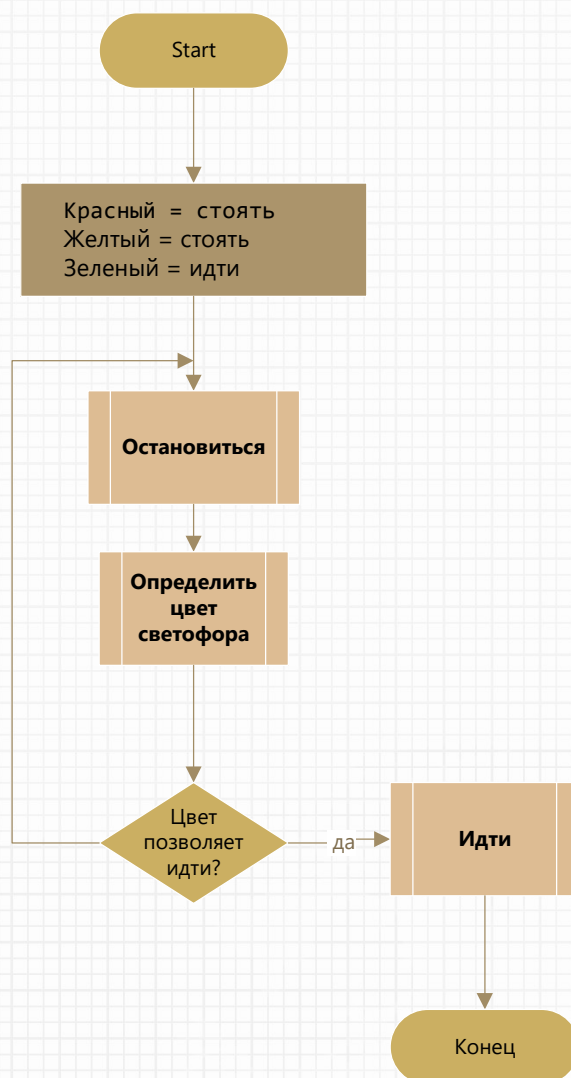


Traffic Light



Code:

```
using System;

namespace TrafficLight
{
    internal class Program
    {
        /// <summary>
        /// Текущий цвет светофора
        /// </summary>
        private static ControlColor CurrentColor = ControlColor.Red;
        /// <summary>
        /// Цвета светофора
        /// </summary>
        private enum ControlColor
        {
            Red = 114,
            Yellow = 121,
            Green = 103
        }
        /// <summary>
        /// Метод обработки значения Стоп.
        /// </summary>
        private static void StopWalk()
        {
            Console.WriteLine("Stop Walk!");
        }
        /// <summary>
        /// Метод обработки значения Иди.
        /// </summary>
        private static void GoWalk()
        {
            Console.WriteLine("Please Walk!\nThank you, Bon Voyage!");
        }
        /// <summary>
        /// Считываем цвет светофора в CurrentColor
        /// </summary>
        private static void ReadColor()
        {
            Console.WriteLine("Please, tell me what color is the traffic light?\n(r,y,g)");
            var ReadKey = Console.ReadKey();
            CurrentColor = (ControlColor) Convert.ToInt32(ReadKey.KeyChar);
            if (CurrentColor == ControlColor.Red)
                Console.WriteLine(" You said red.");
            else if (CurrentColor == ControlColor.Yellow)
                Console.WriteLine(" You said yellow.");
            else if (CurrentColor == ControlColor.Green)
                Console.WriteLine(" You said green.");
            else
            {
                Console.WriteLine(" You said don't understand what: " +
                    ReadKey.Key + "(");
                CurrentColor = ControlColor.Red;
            }
        }
        /// <summary>
        /// По полученному цвету светофора определяем действие
        /// </summary>
        /// <param name="controlColor"> полученный цвет светофора</param>
        /// <returns></returns>
        private static bool ColorEvaluation(ControlColor controlColor)
        {
            switch (controlColor)
            {
                case ControlColor.Red:
                    ChMove = false;
                    break;
                case ControlColor.Yellow:
                    ChMove = false;
                    break;
                case ControlColor.Green:
                    ChMove = true;
                    break;
                default:
                    throw new ArgumentOutOfRangeException(nameof(controlColor),
                        controlColor, null);
            }
            return ChMove;
        }
        private static bool chMove;
        /// <summary>
        /// Свойство выбора движения с управляемыми цветом светофора значениями
        /// </summary>
        private static bool ChMove
        {
            get => chMove;
            set
            {
                // ReSharper disable once ComplexConditionExpression
                if (CurrentColor == ControlColor.Red ||
                    CurrentColor == ControlColor.Yellow)
                {
                    StopWalk();
                    chMove = value;
                }
                else
                {
                    GoWalk();
                    chMove = value;
                }
            }
        }
    }
    static void Main()
    {
        StopWalk();
        do ReadColor();
        while (!ColorEvaluation(CurrentColor));
    }
}
```