**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа энергетики

Отделение электроэнергетики и электротехники

Направление: 09.04.03 Прикладная информатика

Дисциплина: Программирование и основы алгоритмизации

Отчет

по лабораторной работе №2

**Линейная программа и программа с ветвлениями**

Вариант 12

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. О-5КМ41 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Рустамов Р.А. |
| Проверил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Гончаров А.С. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск-2025

# ХОД РАБОТЫ

При запуске программа показывает меню и ждёт выбор пользователя. Любой ввод проверяется через TryParse(): если введено не число или число вне допустимого диапазона, выводится понятная подсказка красным цветом и предлагается ввести заново. После выполнения выбранного пункта меню программа возвращается к списку действий.

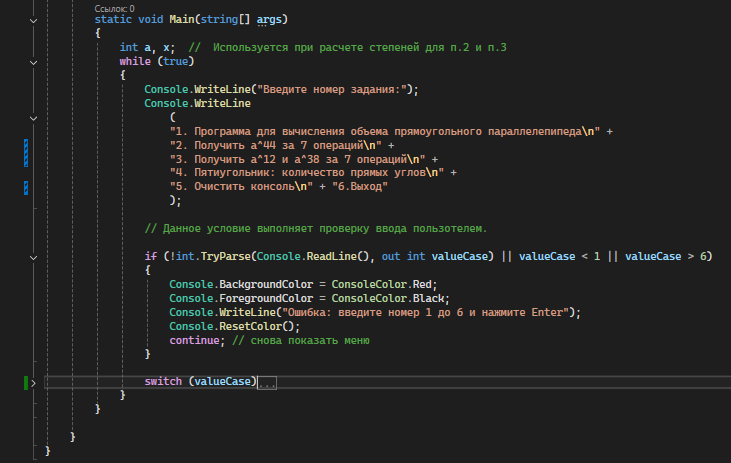


Рисунок – Код основного меню

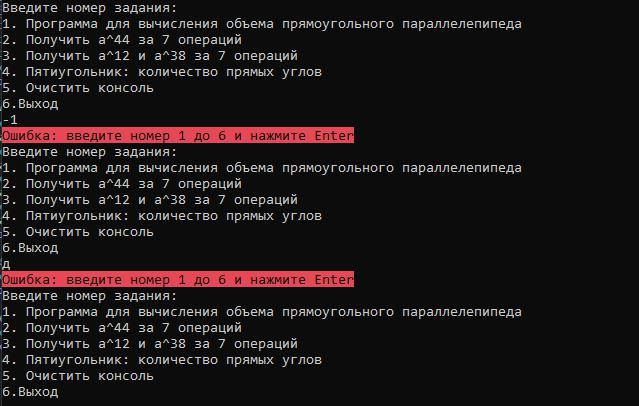


Рисунок – Пример работы кода

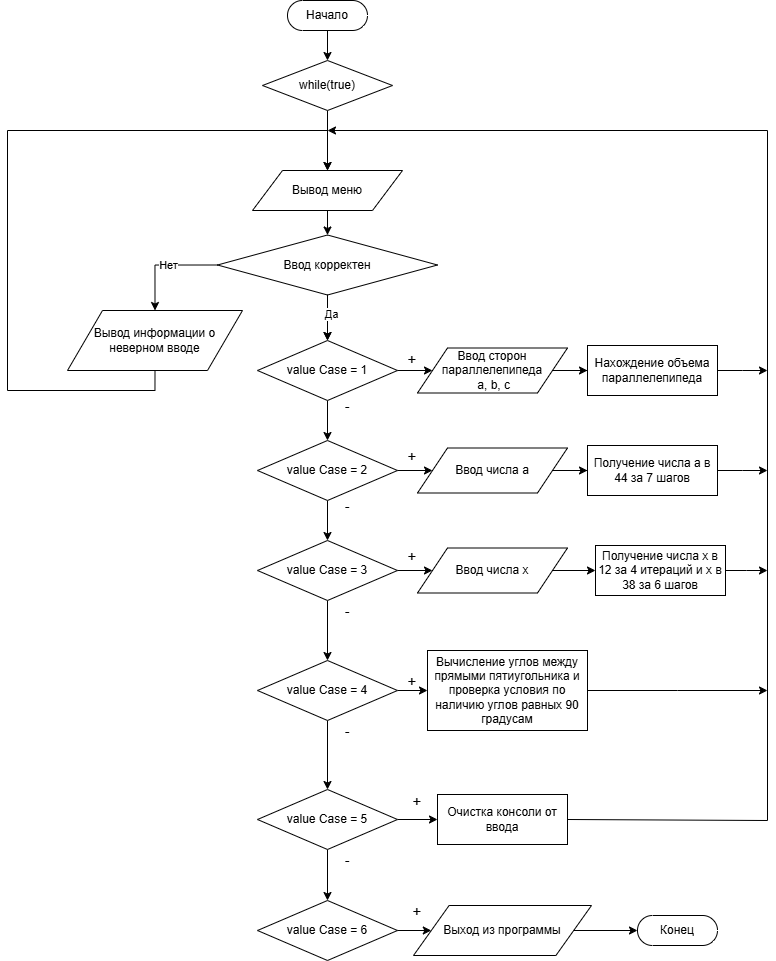


Рисунок – Блок схема работы внутри Main

1. Написать программу вычисления объема прямоугольного параллелепипеда. Извне вводятся длина a, ширина b и высота c. пример: a=10, b=15, с=20 Ответ: 3000

