

Sherlock's Mystery

In this problem basically we have to find all the combinations of X, Y, Z such that $(X+Y+Z)\%2==0$ and $X \neq Y \neq Z$ and besides there should be **no repetition** in the pairs of any kind.

The possible combinations of $(X+Y+Z)\%2==0$ will only result to be true when X, Y and Z are all even or when any two of them are odd.

This means :- **ALL EVEN**

OR

TWO ODD & ONE EVEN

For this firstly we have to find the number of odd numbers and even numbers within the given range i.e. from 1 to N (inclusive).

Even numbers = $\text{number}/2$;

All left out numbers are odd = $\text{number} - \text{even numbers}$

Formula to be used:- $nCr = \frac{n!}{r! * (n-r)!}$

Where r is the number of items being chosen at a time

Now all possible pairs of 3 even numbers without repetition :-

$$(\text{Even numbers})! / (3! * (\text{Even numbers}-3)!)$$

$(\text{Even numbers}-3)!$ Means all factorials of the number will be present till $(\text{even numbers}-3)$

Therefore no need to find the factorials separately. We just need to multiply :- $((\text{even numbers}) * (\text{even numbers}-1) * (\text{even numbers}-2)) / (3*2)$

$$3! = 3*2*1$$

Now for all pairs of 2 odd and one even number (without repetition):-

We can find the number of pairs of 2 of odd numbers and multiply each pair by number of even numbers as each pair gets to go with each even number.

$$[(\text{Odd numbers})! / (2! * (\text{Odd numbers}-2)!)] * (\text{even numbers})$$

$(\text{Odd numbers}-2)!$ Means all factorials of the number will be present till $(\text{odd numbers}-2)$

Therefore no need to find the factorials separately. We just need to multiply :- $((\text{odd numbers}) * (\text{odd numbers}-1)) / 2$

Now multiply the result by the total number of even numbers.

$2!=2*1$

- The number can be large so we have to use *long*.

CODE :-

```
import java.util.*;
import java.io.*;
public class Mystery
{
    public static void main (String[] args) throws java.lang.Exception {

        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while (t-- > 0)
        {
            int n = sc.nextInt();
            int even = n / 2;
            int odd = n - even;

            long extra = (long)((odd * (odd - 1)) * even) / 2;
            long extra1 = (long)(even * (even - 1) * (even - 2)) / 6;
            System.out.println(extra + extra1);
        }
    }
}
```