info@codeanddebug.in —— Code and Debug —— codeanddebug.in

WEEK 2 - ASSIGNMENT 3
BASIC WHILE LOOP

NOTE:

- **No need to submit anywhere, just keep track of all the PDF you made in a specific folder.**
- **Compare your solution with the solution I'll provide, in case of doubts, kindly reach out to me.**
- **You may get assignment solution in format of PDF or VIDEO solution, depending on the difficulty level.**

Q1. Use a while loop to calculate the sum of the first 10 natural numbers.

Q2. Ask a positive number from user. Print all the numbers from **positive number** to 1.

Example n = 7

Output: 7 6 5 4 3 2 1

Q3. Print this pattern using while loop.

1 8 15 22 29 36 43 50 57 64 71 78 85 92 99

Q4. Print all odd numbers less than 15 using a while loop.

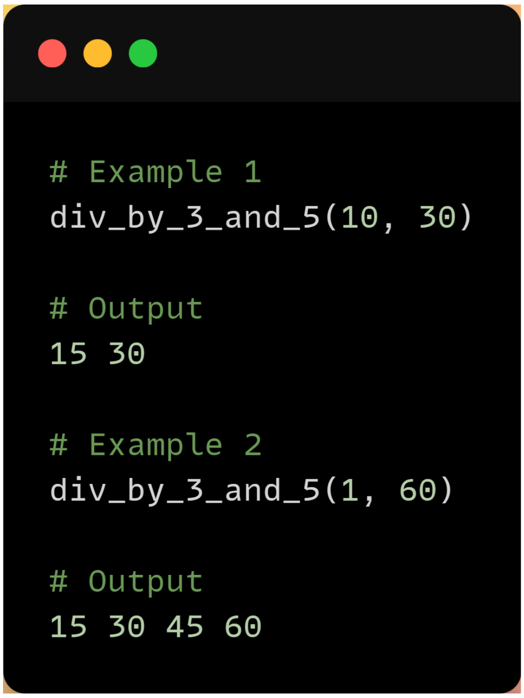
Q5. Calculate the factorial of a given number using a while loop.

Example, n = 5

5 x 4 x 3 x 2 x 1

Output: 120

Q6. Create a function named **div_by_3_and_5** which takes 2 integers as a arguments **(n1,n2)**, and print all the numbers divisible by 3 and 5 between n1 and n2.



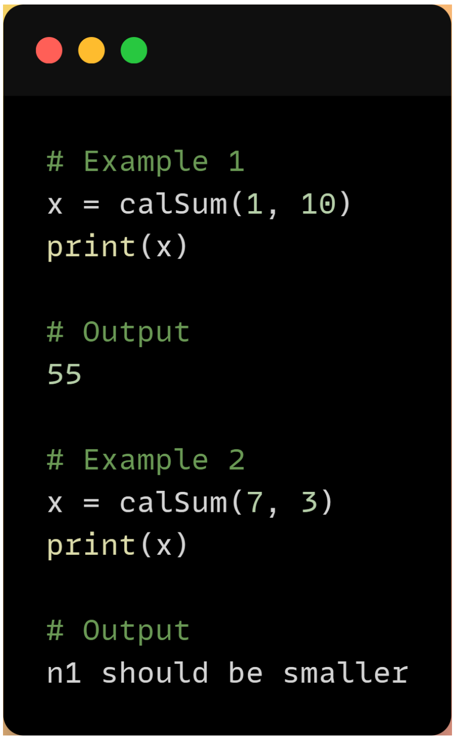
```
# Example 1
div_by_3_and_5(10, 30)

# Output
15 30

# Example 2
div_by_3_and_5(1, 60)

# Output
15 30 45 60
```

Q7. Create a function named **calSum()** which takes 2 integers **n1** and **n2** as a argument. Calculate the sum of all the numbers from **n1** and **n2** and **RETURN THAT SUM.** Also make sure that **n1 is smaller than n2**. If it is not, then return **“n1 should be smaller”.**



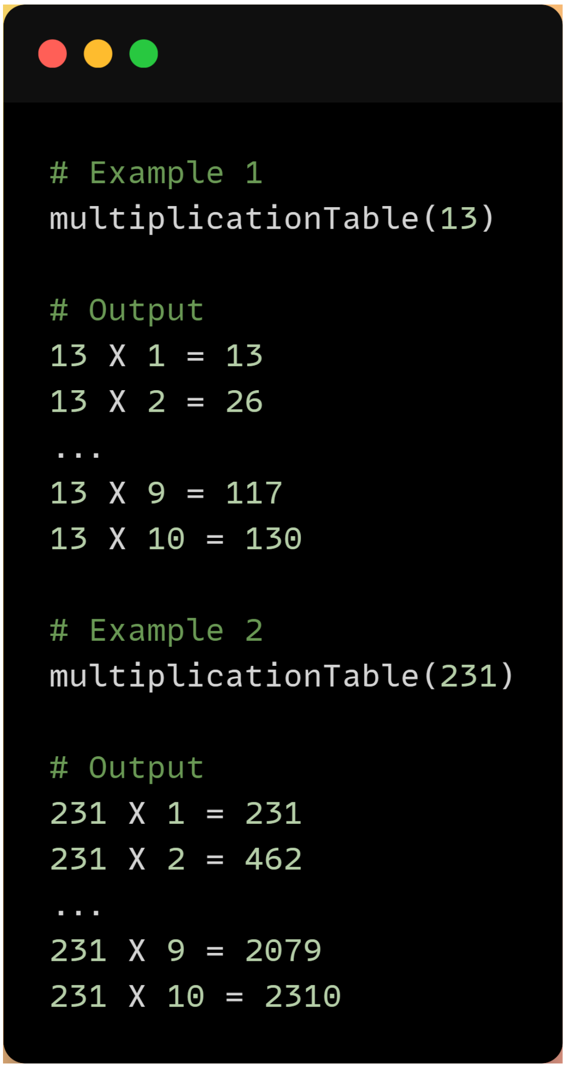
```
# Example 1
x = calSum(1, 10)
print(x)

# Output
55

# Example 2
x = calSum(7, 3)
print(x)

# Output
n1 should be smaller
```

Q8. Create a function named **multiplicationTable** that takes an integer **num** as an argument. Print the multiplication table of that number up to 10 numbers.