Пользователь вводит три положительных целых a, b, c. Объём считается как произведение сторон. Результат хранится в типе long, чтобы не было переполнения при больших значениях. На экран выводится «Объём равен: …».



Рисунок – Вычисление объема параллелепипеда

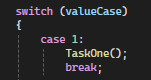


Рисунок – Вызов метода TaskOne()

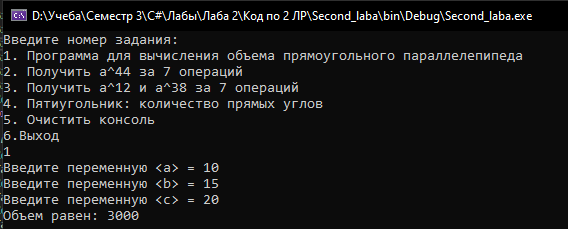


Рисунок – Пример расчета объема

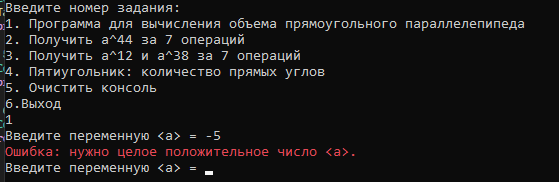


Рисунок – Контроль исправности ввода

1. Не используя никакие арифметические действия, кроме умножения, записать операторы присваивания, которые позволят получить для заданного а^44 за 6 операций;

Вводится целое a. Степень вычисляется «умной» цепочкой промежуточных степеней (так называемая addition chain), что позволяет уложиться в 7 умножений — это оптимально для 44. Для надёжности промежуточные значения считаются в BigInteger. В коде показаны сразу два эквивалентных способа; оба дают одинаковый результат, который выводится для проверки.

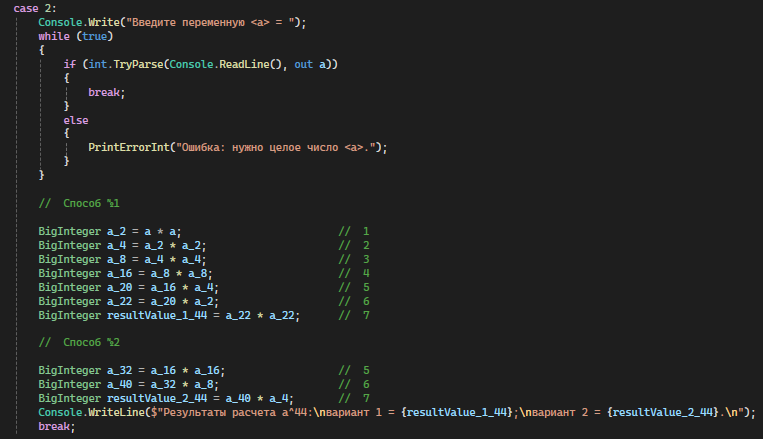


Рисунок – Пример получения числа "а" в 44 степени

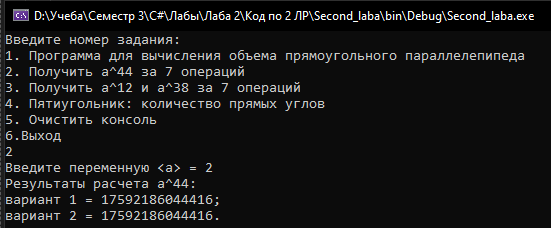


Рисунок – Пример работы алгоритма по поиску числа а в степени 44



Рисунок – Пример расчета "а" в Mathcad 15

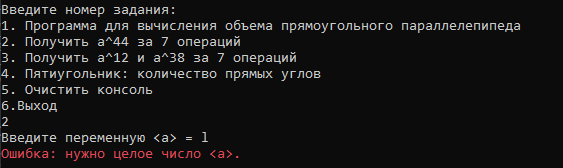


Рисунок – Контроль ввода

1. Не используя никакие арифметические действия, кроме умножения, записать операторы присваивания, которые позволят получить для заданного а^12 и а^38 за 6 операций.

