

Lesson:



Problems based on Recursion – 3



Pre-Requisites

- Recursion basics
- Working rules of recursive functions

List of Concepts Involved

- Given a number num and a value k. Print k multiples of num.
- Given a number n. Find the sum of natural numbers till n but with alternate signs.

Problem 1: Given a number num and a value k. Print k multiples of num.

Constraints : $k > 0$

Input 1: num = 12 , k = 5

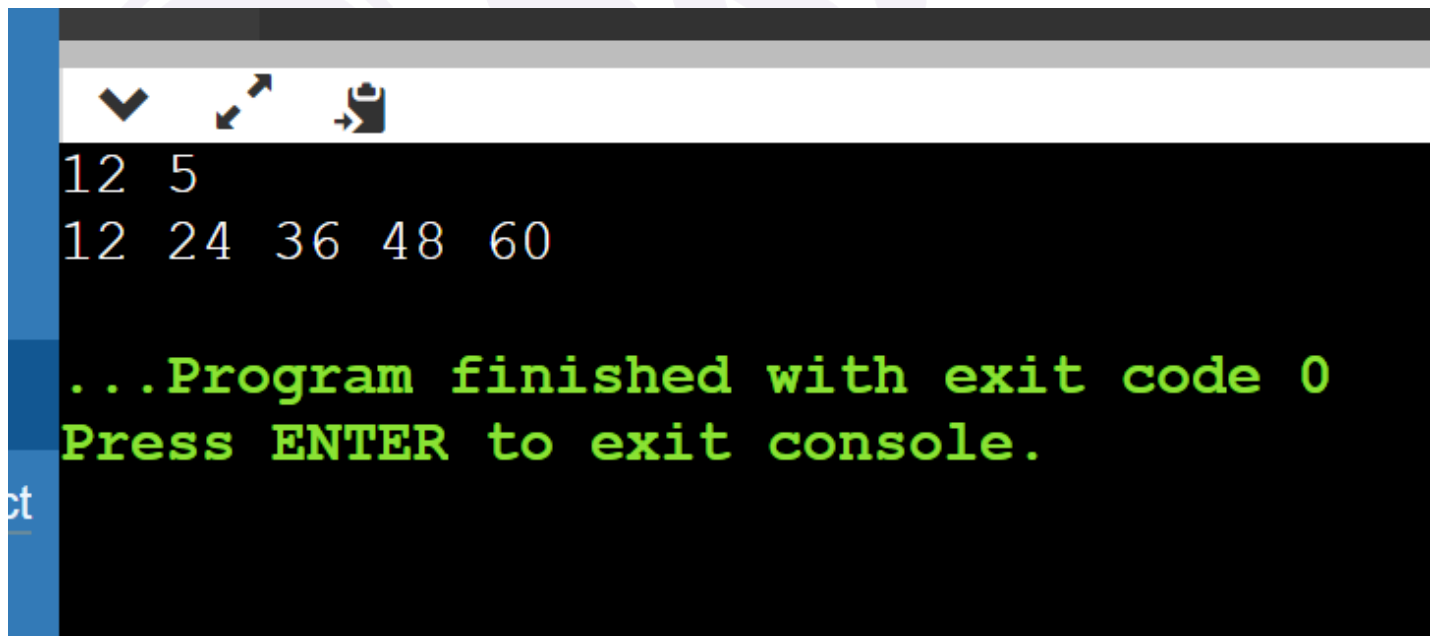
Output 1: 12 , 24 , 36 , 48 , 60

Input 2: num = 3 , k = 8

Output 2: 3 , 6 , 9 , 12 , 15 , 18 , 21 , 24.

Solution:

Code : <https://pastebin.com/DVF49AV>



```
12 5
12 24 36 48 60

...Program finished with exit code 0
Press ENTER to exit console.
```

- Here, we have created a go function of void type to print the k multiples of given number in which we have passed 3 parameters **n**(number whose multiple user wants), **k**(number of multiples that user want), and **i**(variable just to increment the multiplier in every new function call).
- Here the base case condition is if the value of $k = 0$ i.e we need no more multiples of n , so we need to return from this function.
- If the base case condition is not satisfied i.e $k > 0$, then we can only print the current value(i.e $n*i$) and then we can recursively call the function to print other multiples of n since we do not know the answer for them.
- The only thing that we know here is our own value i.e $(n*i)$.
- The variable i is just to maintain the count of multiple like $n*1$ then in next iteration $n*2$, $n*3$ and so on.

