";
        cout << "3 - вывести граф в другой форме\n";
        cout << "4 - выйти из программы\n";
        cin >> action;
        if (action == '1') //ввод графа с клавиатуры
        {
            cout << "Введите количество вершин графа: "; cin >> n;
            cout << "Введите граф в виде матрицы смежности:\n";
            for (i = 0; i < n; i++)
                for (j = 0; j < n; j++)
                    cin >> a[i][j];

            cout << endl;
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                    cout << a[i][j] << " ";
                cout << endl;
            }
            cout << endl;
        }
        if (action == '2') //ввод графа из файла
        {
            ifstream file;
            file.open ("D:\\03.txt");
            if (!file.is_open())
            {
                cout << "Ошибка открытия файла!\n";
            }
            else
            {
                getline(file, line);
                n = (line.length()+1)/2;
                file.clear();
                file.seekg(0, ios_base::beg);
                for (i = 0; i < n; i++)
                    for (j = 0; j < n; j++)
                        file >> a[i][j];

                file.close();
                cout << "Граф из файла:\n";
                for (i = 0; i < n; i++)
                {
                    for (j = 0; j < n; j++)
                        cout << a[i][j] << " ";
                    cout << endl;
                }
            }
        }
    }
}
```

```

        cout << endl;
    }
}
if (action == '3')
{
    cout << "Вывод графа в виде:\n";
    cout << "1 - матрицы инцидентий\n";
    cout << "2 - списков смежности\n";
    cout << "3 - списков инцидентий\n";
    cout << "0 - выход в основное меню\n";
    cin >> action;
    if (action == '1')
    {
        cout << "Граф в виде матрицы инцидентий:\n";
        k = 0;
        for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)
                if (a[i][j] == 1) k++;

        l = 0;
        for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)
                if ((i < j) & (a[i][j] == 1))
                {
                    for (m = 0; m < n; m++)
                    {
                        if ((m == i) | (m == j))
                            b[l][m] = 1;
                        else b[l][m] = 0;
                    }
                    l++;
                }
        for (l = 0; l < k / 2; l++)
        {
            for (m = 0; m < n; m++)
                cout << b[l][m] << " ";
            cout << endl;
        }
        cout << endl;
    }
    if (action == '2')
    {
        cout << "Граф в виде списков смежности:\n";
        l = 0;
        for (i = 0; i < n; i++)
        {
            cout << i + 1 << ": ";
            for (j = 0; j < n; j++)
                if (a[i][j] == 1) l++;
            for (j = 0; j < n; j++)
                if (a[i][j] == 1)
                {
                    cout << j + 1;
                    if (l - 1 > 0) cout << ", ";
                    l--;
                }
            cout << endl;
        }
        cout << endl;
    }
    if (action == '3')
    {
        cout << "Граф в виде списков инцидентий:\n";
        for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)
                if ((i < j) & (a[i][j] == 1))

```

```

        cout << "[" << i + 1 << ", " << j + 1 <<
        "]" \n";
        cout << endl;
    }
    if (action == '0')
    {
        cout << endl;
    }
}
//требование корректного ввода
if ((action != '0') & (action != '1') & (action != '2') & (action != '3') & (action !=
'4'))
    cout << "Введите корректное значение!\n\n";
}
}

```

## Выводы

На языке C++ разработано консольное приложение, позволяющее вводить с клавиатуры или из файла граф в виде матрицы смежности и осуществлять представление графа в формах матрицы инцидентий, списков смежности и списков инцидентий.