* строится за **4 умножения**: .
* строится за **6 умножений** (если уже посчитано):. Результаты печатаются на экран.

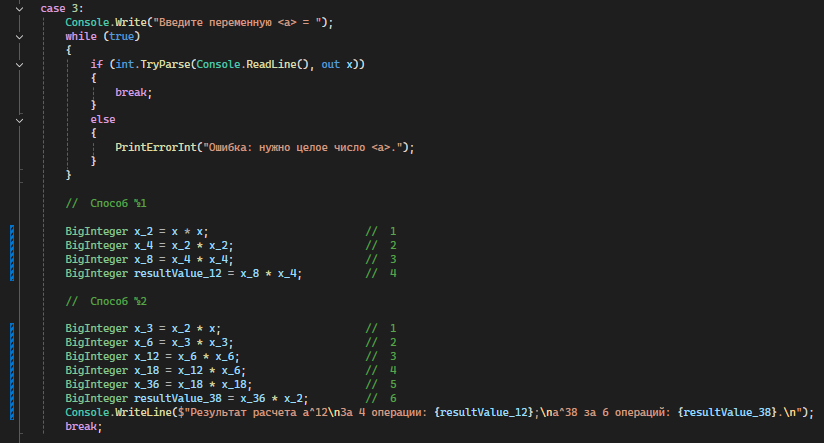


Рисунок –Пример получения числа "а" в 12 и 38 степени

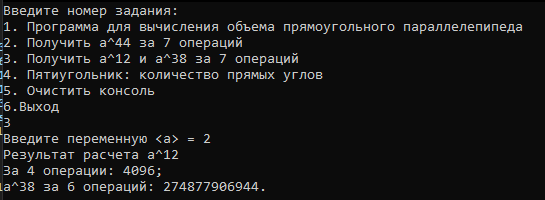


Рисунок – Пример поиска числа а в степени 12 и 38

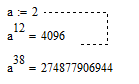


Рисунок – Пример расчета "а" в Mathcad 15

1. В пятиугольнике, заданном координатами своих вершин, подсчитать количество прямых углов (если они есть).

На вход подаются координаты пяти точек. Для каждой вершины проверяется, прямой ли угол, через скалярное произведение соседних рёбер: если оно равно нулю — угол 90°. Программа сообщает по каждой вершине «прямой/не прямой» и выводит общее количество прямых углов. В Main добавлены четыре готовых примера пятиугольников для наглядной проверки.

Так как при косинус равен нулю, то знаменатель можно не учитывать в расчете.

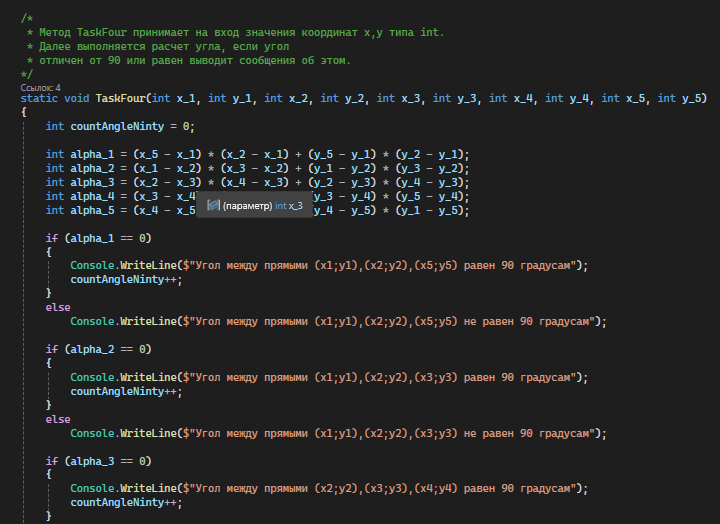


Рисунок – Поиск прямых углов между сторонами пятиугольника

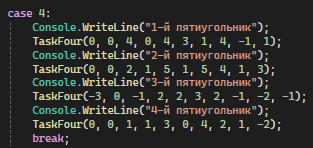


Рисунок – Вызов метода TaskFourt()