Problem 2 : Given a number n . Find the sum of natural numbers till n but with alternate signs.

That means if $n = 5$ then you have to return $1-2+3-4+5 = 3$ as your answer.

Constraints : $0 \leq n \leq 1e6$

Input1 : $n = 10$

Output 1 : -5

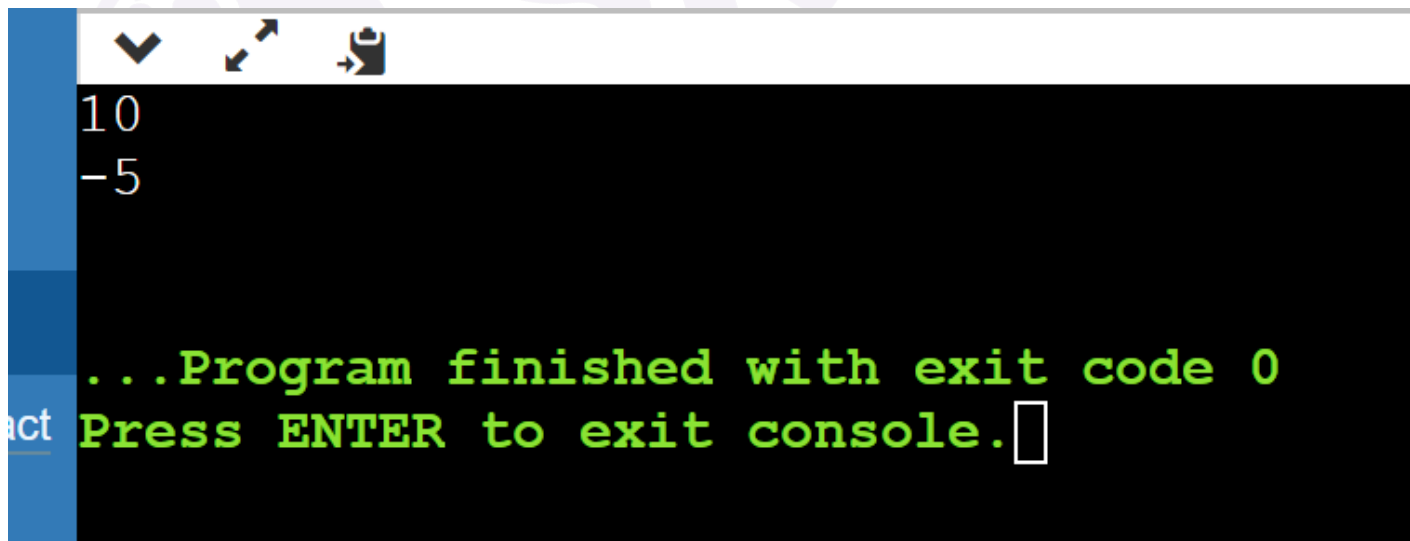
Explanation : $1-2+3-4+5-6+7-8+9-10 = -5$

Input 2 : $n = 5$

Output 2 : 3

Solution:

Code : <https://pastebin.com/ewX39MQg>



```

10
-5

...Program finished with exit code 0
Press ENTER to exit console.

```

- Here, we have created a go function of int type to get the sum of natural numbers with alternate signs till given n .
- We have passed 2 parameters , n (the number till you want the alternating sequence sum) , a variable i (to keep track of odd and even numbers) .

- If i has reached $n+1$ (base case condition) that means we have already calculated the sum of the sequence till n so we returned from this function.
- If this is not the case (i.e. $i \leq n$) we can clearly observe from the sequence that every even number of this sequence is added with a negative polarity while every odd number is added with positive polarity.
- We have done the same as we observed in the pattern. For every ' i ' we check if i is odd then we return the sum of (i) and value returned by $go(n, i+1)$ and if i is even then we return the sum of $(-i)$ and value returned by $go(n, i+1)$. This is how we are recursively calling the go function.

Upcoming Class Teasers

- Problems based on recursion.

