

3. Комбинации

Напишете програма, която изчислява **колко решения в естествените числа** (включително и нулата) има уравнението:

$$x1 + x2 + x3 = n$$

Числото n е цяло число и се въвежда от конзолата.

Примерен вход и изход

Вход	Изход	Обяснения	Вход	Изход	Вход	Изход
25	351	Генерираме всички комбинации от 3 числа, като първата е: $0+0+0=0$, но понеже не е равна на 25, продължаваме: $0+0+1=1$ – също не е 25 и т.н. Стигаме до първата валидна комбинация: $0 + 0 + 25 = 25$, увеличаваме броя на валидни комбинации с 1, втората валидна комбинация е: $0 + 1 + 24 = 25$ Третата: $0 + 2 + 23 = 25$ и т.н. След генериране на всички възможни комбинации, броят на валидните е 351.	20	231	5	21

Насоки

1. Прочетете входните данни – едно цяло число, въведено от потребителя и го запаметете в променлива:

```
int n = int.Parse(Console.ReadLine());
```

2. Създайте 3 вложени for-цикъла, с които да итерирате всяка възможна стойност на едно от 3те числа в уравнението:

```
int n = int.Parse(Console.ReadLine());

for (int x1 = 0; x1 <= n; x1++)
{
    for (int x2 = 0; x2 <= n; x2++)
    {
        for (int x3 = 0; x3 <= n; x3++)
        {
        }
    }
}
```

3. Направете проверка в най-вътрешния вложен цикъл за стойностите на $x1$, $x2$, $x3$ във всяка една итерация. За да бъде валидно уравнението, техният сбор трябва да е равен на n . Създайте променлива **validCombinationsCount**, която да пази броя на валидните комбинации и добавяйте към нея всеки път, когато генерирате такава:

```
int n = int.Parse(Console.ReadLine());

for (int x1 = 0; x1 <= n; x1++)
{
    for (int x2 = 0; x2 <= n; x2++)
    {
        for (int x3 = 0; x3 <= n; x3++)
        {
            if (x1 + x2 + x3 == n)
            {
                // TODO: Add to valid combination counter
            }
        }
    }
}
```

4. Накрая принтирайте броя на валидните комбинации (**validCombinationsCount**).