Ниже приводятся примеры для 1 и 3 пятиугольника, которые подтверждают корректность работы алгоритма.

|  |  |
| --- | --- |
|  | |
|  |
|  |
|  |

Рисунок – Пример работы метода поиска прямых углов

# Вывод

Программа корректно обрабатывает пользовательский ввод, демонстрирует приёмы оптимизации вычисления степеней (через цепочки умножений), безопасно работает с большими числами (BigInteger) и решает простую геометрическую задачу определения прямых углов. Всё необходимое для проверки выводится на экран в понятном виде.

ПРИЛОЖЕНИЕ А

using System;

using System.Numerics; // Для использования BigInteger

namespace SecondLaba

{

class Program

{

static void PrintErrorInt(string error)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(error);

Console.ResetColor();

}

// Данная функция вычисляет объём параллелепипеда по трем введенным переменным а, b, c

static void TaskOne()

{

int a, b, c;

while (true)

{

Console.Write("Введите переменную <a> = ");

if (int.TryParse(Console.ReadLine(), out a) && a > 0)

{

break;

}

else

{

PrintErrorInt("Ошибка: нужно целое положительное число <a>.");

}

}

while (true)

{

Console.Write("Введите переменную <b> = ");

if (int.TryParse(Console.ReadLine(), out b) && b > 0)

{

break;

}

else

{

PrintErrorInt("Ошибка: нужно целое положительное число <b>.");

}

}

while (true)

{

Console.Write("Введите переменную <c> = ");

if (int.TryParse(Console.ReadLine(), out c) && c > 0)

{

break;

}

else

{

PrintErrorInt("Ошибка: нужно целое положительное число <с>.");

}

}

long resultVolume = (long)a \* b \* c; // Объем параллелепипеда

Console.WriteLine($"Объем равен: {resultVolume}\n");

}

/\*

\* Метод TaskFour принимает на вход значения координат х,у типа int.

\* Далее выполняется расчет угла, если угол

\* отличен от 90 или равен выводит сообщения об этом.

\*/

static void TaskFour(int x\_1, int y\_1, int x\_2, int y\_2, int x\_3, int y\_3, int x\_4, int y\_4, int x\_5, int y\_5)

{

int countAngleNinty = 0;

int alpha\_1 = (x\_5 - x\_1) \* (x\_2 - x\_1) + (y\_5 - y\_1) \* (y\_2 - y\_1);

int alpha\_2 = (x\_1 - x\_2) \* (x\_3 - x\_2) + (y\_1 - y\_2) \* (y\_3 - y\_2);

int alpha\_3 = (x\_2 - x\_3) \* (x\_4 - x\_3) + (y\_2 - y\_3) \* (y\_4 - y\_3);

int alpha\_4 = (x\_3 - x\_4) \* (x\_5 - x\_4) + (y\_3 - y\_4) \* (y\_5 - y\_4);

int alpha\_5 = (x\_4 - x\_5) \* (x\_1 - x\_5) + (y\_4 - y\_5) \* (y\_1 - y\_5);

if (alpha\_1 == 0)

{

Console.WriteLine($"Угол между прямыми (x1;y1),(x2;y2),(x5;y5) равен 90 градусам");

countAngleNinty++;

}

else

Console.WriteLine($"Угол между прямыми (x1;y1),(x2;y2),(x5;y5) не равен 90 градусам");

if (alpha\_2 == 0)

{

Console.WriteLine($"Угол между прямыми (x1;y1),(x2;y2),(x3;y3) равен 90 градусам");

countAngleNinty++;

}

else

Console.WriteLine($"Угол между прямыми (x1;y1),(x2;y2),(x3;y3) не равен 90 градусам");

if (alpha\_3 == 0)

{

Console.WriteLine($"Угол между прямыми (x2;y2),(x3;y3),(x4;y4) равен 90 градусам");

countAngleNinty++;

}

else

Console.WriteLine($"Угол между прямыми (x2;y2),(x3;y3),(x4;y4) не равен 90 градусам");

if (alpha\_4 == 0)

{

Console.WriteLine($"Угол между прямыми (x3;y3),(x4;y4),(x5;y5) равен 90 градусам");

countAngleNinty++;

}

else

Console.WriteLine($"Угол между прямыми (x3;y3),(x4;y4),(x5;y5) не равен 90 градусам");

if (alpha\_5 == 0)

