

1.2.4. Pattern - 2

00:39



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.

1.2.3. Pattern - 1

01:44



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

Explorer

```
1 num=int(input())
2 for i in range(1,num+1):
3     print("*"*i)
```

Debugger

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

1.2.2. Fibonacci series using Recursive Function

00:36



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

```
1  def fib(n):
2      if n<=1:
3          return n
4      else:
5          return fib(n-1)+fib(n-2)
6
7
8
9
10
11
12
13
14
15
16  n=int(input("Enter terms for Fibonacci series: "))
17  for i in range (n):
18      print(fib(i),end=" ")
```

Terminal

Test cases

Activate Windows

Go to Settings to activate Windows.

1.2.1. Pass or Fail

00:24



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



Explorer

passorFa...



Submit

Debugger

```
1 def calculate_grade(num_courses, marks):
2     if any(mark < 40 for mark in marks):
3         print("Fail")
4         return
5
6     aggregate_percentage = sum(marks) / num_courses
7
8     print(f"Aggregate Percentage: {aggregate_percentage:.2f}")
9
10    if aggregate_percentage > 75:
11        print("Grade: Distinction")
12    elif aggregate_percentage >= 60:
13        print("Grade: First Division")
14    elif aggregate_percentage >= 50:
15        print("Grade: Second Division")
16    elif aggregate_percentage >= 40:
17        print("Grade: Third Division")
18
19    num_courses = int(input())
20    marks = list(map(int, input().split()))
21
22
23    calculate_grade(num_courses, marks)
```

Terminal

Test cases

Activate Windows
Go to Settings to activate Windows.