Problem 1. Find the sum of all the multiples of 3 or 5 below 1000.

As you have solved this problem we do not have to explain that all numbers divisible by 3 and/or by 5 should be counted.
So the first numbers to be added would be:
3, 5, 6, 9, 10, 12, 15 and so on.

A simple way to do this is to go through all numbers from 1 to 999 and test whether they are divisible by 3 or by 5.
This would result in code like:

```
target=999
sum=0
for  i=1 to target do
if (i mod 3=0) or (i mod 5)=0 then sum:=sum+i
output sum
```

(In some programming languages the `mod` operator is written as %)
Simple enough you might say.
But wait a minute: if we had asked to do the same for all numbers less than 1,000,000,000 that is going to take quite a while. Perhaps you would like to try out that first (make sure your sum variable does not overflow).

To get a more efficient solution you could also calculate the sum of the numbers less than1000 that are divisible by 3, plus the sum of the numbers less than1000 that are divisible by 5. But as you have summed numbers divisible by 15 twice you would have to subtract the sum of the numbers divisible by 15.

If we now define a function:

```
Function SumDivisibleBy(n)
    Details to be filled in
EndFunction
```

Then the answer would be

```
SumDivisibleBy(3)+SumDivisibleBy(5)-SumDivisibleBy(15)
```

Let's look at the details of our function and take as example n=3.
We would have to add:
3+6+9+12+.....+999=3*(1+2+3+4+...+333)
For n=5 we would get:
5+10+15+...+995=5*(1+2+....+199)
Now note that 199=995/5 but also 999/5 rounded down to the nearest integer.
In many programming languages there exists a separate operator for that: `div` or `\.`
If we now also note that 1+2+3+...+p=½*p*(p+1) our program becomes:

```
target=999

Function SumDivisibleBy(n)
  p=target div n
  return n*(p*(p+1)) div 2
EndFunction

Output SumDivisibleBy(3)+SumDivisibleBy(5)-SumDivisibleBy(15)
```


Author: hk