***Вариант 1***

Найдите сумму цифр всех чисел от 1 до включительно в -ичной системе (сумма считается в десятичной системе счисления).

Для этого реализуйте следующие методы:

* **int CountDigitSum(int n, int mod)** - находит сумму цифр числа в -ичной системе
* **int CountDigitSumOfAllAndCountAllEndsWithZero(int n, int mod, out int count)** - находит сумму цифр всех чисел от до включительно в -ичной системе, а также количество чисел в диапазоне от до , запись которых в системе счисления оканчивается на 0.
* **int ReadInt(string message, string errorMessage, int maxValue)** - считывает число с клавиатуры, которое должно быть меньшее , выводя при этом на экран вспомогательное сообщение для ввода . В случае ошибки, выводит и запрашивает ввод заново.

Пользователь вводит два числа:

* – максимальное число в диапазоне от до
* - основание системы счисления (в диапазоне от 2 до 16)

Вывести сумму цифр всех чисел от 1 до включительно. Также выведите на экран количество чисел, оканчивающихся на 0 в этой системе счисления.

Снабжайте все выводы на экран вспомогательными текстовыми сообщениями.

Нужно реализовать повтор решения. Также можно реализовывать свои дополнительные методы для решения данной задачи.

**Проверьте себя:**

Тест 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| n = 5  mod = 2 | Сумма = 7  Количество числе, оканчивающихся на 0 = 2 |

Тест 2

|  |  |
| --- | --- |
| Ввод | Вывод |
| n = 1000  mod = 5 | Сумма = 8004  Количество числе, оканчивающихся на 0 = 200 |

***Вариант 2***

Найдите *(сумму) / (побитовую сумму по модулю 2) / (сумму побитовых отрицаний)* всех перевернутых чисел от 1 до включительно.

Для этого реализуйте следующие методы:

* **int ReverseNumber(int n)** - переворачивает число. Например, число 1021 станет 1201, число 1030 превратится в 3.
* **int CalculateAllNumbersAndCountAllPalindromes(int n, int type, out int count)** - *(сумму) / (побитовую сумму по модулю 2) / (сумму побитовых отрицаний)* всех перевернутых чисел от 1 до включительно.
  + Если type = 0, то метод находит сумму всех перевернутых чисел
  + Если type = 1, то метод находит побитовую сумму по модулю 2
  + Если type = 2, то метод находит сумму побитовых отрицаний

Также метод считает количество палиндромов, то есть чисел, которые не меняются от переворота.

* **int ReadInt(string message, string errorMessage, int maxValue)** - считывает число с клавиатуры, которое должно быть меньшее , выводя при этом на экран вспомогательное сообщение для ввода . В случае ошибки, выводит и запрашивает ввод заново.

Пользователь вводит два числа:

* – максимальное число в диапазоне от до
* – тип операции (0, 1 или 2), которую надо совершить. Если введено неверное число , то надо вывести сообщение об ошибке.

Вывести результат применения операции для всех чисел от 1 до включительно. Также выведите на экран количество палиндромов.

Снабжайте все выводы на экран вспомогательными текстовыми сообщениями.

Нужно реализовать повтор решения. Также можно реализовывать свои дополнительные методы для решения данной задачи.

**Проверьте себя:**

Тест 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| n = 20  type = 1 | Результат: 105  Количество палиндромов = 10 |

Тест 2

|  |  |
| --- | --- |
| Ввод | Вывод |
| n = 50  type = 2 | Результат: -2000  Количество палиндромов = 13 |

***Вариант 3***

Программа запрашивает у пользователя два целых положительных числа и выводит все целые положительные числа из интервала , у которых сумма цифр равна количеству цифр. Если таких чисел нет, то программа выводит все целые положительные числа из интервала , где сумма цифр в два раза больше количества цифр. Если и таких нет, программа сообщает об этом пользователю.

В программе должны быть реализованы три метода со следующими прототипами:

1. **static int ReadNumber(string message, string errorMessage);**
2. **static int CountSum(int number);**
3. **static int CountDigits(int number);**

Метод ReadNumber выводит пользователю сообщение message и считывает из консоли число. При неудачном преобразовании выводит пользователю сообщение errorMessage и требует повторить ввод. При удачном возвращает считанное число.

Метод CountSum возвращает сумму цифр числа number.

Метод CountDigits возвращает количество цифр числа number.

При несоответствии вводимых пользователем данных спецификации и здравой логике необходимо выводить осмысленное сообщение об ошибке и требовать повторить ввод. Требуется организовать повтор решения задачи.

**Проверьте себя:**

Тест 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| a = 10  b = 13 | Результат:  11 |

Тест 2

|  |  |
| --- | --- |
| Ввод | Вывод |
| a = 13  b = 18 | Результат:  13 |

***Вариант 4***

Пользователь вводит значение целого числа из интервала . Необходимо для каждого числа из интервала вывести на экран два возможных числа, составленных из четных и нечетных цифр исходного, соответственно (0 не учитывать, т.е. если введено число 409, то необходимо вывести на экран 9 и 4). В случае, если какое-либо число составить невозможно, вывести -1.

В программе должны быть реализованы три метода со следующими прототипами:

1. **static int ReadNumber(string message, string errorMessage);**
2. **static void** **FormNumbers(int number, out int even, out int odd);**
3. **static int CountDigits(int number);**

Метод ReadNumber выводит пользователю сообщение message и считывает из консоли число. При неудачном преобразовании выводит пользователю сообщение errorMessage и требует повторить ввод. При удачном возвращает считанное число.

Метод FormNumbers по числу number формирует два возможных числа: even – состоящее только из четных цифр и odd – число, состоящее только из нечетных цифр. В случае, если какое-либо число составить невозможно вместо него вернуть -1. Для подсчета количества цифр в числе использовать CountDigits.

Метод CountDigits возвращает количество цифр числа number.

При несоответствии вводимых пользователем данных спецификации и здравой логике необходимо выводить осмысленное сообщение об ошибке и требовать повторить ввод. Требуется организовать повтор решения задачи.

**Проверьте себя:**

Тест 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| n= 1000 | Результат:  число = 1000: 0; 1 |

|  |  |
| --- | --- |
| Ввод | Вывод |
| n= 999 | Результат:  число = 999: -1; 999  число = 1000: 0; 1 |