

S1. Coding Questions

Q1. 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.

Problem Constraints

$1 \leq T \leq 10$

$1 \leq A \leq 10^9$

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 i^{th} line denotes the answer of i^{th} 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():
    # YOUR CODE GOES HERE
    # Please take input and print output to standard input/output (stdin/stdout)
    # E.g. 'input()/raw_input()' for input & 'print' for output
    T = int(input())
    for i in range(T):
        A = int(input())
```

```

        total = 0
        while A > 0:
            remainder = A % 10
            total = total + remainder
            A = A // 10

        print(total)

if __name__ == '__main__':
    main()

```

Q2. 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):
    # write your code here
    def is_prime(num):
        if num < 2:
            return false
        for i in range(2,int(num ** 0.5)+1):
            if num % i == 0:
                return False
        return True

```

```
total = 0
for num in range(2,n+1):
    if is_prime(num):
        total += num
return total
```

Q3. 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):
    # write your code here
    if lines > 20:
        print('Old')
    elif lines >= 10 and lines <= 20:
        print('Not too old')
    elif lines >= 2 and lines <= 9:
        print('Just became big')
    else:
        print('Started growing')
```

S2. Multiple Choice Questions

Q1. 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')
```

- 5
- 3
- 6
- 2
- 4

Q2. 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)
```

- 2
- 89
- 72
- 8

Q3. Which of the following code snippets from the options given will give the output as below?

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

Options:

A.

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

B.

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

C.

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

D.

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

- A
- B
- C
- D

Q4. 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)
```

- Output will always be equal to 0
- For $x > 5$, the output is thrice the initial value of x
- For $x < 5$, the output is the initial value of x
- For $x > 5$ and $x < 15$, output is thrice the initial value of x