

# DSML Module Test: Beginner Python 1 Practice Test

Question 1:

Sum of Digits (II)

## Problem Description

Take **T** (number of test cases) as input.

For each test case, take integer **A** as input and print the sum of digits of that number.

</div>

## Problem Constraints

$1 \leq T \leq 10$

$1 \leq A \leq 109$

## Input Format

The first line of the input contains an integer **T**, which denotes the number of test cases.

Each of the next **T** lines contains an integer **A**.

## Output Format

Print **T** lines where ith line denotes the answer of ith case (i.e. integer denoting the sum of digits of the number **A**)

## Example Input

```
2
46
11
```

## Example Output

```
10
2
```

## Example Explanation

Sum of digits in 46 = 4 + 6 = 10

Sum of digits in 11 = 1 + 1 = 2

### User Code

```
def main():
    T = int(input())
    while T > 0:
        sum = 0
        A = input()
        for element in A:
            sum += int(element)
        print(sum)
        T -= 1

    return 0

if __name__ == '__main__':
    main()
```

### Question 2:

prime numbers less than or equal to n

Write a function to find the sum of all the prime numbers less than or equal to a given positive integer n. The function should take an integer n as input and return the sum as an integer.

### Input format:

int

### Output format:

int

### Sample input:

10

### Sample output:

17

### Sample Explanation:

Given n = 10

Possible prime number less than or equal to 10 are:

2, 3, 5, 7 and they all sum up to 17

### User Code

```
def sum_of_primes(n):
    sum = 0
    if n < 2:
        return 0
    for element in range(2, n+1):
        is_prime = True
        for j in range(2, int(element**0.5)+1):
            if element % j == 0:
                is_prime = False
                break
        if is_prime:
            sum += element
```

```
return sum
```

Question 3:

Age of tree II

The age of a tree can be determined by looking at the lines inside its bark. Write a function that takes an input integer lines and classify the tree based on the following criteria:

- If lines are greater than **20** print Old
- If lines are between **10** and **20**, **both inclusive**, print Not too old
- If lines are between **2** and **9**, **both inclusive**, print Just became big
- If lines are **strictly** less than **2** print Started growing

**NOTE: Do not return anything from the function**

### Input Format

Single line input containing an integer

### Output Format

String based on the criteria given in the problem description

### Example Input

Input 1:

21

Input 2:

15

Input 3:

1

### Example Output

Output 1:

Old

Output 2:

Not too old

Output 3:

Started growing

### Example Explanation

Explanation 1:

Since lines are greater than 20 Old is printed

Explanation 2:

Since lines are between 10 and 19 Not too old is printed

Explanation 3:

Since lines are less than 2 Started growing is printed

### User Code

```
def tree_age(lines):  
    if lines < 2 :  
        print("Started growing")  
    elif lines >=2 and lines <= 9:  
        print("Just became big")
```

```
elif lines >=10 and lines <=20:  
    print("Not too old")  
else:  
    print("Old")
```

Question 4:

Who's There?

What should be the input for the value of num in the code below so that the output of this code is Hello, this is Raj?

```
num = int(input())  
val = 0  
for i in range(2, num):  
    val = val + i  
if val > 10:  
    print('Hello, this is Raj')  
else:  
    print('There is no one')
```

**6**

Question 5:

Reduce the num

What should be the input for value of num in the code below so that the value of num becomes 1 at the end of the execution of the while loop?

```
num = int(input())  
while num > 1:  
    num = num // 3  
print(num)
```

**89**

Question 6:

Knock Knock

Which of the following code snippets from the options given will give the output as below?

```
Knock Knock  
Who's There?  
No One
```

**C**

```
if True:
    print("Knock Knock")
if True:
    print("Who's There?")
if True:
    print("No One")
```

**Question 7:**

**right about if**

You have been given the following piece of code. Assume that x has already been declared.

```
if x > 5:
    x = x*3
if x > 15:
    x = 0
print(x)
```

**For  $X < 5$ , the output is initial value of X.**