{

Console.WriteLine($"Угол между прямыми (x1;y1),(x4;y4),(x5;y5) равен 90 градусам");

countAngleNinty++;

}

else

Console.WriteLine($"Угол между прямыми (x1;y1),(x4;y4),(x5;y5) не равен 90 градусам");

if (countAngleNinty != 0)

Console.WriteLine($"Всего углов равных 90 градусам: {countAngleNinty}\n");

else

Console.WriteLine("В данном пятиугольнике нет прямых углов\n");

}

static void Main(string[] args)

{

int a, x; // Используется при расчете степеней для п.2 и п.3

while (true)

{

Console.WriteLine("Введите номер задания:");

Console.WriteLine

(

"1. Программа для вычисления объема прямоугольного параллелепипеда\n" +

"2. Получить a^44 за 7 операций\n" +

"3. Получить a^12 и a^38 за 7 операций\n" +

"4. Пятиугольник: количество прямых углов\n" +

"5. Очистить консоль\n" + "6.Выход"

);

// Данное условие выполняет проверку ввода пользотелем.

if (!int.TryParse(Console.ReadLine(), out int valueCase) || valueCase < 1 || valueCase > 6)

{

Console.BackgroundColor = ConsoleColor.Red;

Console.ForegroundColor = ConsoleColor.Black;

Console.WriteLine("Ошибка: введите номер 1 до 6 и нажмите Enter");

Console.ResetColor();

continue; // снова показать меню

}

switch (valueCase)

{

case 1:

TaskOne();

break;

case 2:

Console.Write("Введите переменную <a> = ");

while (true)

{

if (int.TryParse(Console.ReadLine(), out a))

{

break;

}

else

{

PrintErrorInt("Ошибка: нужно целое число <a>.");

}

}

// Способ №1

BigInteger a\_2 = a \* a; // 1

BigInteger a\_4 = a\_2 \* a\_2; // 2

BigInteger a\_8 = a\_4 \* a\_4; // 3

BigInteger a\_16 = a\_8 \* a\_8; // 4

BigInteger a\_20 = a\_16 \* a\_4; // 5

BigInteger a\_22 = a\_20 \* a\_2; // 6

BigInteger resultValue\_1\_44 = a\_22 \* a\_22; // 7

// Способ №2

BigInteger a\_32 = a\_16 \* a\_16; // 5

BigInteger a\_40 = a\_32 \* a\_8; // 6

BigInteger resultValue\_2\_44 = a\_40 \* a\_4; // 7

Console.WriteLine($"Результаты расчета a^44:\nвариант 1 = {resultValue\_1\_44};\nвариант 2 = {resultValue\_2\_44}.\n");

break;

case 3:

Console.Write("Введите переменную <a> = ");

while (true)

{

if (int.TryParse(Console.ReadLine(), out x))

{

break;

}

else

{

PrintErrorInt("Ошибка: нужно целое число <a>.");

}

}

// Способ №1

BigInteger x\_2 = x \* x; // 1

BigInteger x\_4 = x\_2 \* x\_2; // 2

BigInteger x\_8 = x\_4 \* x\_4; // 3

BigInteger resultValue\_12 = x\_8 \* x\_4; // 4

// Способ №2

BigInteger x\_3 = x\_2 \* x; // 1

BigInteger x\_6 = x\_3 \* x\_3; // 2

BigInteger x\_12 = x\_6 \* x\_6; // 3

BigInteger x\_18 = x\_12 \* x\_6; // 4

BigInteger x\_36 = x\_18 \* x\_18; // 5

BigInteger resultValue\_38 = x\_36 \* x\_2; // 6

Console.WriteLine($"Результат расчета a^12\nЗа 4 операции: {resultValue\_12};\na^38 за 6 операций: {resultValue\_38}.\n");

break;

case 4:

Console.WriteLine("1-й пятиугольник");

TaskFour(0, 0, 4, 0, 4, 3, 1, 4, -1, 1);

Console.WriteLine("2-й пятиугольник");

TaskFour(0, 0, 2, 1, 5, 1, 5, 4, 1, 3);

Console.WriteLine("3-й пятиугольник");

TaskFour(-3, 0, -1, 2, 2, 3, 2, -1, -2, -1);

Console.WriteLine("4-й пятиугольник");

TaskFour(0, 0, 1, 1, 3, 0, 4, 2, 1, -2);

break;

case 5:

Console.Clear();

break;

case 6:

Console.WriteLine("Выход из программы.");

return; // завершаем Main

}

}

}

}

}