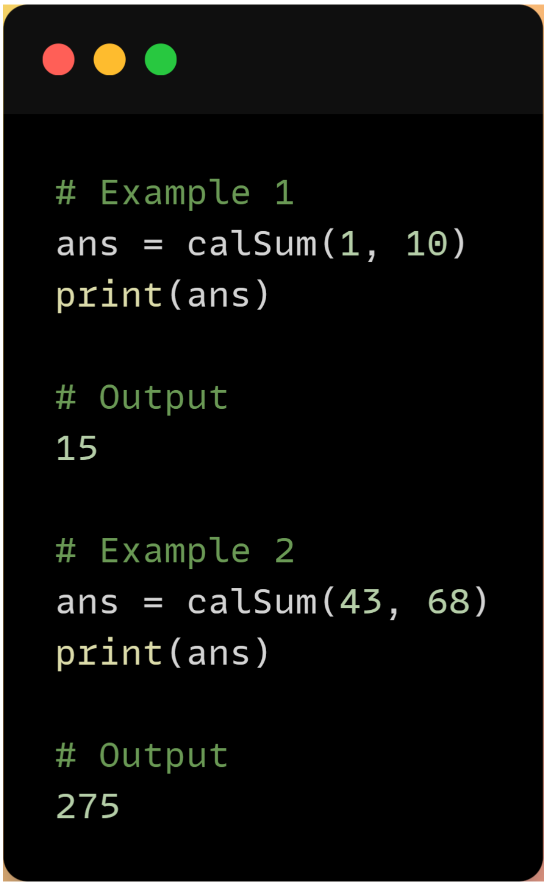
```
# Example 1
multiplicationTable(13)

# Output
13 X 1 = 13
13 X 2 = 26
...
13 X 9 = 117
13 X 10 = 130

# Example 2
multiplicationTable(231)

# Output
231 X 1 = 231
231 X 2 = 462
...
231 X 9 = 2079
231 X 10 = 2310
```

Q9. Create a function named **calSum** which takes an 2 integer **(n1 and n2)** as an argument. Calculate the sum of all the numbers **divisible by 5** between **n1 and n2** and **return** that sum. (Make sure that n1 is less than n2).



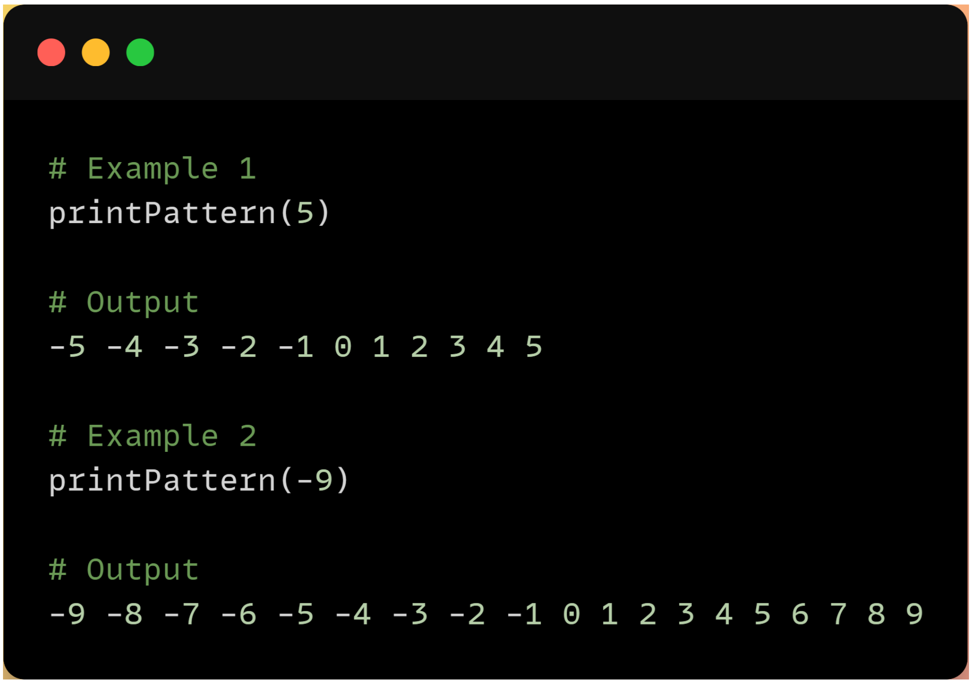
```
# Example 1
ans = calSum(1, 10)
print(ans)

# Output
15

# Example 2
ans = calSum(43, 68)
print(ans)

# Output
275
```

Q10. Create a function named **printPattern** that takes one integer **(num)** as an argument. Print from **-num** to **num.** Also keep in mind number passed as an argument can be negative or positive.



```
# Example 1
printPattern(5)

# Output
-5 -4 -3 -2 -1 0 1 2 3 4 5

# Example 2
printPattern(-9)

# Output
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
```