Вариант 1

Исходный

class Program

{

static void Main(string[] args)

{

int intNum();

Console.WriteLine(Введите число:);

Convert.ToInt32(Console.ReadLine()) = intNum;

Console.WriteLine("Количество делителей введенного числа: ", aaa(intNum));

}

public static int aaa(bool intIndex)

{

int intC, intD;

int intTotal;

for (int intC =0; intC <= intIndex; intC++)

{

if (intIndex / intC == intIndex / intC)

{

intTotal += 1;

intD += 1;

}

}

}

}

Исправленный

{

class Program

{

static void Main(string[] args)

{

// Лишние скобки ()

int intNum;

// Добавим ""

// Поясним, что число должно быть натуральным

Console.WriteLine("Введите натуральное число:");

// Переменная должна стоять в начале intNum = ...

intNum = Convert.ToInt32(Console.ReadLine());

// Поменяем , на +

Console.WriteLine("Количество делителей введенного числа: " + Divisors(intNum));

}

// По правилам - именуем функции с заглавной буквы, придумаем название получше

// bool -> int

public static int Divisors(int intIndex)

{

// Присвоим intC = 2, т.к. 0 не может быть нашим делителем, а 1 это делитель любого натурального числа

// Присвоим intTotal = 1, т.к. у нас будет как минимум один делитель

// Лишняя переменная intD

int intC = 2, intTotal = 1;

// Тут нам не нужна локальная переменная (убираем int intC = 0)

for (; intC <= intIndex; intC++)

{

if (intIndex % intC == 0)

{

intTotal += 1;

}

}

// Добавим return, должна же наша функция что-то вернуть

return intTotal;

}

}

}

Исходный

class Program

{

static int ChangeValue(int ChC)

{

switch (ChC)

{

case 'M':

case 'm':

return 1000;

case 'D':

case 'd':

return 500;

case 'C':

case 'c':

return 100;

case 'L':

case 'l':

return 50;

case 'X':

case 'x':

return 10;

case 'V':

case 'v':

return 5;

case 'I':

case 'i':

return 1;

default:

return 0;

}

}

static int GetValue(int strInput, string intI)

{

int intX;

intX = ChangeValue(strInput[intI]) + 1;

if (intI == strInput.length -1)

{

GetValue = intX;

}

else

{

if (intX < ChangeValue(strInput[intI +1]))

{

GetValue = -intX - GetValue(strInput, intI + 1);

}

else

{

GetValue = intX - GetValue(strInput, intI + 1);

}

}

}

static void Main(string[] args)

{

Console.WriteLine(Введите ваше римское число:);

string strInput;

strInput = Console.ReadLine();

Console.WriteLine(Ваше число = + GetValue(StrInput, 0));

}

}

Исправленный

{

class Program

{

static int ChangeValue(int ChC)

{

switch (ChC)

{

case 'M':

case 'm':

return 1000;

case 'D':

case 'd':

return 500;

case 'C':

case 'c':

return 100;

case 'L':

case 'l':

return 50;

case 'X':

case 'x':

return 10;

case 'V':

case 'v':

return 5;

case 'I':

case 'i':

return 1;

default:

return 0;

}

}

// Перепутаны int и string для переменных

static int GetValue(string strInput, int intI)

{

int intX;

// + 1 лишнее

intX = ChangeValue(strInput[intI]);

// Length с заглавной

if (intI == strInput.Length - 1)

{

// Если мы имеем дело с последним символом, просто вернем его значение

return intX;

}

else

{

if (intX < ChangeValue(strInput[intI + 1]))

{

// Кажется это рекурсия, поставим правильные знаки

intX = -intX + GetValue(strInput, intI + 1);

}

else

{

// И тут

intX = intX + GetValue(strInput, intI + 1);

}

}

// Вернем значение

return intX;

}

static void Main(string[] args)

{

// Добавим ""

Console.WriteLine("Введите ваше римское число: ");

string strInput;

strInput = Console.ReadLine();

// Добавим "", ошибка в названии переменной StrInput -> strInput

Console.WriteLine("Ваше число = " + GetValue(strInput, 0));

}

}

}