## 1.1.2. Conditional Calculation Based on the Number of Digits



Write a Python program that accepts an integer n as input. Depending on the number of digits in n.

### Constraints: $1 \le n \le 999$

### Input Format:

The input consists of a single integer n.

**Output Format:** 

## If n is a single-digit number, print its square.

If n is a two-digit number, print its square root (rounded to two decimal places).

If n is a three-digit number, print its cube root (rounded to two decimal places). Else print "Invalid".

n=int(input()) if(0<n<10): 10 11

> Terminal

condition...

print(n\*n) elif(10<=n<100): p=n\*\*0.5 print( "%0.2f"%p) \_ elif(100<=n<=999): r=n\*\*(1/3) print("%.2f"%r) , else: print("Invalid")

Sample Test Cases



### 1.1.3. Age and Salary Calculation



birthDate...

Write a Python program that reads the birth date and salary of employees.

A floating-point number representing the salary of the employee in rupees.

## Input Format:

The input consists of: A string representing the birth date of the employee in the format DD - MM - YYYY.

### **Output Format:**

The output should include:

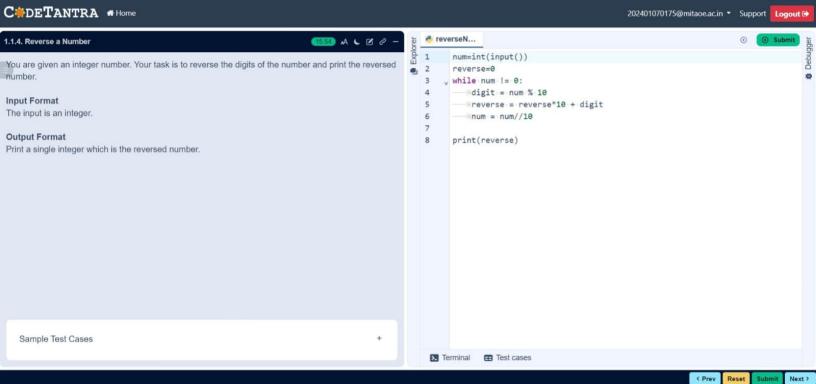
The age of the employee.

The salary of the employee in dollars.

### Note:

1INR=0.012USD

Explorer def calculate age(birthdate): date object = datetime.strptime(birthdate. "%d-%m-%Y") today = datetime.today() if ((today.month, today.day ) < (date object.month, date object.day)): age = today.year-date\_object.year-((today.month, today.day ) (date object.month, date object.day)) > return age elif((today.month, today.day) >> (date object.month, date object.day)): 10 > age = today.year-date\_object.year- ((today.month, today.day) > (date object.month, date object.dav)) 11 return age 12 13 14 def convert salary to dollars(salary in rupees): 15 salary=salary\_in\_rupees\*0.012 16 return salary 17 18 birthdate = input() 19 salary in rupees = float(input()) 20 age = calculate\_age(birthdate) 21 salary\_in\_dollars = convert\_salary\_to\_dollars(salary\_in\_rupees) print(f"Age: {age}") print(f"Salary in dollars: {salary in dollars: 2f}") 



1.2.1. Pass or Fail ALBO-

Write a Python program that accepts the number of courses and the marks of a student in those courses.

The grade is determined based on the aggregate percentage:

- · If the aggregate percentage is greater than 75, the grade is Distinction.
- · If the aggregate percentage is greater than or equal to 60 but less than 75, the grade is First Division.
- · If the aggregate percentage is greater than or equal to 50 but less than 60, the grade is Second
- **Division** If the aggregate percentage is greater than or equal to 40 but less than 50, the grade is Third

## Input Format:

Division.

The first input will be an integer n, the number of courses.

The second input will be n integers representing the marks of the student in each of the n courses. separated by a space.

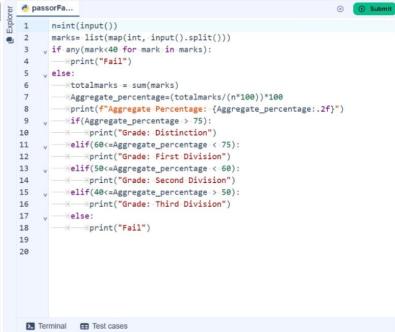
### **Output Format:**

If the student passes all courses:

Sample Test Cases

- · Print the aggregate percentage (rounded to two decimal places).
- · Print the grade based on the aggregate percentage.

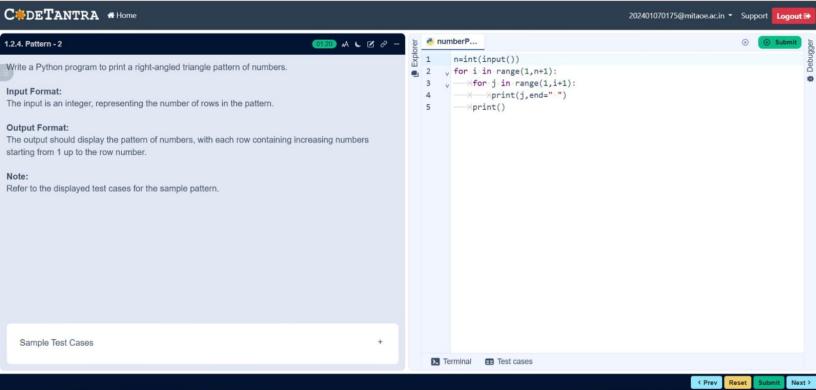
If the student fails any course (marks < 40 in any course), print:



Logout @

Submit

Next >





# Instructions

Sample Test Cases

· Your input and output must follow the input and output layout mentioned in the visible sample test case.

· Hidden test cases will only pass when users' input and output match the expected input and output.

+

return 0

elif n == 1: return 1

else: return fib(n-1) + fib(n-2)

for i in range (n):

⊞ Test cases

print(fib(i),end=" ")

n=int(input("Enter terms for Fibonacci series: "))







< Prev Reset Submit Next >









9

10

11 12

13

14

15

17

> Terminal









if n <= 0:





