

Home Task 1:

Write a Python code for the following:

- Ask the user to enter a Number, **N**.
- From **1 to N (inclusive)**, display the summation of all numbers that are multiples of **either 7 or 9 but not a multiple of both**.

=====

Sample Input 1:

30

Sample Output 1:

124

Explanation: The summation of multiples of either 7 or 9 but not both from 1 up to 30 is $7 + 9 + 14 + 18 + 21 + 27 + 28 = 124$

=====

Sample Input 2:

75

Sample Output 2:

385

Explanation: The summation of multiples of either 7 or 9 but not both from 1 up to 75 is $7 + 9 + 14 + 18 + 21 + 27 + 28 + 35 + 36 + 42 + 45 + 49 + 54 + 56 + 70 + 72 = 583$

Home Task 2:

Write a Python program that takes a number from the user and prints its digits from **left to right**.

[Consider the input number to be an INTEGER. You are not allowed to use String indexing for solving this task]

Example: if the user gives 32768, then print 3, 2, 7, 6, 8

=====

Hint(1): The input() function takes the input data as String data type by default. Please convert it to an integer before starting your code for the task.

Hint(2):

Step 1: First, count how many digits are there in the input number

Step 2: Then, calculate 10 to the power (number of digits - 1).

Step 3 with explanation: Say, the input given by the user as in our example, 32768 has 5 digits, so calculating 10 to the power 4 gives us 10000. Then floor dividing 32768 by 10000, we can get 3.

Proceeding further, the remainder of 32768 by 10000 ($32768 \% 10000$) gives us 2768. This time to get 2 we need to floor divide 2768 by 1000 which is basically our $10000/10$. Again, taking the remainder of 2768 by 1000 gives us 768 which we then divide by 100 (i.e. $1000/10$) and keep on doing this until there are no more digits left (zero).

To summarize and clarify:

Loop 1: First, we count digits, say 5 in this case for 32768

Loop 2: Then, we calculate 10 to the power 4 (5-1), that is 10000.

Loop 3: Then we keep repeating the three steps of floor dividing, modulus and dividing by 10 as demonstrated below.

$32768 // 10000 = 3$

$32768 \% 10000 = 2768$

$10000 // 10 = 1000$

$2768 // 1000 = 2$

$2768 \% 1000 = 768$

$1000 // 10 = 100$

$768 // 100 = 7$

$768 \% 100 = 68$

$100 // 10 = 10$

$68 // 10 = 6$

$68 \% 10 = 8$

$10 // 10 = 1$

$8 // 1 = 8$

$8 \% 1 = 0$

$1 // 10 = 0$

Done. Loop ends as the number has become 0.

Home Task 3:

Write a Python program that asks the user for an integer number and tells if it is a prime number or a perfect number or neither.

Note: A number cannot be both prime and perfect.

Prime Number: If a number has only two divisors, (1 and itself), then it is a prime number. Else, then it is not a prime number.

Perfect Number: A number is said to be a perfect number if the sum of its divisors, including 1 but not the number itself is equal to that number.

Hint: You may take help from Class Task 4 for finding the divisors.

Sample Input 1:

6

Sample Output 1:

6 is a perfect number

Explanation:

6 has 4 divisors: 1, 2, 3, and 6.

If we add all divisors of 6 except 6 itself, $1 + 2 + 3 = 6$. The sum of the divisors excluding the number itself sums up to the number, therefore "6 is a perfect number" is printed.

Sample Input 2:

11

Sample Output 2:

11 is a prime number

Explanation: 11 has only 2 divisors: 1 and 11.

Sample Input 3:

33

Sample Output 3:

33 is not a prime or perfect number

Explanation:

33 has 4 divisors: 1, 3, 11, and 33. So it is not a prime number.

If we add all divisors except 33 itself, $1 + 3 + 11 = 15$. The sum is not equal to the number, therefore, 33 is not a perfect number.

Home Task 4:

Write a Python program that asks the user for a quantity, then takes that many numbers as input and prints the maximum, minimum and average of those numbers.

[Please note that you CANNOT use max, min built-in functions]

[Also, you DO NOT need to use lists for this task]

=====

Example: If the user enters 5 as an input for quantity and the enters the 5 numbers, 10, 4, -1, -100, and 1.

The output of your program should be: "Maximum 10", "Minimum -100", "Average is -17.2" as shown below.

Input:

5

10
4
-1
-100
1

Output:

Maximum 10
Minimum -100
Average is -17.2

Explanation: Average calculation: $(10 + 4 + (-1) + (-100) + 1)/5 = -86/5 = -17.2$

Home Task 5:

Write a python program that prints a right-angled triangle of height N using incrementing numbers where N will be given as input.

Hint: You may need to use nested loops. Try to think up to which point the inner loop should run.

=====

Sample Input 1:

4

Sample Output 1:

1
12
123
1234

Explanation: For an input of 4, we have 4 rows/lines where in each line, the respective column number is printed sequentially up to the line/row number. So, in line number 1, we have 1 only. In line 2, 12 is printed. In line 3, we have 123 and so on.

=====

Sample Input 2:

5

Sample Output 2:

1
12
123
1234
12345

Explanation: Numbers are printed sequentially up to the line number for each of the lines.