

2. Еднакви суми на четни и нечетни позиции

Напишете програма, която чете от конзолата **две шестцифрени цели числа** в диапазона от 100000 до 300000. Винаги **първото** въведено число ще бъде **по малко от второто**. На конзолата да се отпечата на **1 ред** **разделени с интервал** всички числа, които се намират **между двете**, прочетени от конзолата числа и отговарят на следното **условие**:

- сумата от цифрите на **четни** и **нечетни** позиции да са **равни**. Ако няма числа, отговарящи на условието на конзолата не се извежда резултат.

Примерен вход и изход

Вход	Изход	Обяснения			
100000 100050	100001 100012 100023 100034 100045	Първото число, което генерираме е числото 100000. Сумата от цифрите на четни позиции (жълто) е $0+0+0=0$. Сумата от цифрите на нечетни позиции (зелено) е $0+0+1=1$. Тъй като двете суми са различни числото не се отпечата. Следващото, число е 100001. Сумата на четни позиции е $1+0+0=1$, а на нечетни $0+0+1=1$. Двете суми са равни и числото се отпечата. Следващото число за проверка е 100002. То не отговаря на условието и не се отпечата. При числото 100045 сумата от четните позиции е $5+0+0=5$, а на нечетни $4+0+1=5$. Двете суми са равни числото се отпечата. И т.н.			
Вход	Изход	Вход	Изход	Вход	Изход
123456 124000	123464 123475 123486 123497 123530 123541 123552 123563 123574 123585 123596 123640 123651 123662 123673 123684 123695 123750 123761 123772 123783 123794 123860 123871 123882 123893 123970 123981 123992	299900 300000	299970 299981 299992	100115 100120	Няма изход

Насоки

- Прочетете входните данни от потребителя:

```
int firstNum = int.Parse(Console.ReadLine());  
int secondNum = int.Parse(Console.ReadLine());
```

- За да преминете през всички числа от интервала, направете **for** цикъл. След като сте прочели входните числа, задайте първото число за начална стойност на контролната променлива. Итерируйте до достигане на **второто число** като **увеличавате** стойността на контролната променлива с **1**:

```
int firstNum = int.Parse(Console.ReadLine());
int secondNum = int.Parse(Console.ReadLine());

for (int i = firstNum; i <= secondNum; i++)
{
    ...
}
```

3. Вземете числото на текущата позиция **като текст**, като използвате метода **.ToString()**:

```
for (int i = firstNum; i <= secondNum; i++)
{
    string currentNum = i.ToString();
    ...
}
```

4. За да обходите всяка цифра от числото, направете **for** цикъл. След като е взето като текст, вземете неговата дължина с **.Length**. Итерируйте до достигане на **дължината на числото** като **увеличавате** стойността на контролната променлива с **1**:

```
string currentNum = i.ToString();
for (int j = 0; j < currentNum.Length; j++)
{
    ...
}
```

5. Продължете към дописване на логиката за намиране на сумата на четна и нечетна позиция за всяко число. Декларирайте по една променлива за четната и нечетна сума. За да получите точната числова стойност на цифрите използвайте метода **int.Parse()**.

```
for (int i = firstNum; i <= secondNum; i++)
{
    string currentNum = i.ToString();
    int oddSum = 0;
    int evenSum = 0;
    for (int j = 0; j < currentNum.Length; j++)
    {
        int currentDigit = int.Parse(currentNum.ToString());
        ...
    }
}
```

6. За да намерите цифрите, които се намират на четна позиция, използвайте условна **if** конструкция, проверете дали индекса му е четно число като го разделите модулно на 2(**index % 2**), ако е четно,

добавете го към сумата на четните, ако не е, към сумата на нечетните.

```
for (int i = firstNum; i <= secondNum; i++)
{
    string currentNum = i.ToString();
    int oddSum = 0;
    int evenSum = 0;
    for (int j = 0; j < currentNum.Length; j++)
    {
        int currentDigit = int.Parse(currentNum[j].ToString());
        if (j % 2 == 0)
        {
            evenSum += currentDigit;
        }
        else
        {
            oddSum += currentDigit;
        }
    }
}
```

7. След като сте намерили сумата на цифрите на четни и нечетни позиции, проверете дали са равни, ако са, принтирайте числото. Програмата ви трябва да изглежда по следния начин:

```
int firstNum = int.Parse(Console.ReadLine());
int secondNum = int.Parse(Console.ReadLine());

for (int i = firstNum; i <= secondNum; i++)
{
    string currentNum = i.ToString();
    int oddSum = 0;
    int evenSum = 0;
    for (int j = 0; j < currentNum.Length; j++)
    {
        int currentDigit = int.Parse(currentNum[j].ToString());
        if (j % 2 == 0)
        {
            evenSum += currentDigit;
        }
        else
        {
            oddSum += currentDigit;
        }
    }
    if (oddSum == evenSum)
    {
        Console.Write(i + " ");
    }
}
```