

1.1.1. Calculate Momentum

12:55



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



calculate...

Submit

```
1
2 m=float(input())
3 v=float(input())
4 p=m*v
5 print('%0.2f'%p,end="")
6 print("kgm/s")
```

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.1.3. Age and Salary Calculation

30:51



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

```
1 from datetime import datetime
2
3 def calculate_age(birthdate):
4     date_object = datetime.strptime(birthdate, "%d-%m-%Y")
5     today = datetime.today()
6     if ((today.month, today.day) < (date_object.month,
7         date_object.day)):
8         age = today.year-date_object.year-((today.month,
9             today.day) < (date_object.month, date_object.day))
10        elif((today.month, today.day) >
11            (date_object.month,date_object.day)):
12            age = today.year-date_object.year-((today.month,
13                today.day) > (date_object.month, date_object.day))
14        return age
15
16 def convert_salary_to_dollars(salary_in_rupees):
17     salary=salary_in_rupees*0.012
18     return salary
19
20 birthdate = input()
```

1.1.2. Conditional Calculation Based on the Number of Digits

19:03



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

Constraints: $1 \leq n \leq 999$ **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



condition...

Submit

```
1 n=int(input())
2 v if(0<n<10):
3   —>print(n*n)
4 v elif(9<n<99):
5   —>print( "%0.2f"%n**0.5)
6 v elif(99<n<1000):
7   —>print("%0.2f"%n**(1/3))
8 v else:
9   —>print("Invalid")
```

Terminal

Test cases

< Prev

Reset

Submit

Next >

1.1.4. Reverse a Number

15/11



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

```
1 n=int(input())
2 i = 0
3 while n != 0:
4     digit = n % 10
5     i = i*10 + digit
6     n = n//10
7 print(i)
```



Terminal



Test cases

Activate Windows

Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.1.5. Multiplication Table

12:09



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



multiplica...

Submit

```
1 i=int(input())
2 n=1
3 while n<=10:
4     print(i,"x",n,"=",i*n)
5     n=n+1
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.2.1. Pass or Fail

23:11



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:

Sample Test Cases



passorFa...

Submit

```
1 n:=int(input(""))
2 marks= list(map(int, input().split()))
3 v if any(mark<40 for mark in marks):
4     print("Fail")
5 v else:
6     totalmarks = sum(marks)
7     Aggregate_percentage=(totalmarks/(n*100))*100
8     print(f"Aggregate Percentage: {Aggregate_percentage:.2f}")
9 v if(Aggregate_percentage>75):
10     print("Grade: Distinction")
11 v elif(75>Aggregate_percentage>=60 ):
12     print("Grade: First Division")
13 v elif(60>Aggregate_percentage>=50):
14     print("Grade: Second Division")
15 v elif(50>Aggregate_percentage>=40):
16     print("Grade: Third Division")
17 v else:
18     print("Fail")
19
```

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.2.2. Fibonacci series using Recursive Function

02:37



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



Explorer

fib.py

Submit

Debugger

```
1  def fib(n):
2      if n <= 0:
3          return 0
4      elif n == 1:
5          return 1
6      else:
7          return fib(n-1) + fib(n-2)
8
9  n=int(input("Enter terms for Fibonacci series: "))
10 for i in range (n):
11     print(fib(i),end=" ")
```

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.2.3. Pattern - 1

24:19



Write a Python program to print a pattern of asterisks in the form of a right-angled triangle.

Input Format:

The input is an integer, representing the number of rows in the pattern.

Output Format

The output should display the pattern of asterisks (*), with each row containing an increasing number of asterisks.

Note:

Refer to the displayed test cases for the sample pattern.

Sample Test Cases



rightangl...

Submit

```
1 n=int(input())
2 for i in range(1,n+1):
3     for j in range(0,i):
4         print("*",end=" ")
5     print()
6
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >

1.2.4. Pattern - 2

10:22

Write a Python program to print a right-angled triangle pattern of numbers.

Input Format:

The input is an integer, representing the number of rows in the pattern.

Output Format:

The output should display the pattern of numbers, with each row containing increasing numbers starting from 1 up to the row number.

Note:

Refer to the displayed test cases for the sample pattern.

Sample Test Cases

+

numberP...

Submit

```
1 n=int(input())
2 for i in range(1,n+1):
3     for j in range(1,i+1):
4         print(j,end=" ")
5     print()
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.

< Prev

Reset

Submit

Next >