using System;

using System.Collections.Generic;

using System.Linq;

namespace UKPO\_4

{

public class Program

{

static void Main(string[] args)

{

1.Console.Write("Введите нижнюю границу диапазона : ");

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

2.if (min < 100\_000)

3. min = 100\_000;

4.Console.Write("Введите нижнюю границу диапазона : ");

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

5.if (max > 999\_999)

6. max = 999\_999;

7.List<int> result = new();

for (

8. int x = min;

9. x <= max;

10. x++

)

{

11. int first\_three = x / 1000;

int second\_three = x % 1000;

12. if (TreeDigitNumberSum(first\_three)==TreeDigitNumberSum(second\_three))

13. result.Add(x);

}

14.if (result.Any()) Console.WriteLine(

15. string.Join(Environment.NewLine, result));

16.else Console.WriteLine("Требуемых чисел нет");

int TreeDigitNumberSum(int number)

{

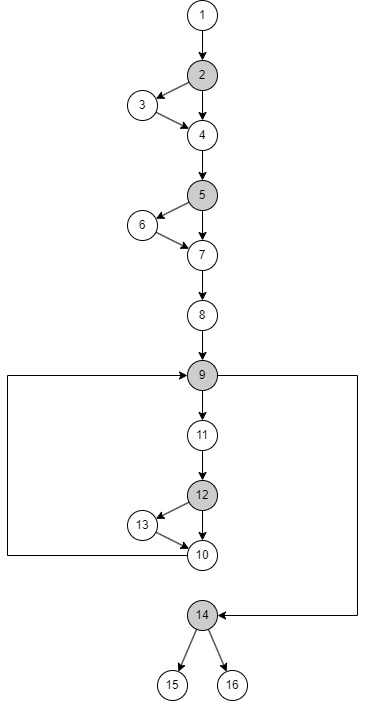
return number / 100 + number % 100 / 10 + number % 10;

}

}

}

}



## Оценка алгоритмической сложности

### Первый критерий

### Второй критерий

Цикломатическое число:

Ациклические маршруты:

Циклические маршруты:

### Метрика Маккейба

Матрица смежности:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 |  |  |  |
| 11 |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |

Матрица достижимости:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |