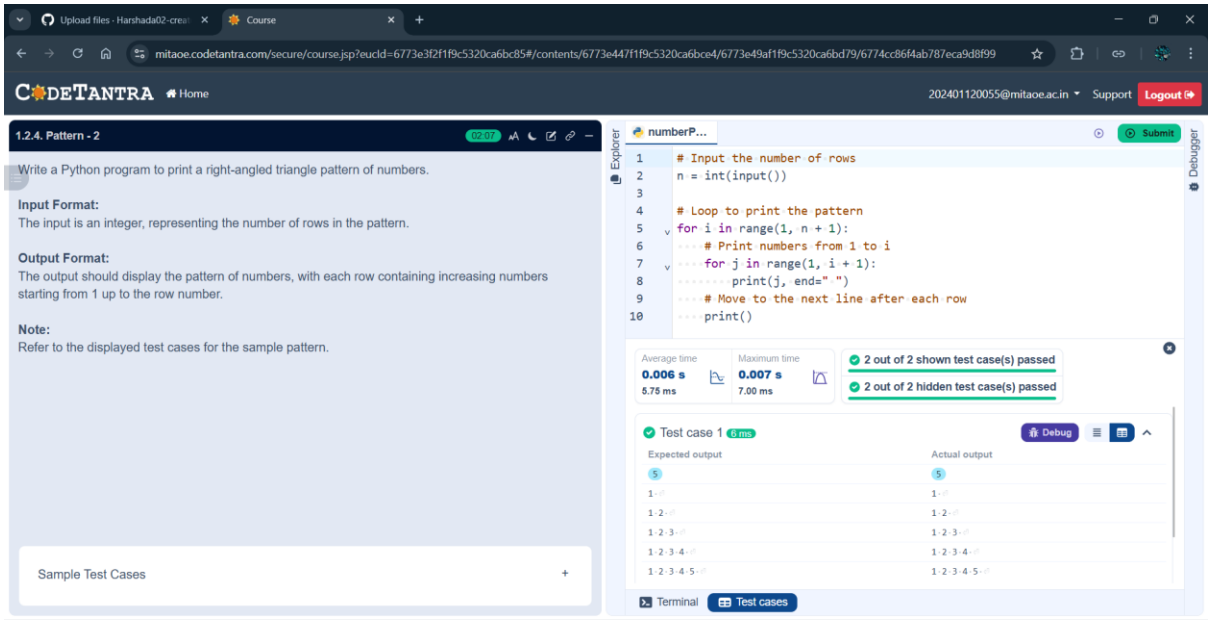
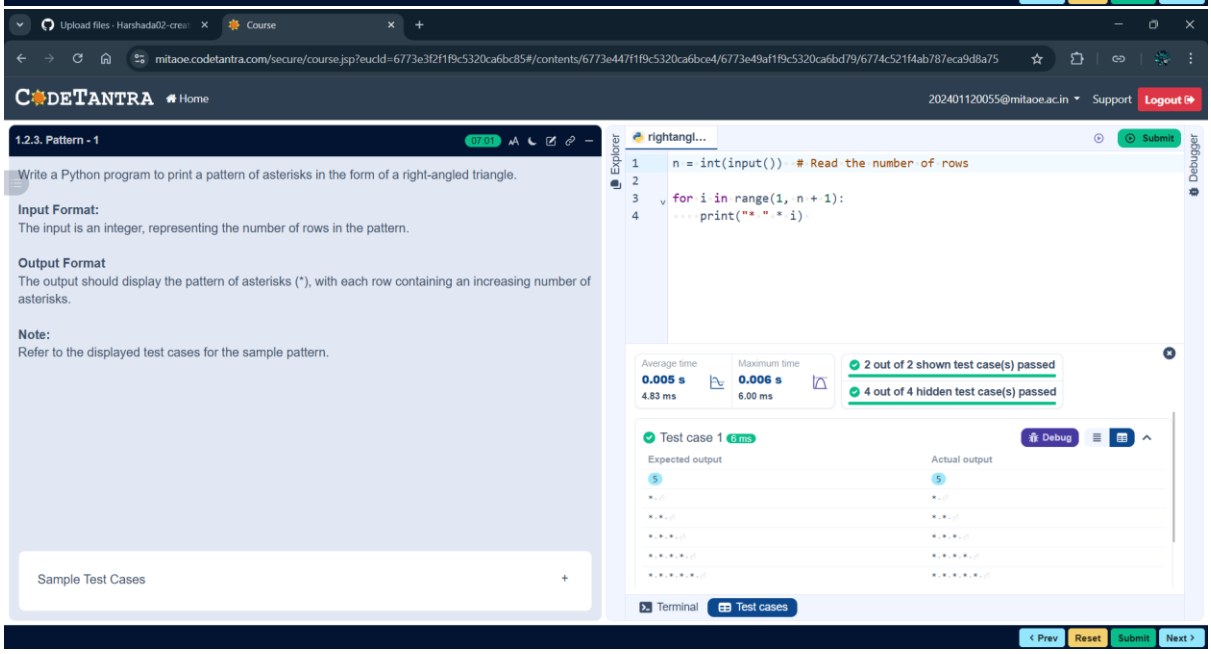
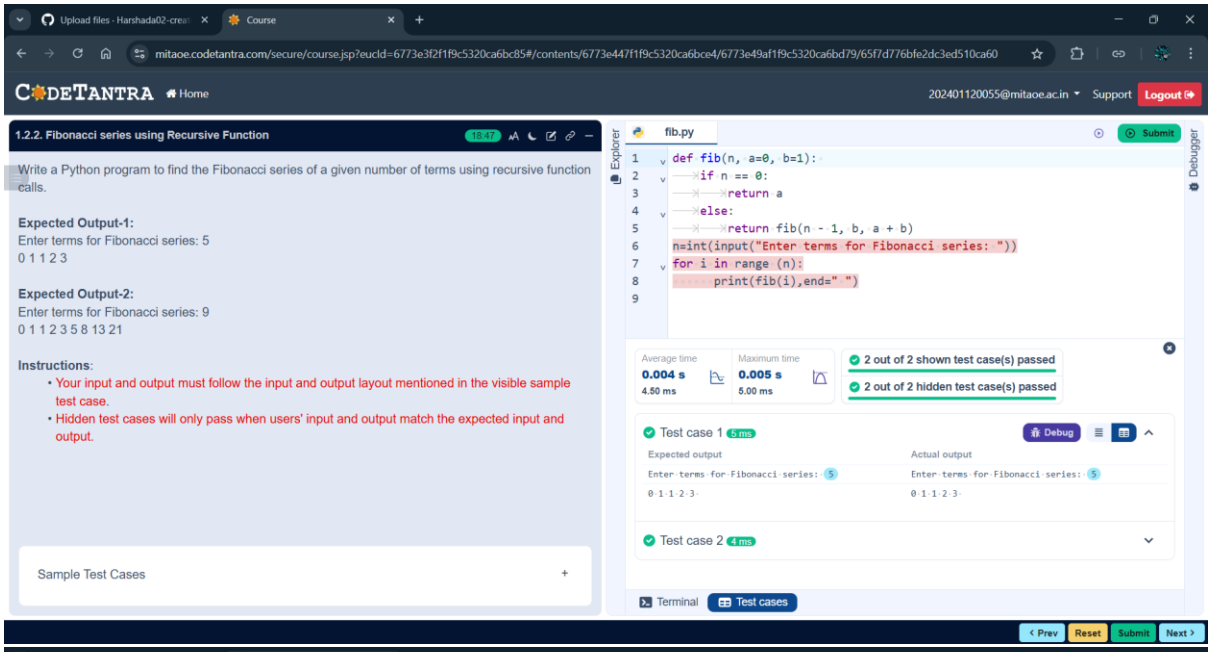
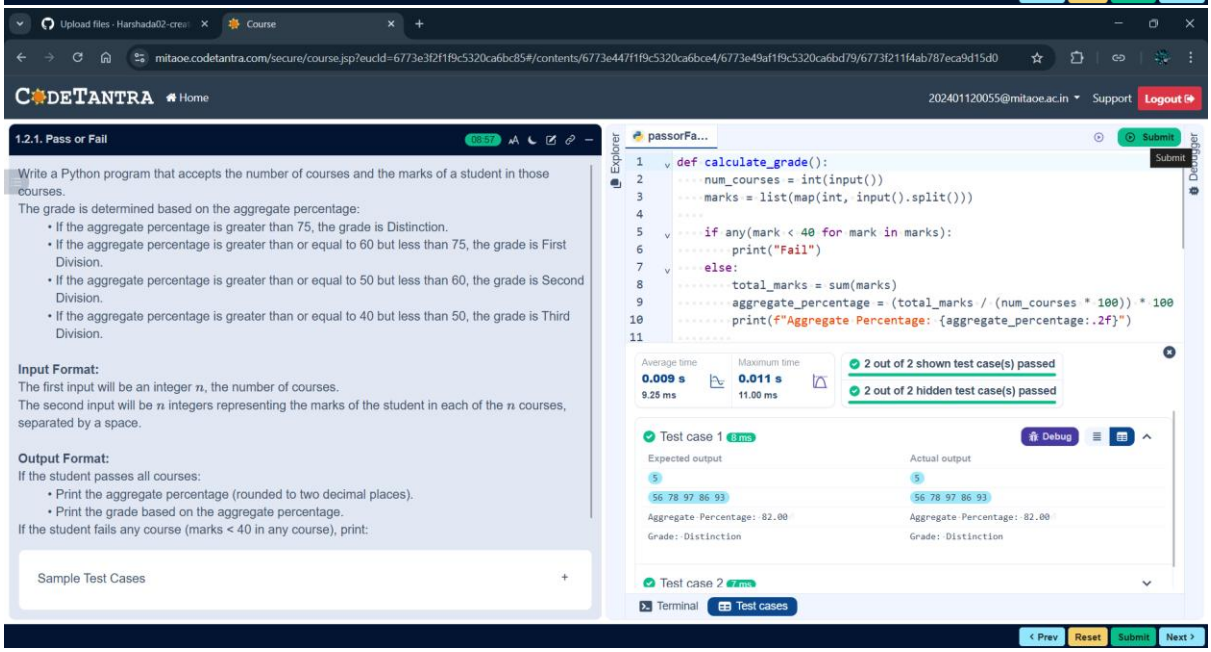
PRACTICAL 1

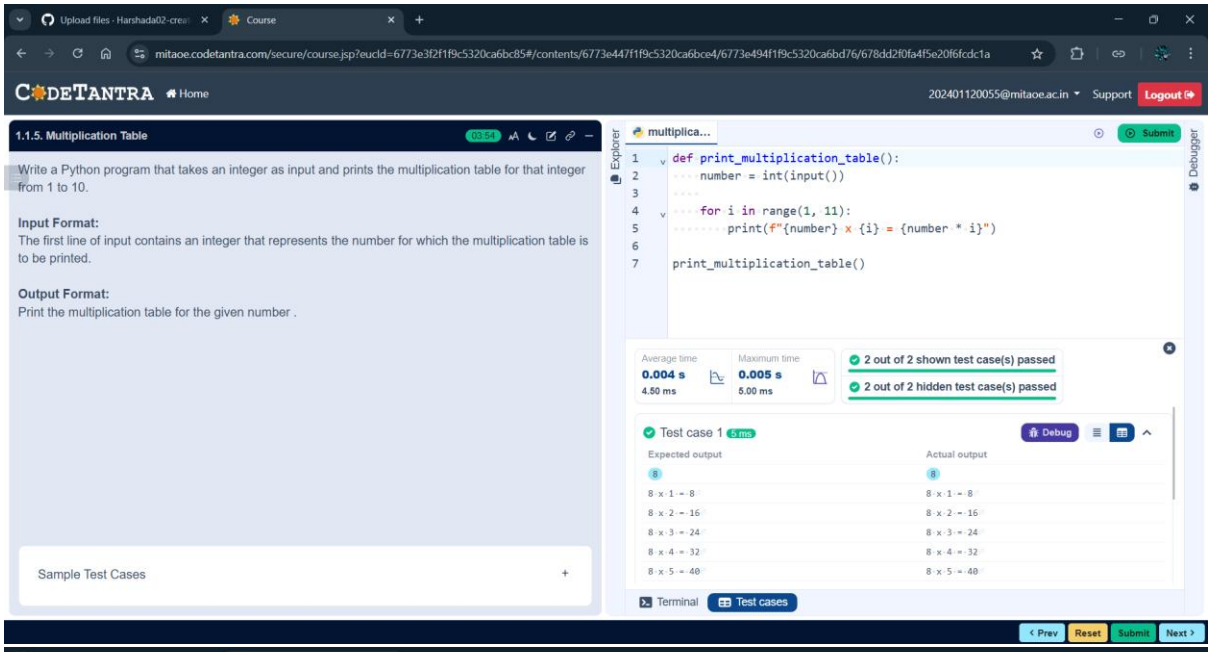
ALL PRACTICALS:

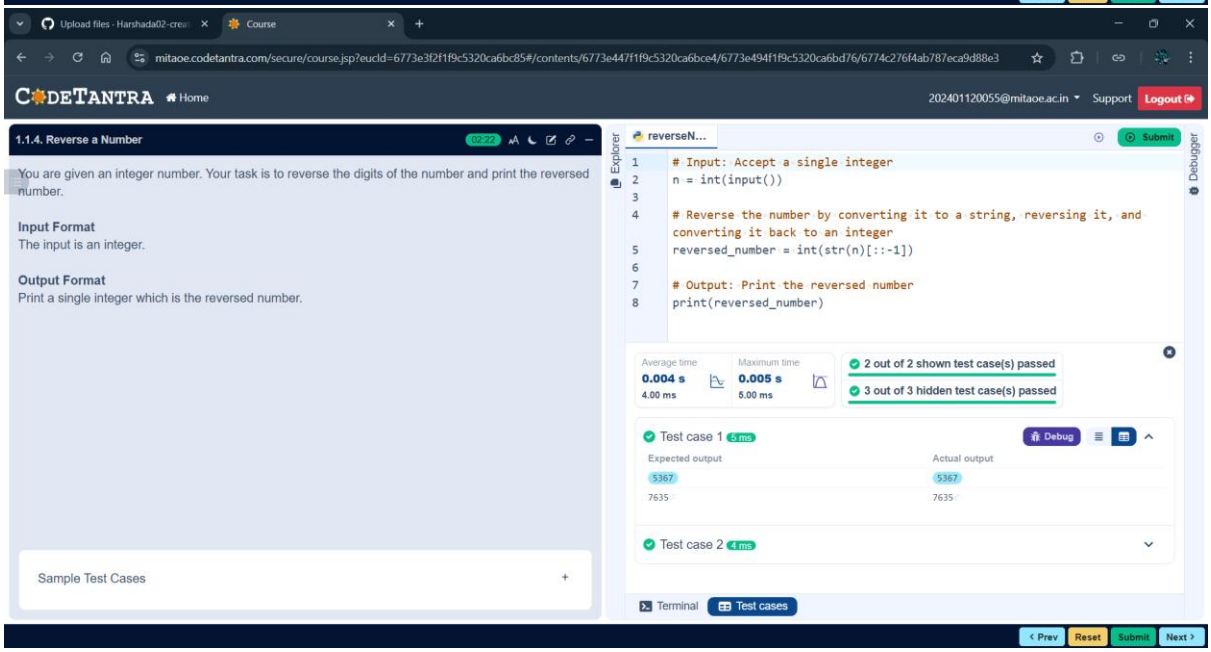


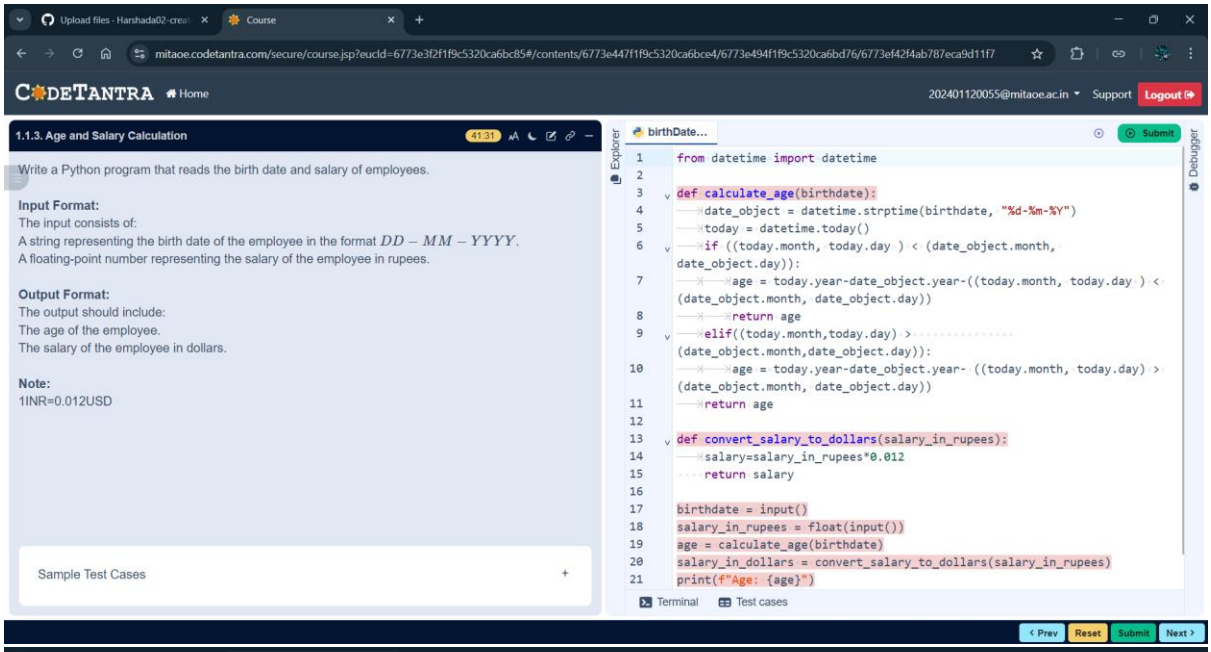


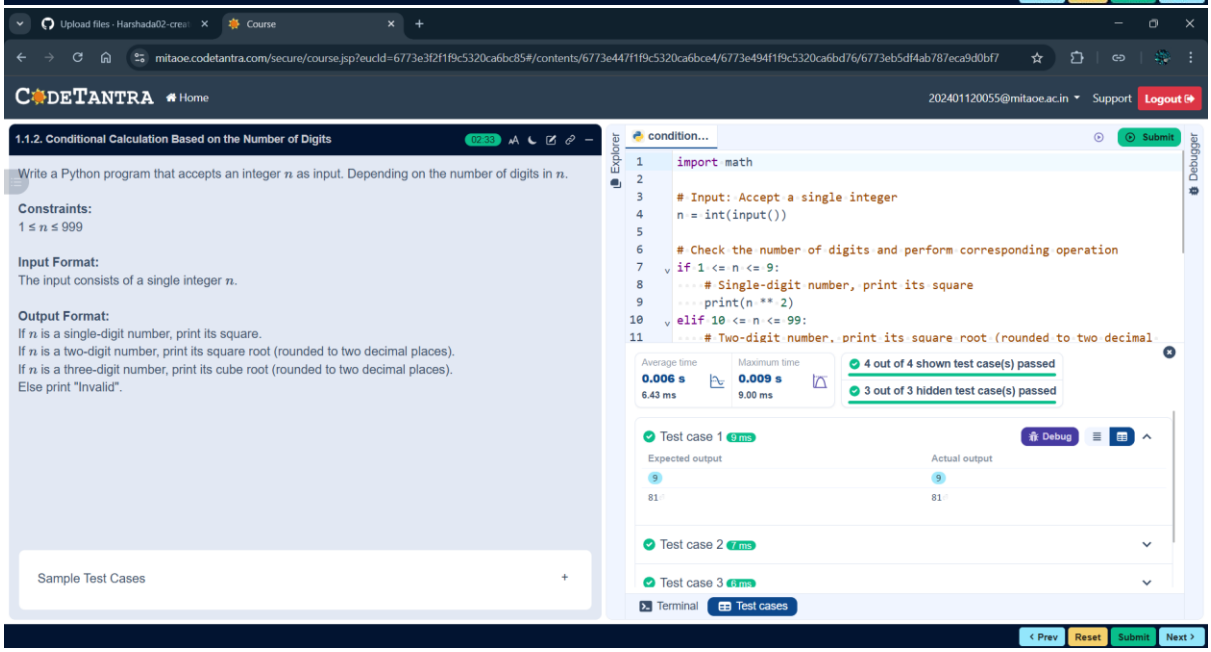

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/65f7d776bfe2dc3ed510ca60
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.2.2. Fibonacci series using Recursive Function
18:47
Write a Python program to find the Fibonacci series of a given number of terms using recursive function calls.
Expected Output-1:
Enter terms for Fibonacci series: 5
0 1 1 2 3
Expected Output-2:
Enter terms for Fibonacci series: 9
0 1 1 2 3 5 8 13 21
Instructions:
• 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.
Sample Test Cases
fib.py
Submit
Explorer
Debugger
def fib(n, a=0, b=1):
if n == 0:
return a
else:
return fib(n - 1, b, a + b)
n=int(input("Enter terms for Fibonacci series: "))
for i in range (n):
print(fib(i),end=" ")
Average time
0.004 s
4.50 ms
Maximum time
0.005 s
5.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1 5 ms
Debug
Expected output
Enter terms for Fibonacci series: 5
0 1 1 2 3
Actual output
Enter terms for Fibonacci series: 5
0 1 1 2 3
Test case 2 4 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›

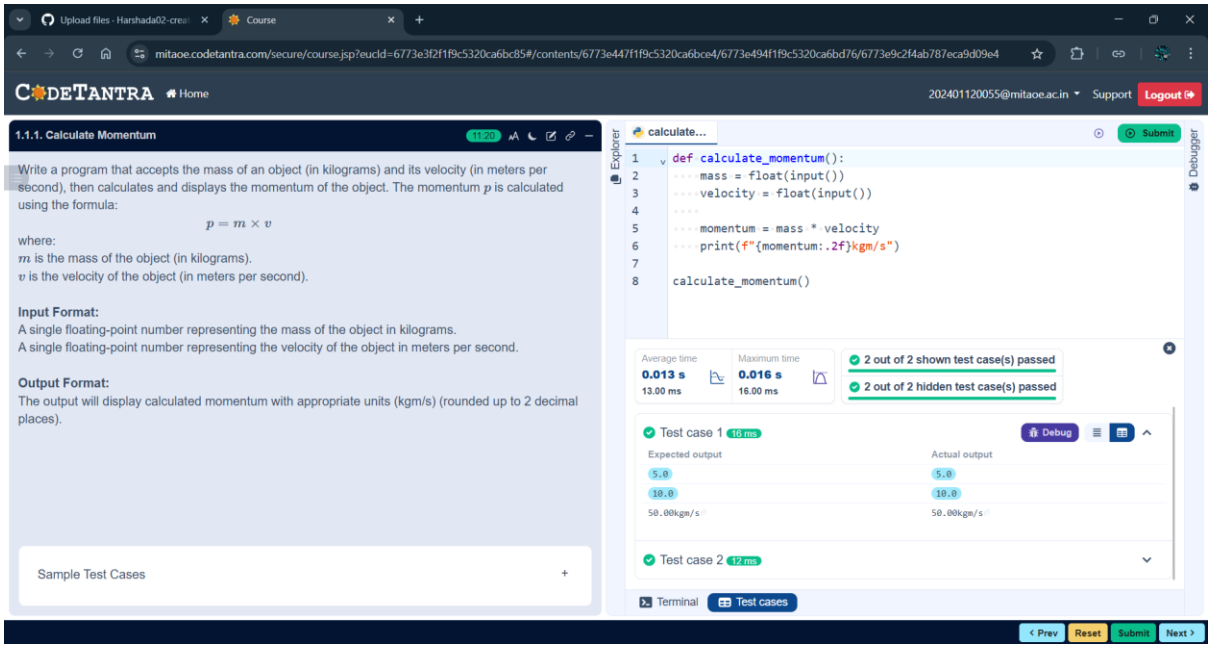

Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6773f211f4ab787eca9d15d0
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.2.1. Pass or Fail
08:57
Write a Python program that 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 Division.
Input Format:
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:
• 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:
Sample Test Cases
passorFa...
Submit
Explorer
Debugger
def calculate_grade():
num_courses = int(input())
marks = list(map(int, input().split()))
if any(mark < 40 for mark in marks):
print("Fail")
else:
total_marks = sum(marks)
aggregate_percentage = (total_marks / (num_courses * 100)) * 100
print(f"Aggregate Percentage: {aggregate_percentage:.2f}")
Average time
0.009 s
9.25 ms
Maximum time
0.011 s
11.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1 8 ms
Debug
Expected output
5
56 78 97 86 93
Aggregate Percentage: 82.00
Grade: Distinction
Actual output
5
56 78 97 86 93
Aggregate Percentage: 82.00
Grade: Distinction
Test case 2 7 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0fa4f5e20f6fcdc1a
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.1.5. Multiplication Table
03:54
Write a Python program that takes an integer as input and prints the multiplication table for that integer from 1 to 10.
Input Format:
The first line of input contains an integer that represents the number for which the multiplication table is to be printed.
Output Format:
Print the multiplication table for the given number .
Sample Test Cases
Explorer
multiplica...
Submit
Debugger
def print_multiplication_table():
number = int(input())
for i in range(1, 11):
print(f"{number} x {i} = {number * i}")
print_multiplication_table()
Average time
0.004 s
4.50 ms
Maximum time
0.005 s
5.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
4 ms
Debug
Expected output
Actual output
8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6774c276f4ab787eca9d88e3
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.1.4. Reverse a Number
02:22
You are given an integer number. Your task is to reverse the digits of the number and print the reversed number.
Input Format
The input is an integer.
Output Format
Print a single integer which is the reversed number.
Sample Test Cases
Explorer
reverseN...
Submit
Debugger
# Input: Accept a single integer
n = int(input())
# Reverse the number by converting it to a string, reversing it, and converting it back to an integer
reversed_number = int(str(n)[::-1])
# Output: Print the reversed number
print(reversed_number)
Average time
0.004 s
4.00 ms
Maximum time
0.005 s
5.00 ms
2 out of 2 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1
4 ms
Debug
Expected output
Actual output
5367
7635
Test case 2
4 ms
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773ef42f4ab787eca9d11f7
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.1.3. Age and Salary Calculation
41:31
Write a Python program that reads the birth date and salary of employees.
Input Format:
The input consists of:
A string representing the birth date of the employee in the format $DD-MM-YYYY$.
A floating-point number representing the salary of the employee in rupees.
Output Format:
The output should include:
The age of the employee.
The salary of the employee in dollars.
Note:
1INR=0.012USD
Sample Test Cases
Explorer
birthDate...
Submit
Debugger
from datetime import datetime
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)):
age = today.year-date_object.year- ((today.month, today.day) > (date_object.month, date_object.day))
return age
def convert_salary_to_dollars(salary_in_rupees):
salary=salary_in_rupees*0.012
return salary
birthdate = input()
salary_in_rupees = float(input())
age = calculate_age(birthdate)
salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
print(f"Age: {age}")
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773eb5df4ab787eca9d0bf7
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.1.2. Conditional Calculation Based on the Number of Digits
02:33
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".
Sample Test Cases
Explorer
condition...
Submit
Debugger
import math
# Input: Accept a single integer
n = int(input())
# Check the number of digits and perform corresponding operation
if 1 <= n <= 9:
# Single-digit number, print its square
print(n ** 2)
elif 10 <= n <= 99:
# Two-digit number, print its square root (rounded to two decimal
Average time
0.006 s
6.43 ms
Maximum time
0.009 s
9.00 ms
4 out of 4 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1
9 ms
Debug
Expected output
Actual output
9
81
9
81
Test case 2
7 ms
Test case 3
8 ms
Terminal
Test cases
Prev
Reset
Submit
Next


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773e9c2f4ab787eca9d09e4
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
1.1.1. Calculate Momentum
11:20
Write a program that accepts the mass of an object (in kilograms) and its velocity (in meters per second), then calculates and displays the momentum of the object. The momentum p is calculated using the formula:
$$p = m \times v$$
where:
m is the mass of the object (in kilograms).
v is the velocity of the object (in meters per second).
Input Format:
A single floating-point number representing the mass of the object in kilograms.
A single floating-point number representing the velocity of the object in meters per second.
Output Format:
The output will display calculated momentum with appropriate units (kgm/s) (rounded up to 2 decimal places).
Sample Test Cases
Explorer
calculate...
Submit
Debugger
1 def calculate_momentum():
2 mass = float(input())
3 velocity = float(input())
4
5 momentum = mass * velocity
6 print(f"{momentum:.2f}kgm/s")
7
8 calculate_momentum()
Average time
0.013 s
13.00 ms
Maximum time
0.016 s
16.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
16 ms
Debug
Expected output
Actual output
5.0
10.0
50.00kgm/s
5.0
10.0
50.00kgm/s
Test case 2
12 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›