Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

Кафедра «Информатика и вычислительная техника»

Отчёт по лабораторной работе № 3

по дисциплине

«Проектирование и тестирование программного обеспечения»

|  |  |
| --- | --- |
|  | Выполнил  Студент гр. МО-181  Грачев Д. В.    (подп., дата)  Проверил  Старший преподаватель каф. ИВТ  Звонов А.О.    (подп., дата) |

Омск 2020

**Задача 5.3 (Единицы)**

PC/UVa IDs: 110504/10127

Для любого заданного целого числа 0 < n < 10 000, не кратного 2 и 5, существует число, кратное n, такое, что в десятичной записи оно является последовательностью единиц. Сколько цифр в наименьшем таком числе? Обратите внимание, что само число определять не обязательно.

Ниже представлена программа:

namespace L31

{

class Program

{

static void Main(string[] args)

{

using (StreamReader str = new StreamReader("file.txt"))

{

string input = null;

while ((input = str.ReadLine()) != null)

{

UInt64 n = Convert.ToUInt64(input);

if (n >= 10000 || n == 0 || n % 2 == 0 || n % 5 == 0)

Console.WriteLine("No solution");

else

Console.WriteLine(GetUnits(n));

}

}

}

static int GetUnits(UInt64 n)

{

int[] notUnits = new int[] { 0, 2, 3, 4, 5, 6, 7, 8, 9 };

for (int i = 1; (UInt64)i \* n <= UInt64.MaxValue; i += 2)

{

if (i % 2 == 0 || i % 5 == 0)

continue;

int[] digs = GetDigitsArray((UInt64)i \* n);

if (digs.Intersect(notUnits).ToArray().Length == 0)

return digs.Length;

}

return 0;

}

static int[] GetDigitsArray(UInt64 n)

{

List<int> digs = new List<int>();

UInt64 d = 10;

while (n > 0)

{

digs.Add((int)(n % d));

n /= 10;

}

digs.Reverse();

return digs.ToArray();

}

}

}

Выбранный путь решения задачи:

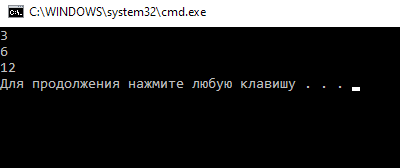
1. Считывание данных

2. Если число подходит под условия (0 < n < 10 000, не кратно 2 и 5), вызвать функцию, умножать это число в цикле for.

3. Если цифры произведения не содержат ничего кроме единиц, вернуть кол-во единиц.  
4. Вывод результата

Используемые структуры данных для решения задачи: массивы, строки

Результат выполнения программы для входных данных 3, 7, 9901:



**Задача 6.3 (Счет)**

PC/UVaIDs: 110603/10198

Маленький Мумба знает, как считать до миллиона африканских долларов, но писать числа он еще только учится. Он уже научился писать цифры 1, 2, 3 и 4. Правда, он ещё не знает, что 4 отличается от 1, так что он думает, что 4 - это просто другой способ написания 1.

Ему нравится простая игра, которую он придумал: он пишет числа из четырех известных ему цифр и складывает их значения. Например:

132 = 1 + 3 + 2 = 6

112314 = 1 + 1 + 2 + 3 + 1 + 1 = 9 (не забывайте, что Мумба считает, что 4 = 1).

Мумбе интересно, сколько таких чисел с суммой равной числу n он может составить. Для n = 2 он может составить 5 чисел: 11 , 14 , 41 , 44 и 2. Тем не менее, он не может найти это количество для n больших 2 и поэтому просит вас помочь ему.

Ниже представлена программа:

namespace L32

{

class Program

{

static void Main(string[] args)

{

using (StreamReader str = new StreamReader("file.txt"))

{

string input = null;

while ((input = str.ReadLine()) != null)

{

int n = Convert.ToInt32(input);

if (n >= 1000 || n < 1)

Console.WriteLine("No solution");

else

Console.WriteLine(GetNumbsQuant(n));

}

}

}

static int GetNumbsQuant(int n)

{

List<int[]> numbs = new List<int[]>();

int[] numb = new int[] { 1 };

numbs.Add(numb);

int bits = 1, quant = 0;

while (bits <= n)

{

numb = IncrNumb(numb);

numbs.Add(numb);

bits = numb.Length;

}

foreach (var number in numbs)

if (CalcSum(number) == n) quant++;

return quant;

}

static int CalcSum(int[] numb)

{

int sum = 0;

foreach (var digit in numb)

sum += digit != 4 ? digit : 1;

return sum;

}

static int[] IncrNumb(int[] numb)

{

List<int> newNumb = new List<int>();

int shift = 1;

for (int i = 0; i < numb.Length; i++)

{

int digit = (numb[i] + shift) % 5;

newNumb.Add(digit == 0 ? 1 : digit);

shift = (numb[i] + shift) / 5;

}

if (shift != 0) newNumb.Add(shift);

return newNumb.ToArray();

}

}

}

Выбранный путь решения задачи:

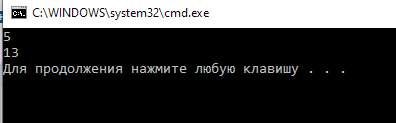
1. Считываем данные

2. Если n >= 1 и n <= 1000, вызвать функци  
3. Для каждого замеченного трюка, вызываем функцию Trick, которая меняет порядок карт в соответствии с трюком

4. Выводим результат

Используемые структуры данных для решения задачи: списки, массивы

Результат работы программы при входных данных 2, 3:



Выводы:

В процессе выполнения лабораторной работы пришло понимание того, что предварительное продумывание проектирования программы сильно помогает в процессе ее написания, также как и прописывание всех переменным и функциям их значение в программе.