

1.2.1. Pass or Fail

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

Sample Test Cases

passorFa...

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

Terminal Test cases

1.2.2. Fibonacci Series

11:56

Write a Python program that uses recursion to print the first n terms of the Fibonacci series.

Input Format:

- A single integer n representing the number of terms to generate.

Output Format:

- A single line containing the first n terms of the Fibonacci sequence, separated by spaces.

Sample Test Cases

fibonacci...

Submit

```
1 def fibonacci(n):
2     if(n==1):
3         return 0
4     elif(n==2):
5         return 1
6     else:
7         return fibonacci(n-1)+fibonacci(n-2)
8
9
10 n = int(input())
11 for i in range(1, n + 1):
12     print(fibonacci(i), end=" ")
13
```

Terminal Test cases

1.2.3. Pattern - 1

10:44

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

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

Terminal Test cases

Essentials of Data Science Laboratory - 2304102L - 2026

Search course ctrl + k

- 1. Practical 1
 - 1.1. Practice Lab Assignment
 - 1.2. Lab Assignment
 - 1.2.1. Pass or Fail
 - 1.2.2. Fibonacci Series
 - 1.2.3. Pattern - 1
 - 1.2.4. Pattern - 2**
- 2. Practical 2
 - 2.1. Practice Lab Assignment
 - 2.2. Lab Assignment
 - 2.2.1. Linear search Technique
 - 2.2.2. Captain of the Team
- 3. Practical 3

1.2.4. Pattern - 2

07:11

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 separated by space, 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...

```
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()
6
```

Terminal Test cases

< Prev Reset Submit Next >

2.2.1. Linear search Technique

07:25

Write a program to check whether the given element is present or not in the array of elements using linear search.

Input format:

- The first line of input contains the array of integers which are separated by space
- The last line of input contains the key element to be searched

Output format:

- If the element is found, print the index.
- If the element is not found, print **Not found**.

Sample Test Case:

Input:
1 2 3 4 3 5 6
3

Output:
2

Sample Test Cases +

CTP1709...

Submit

```
1 arr=list(map(int,input().split()))
2 key=int(input())
3 found=False
4 for i in range(len(arr)):
5     if arr[i] == key:
6         print(i)
7         found=True
8         break
9 if not found:
10    print("Not found")
```

Terminal Test cases

2.2.2. Captain of the Team

07:18

You are provided with the heights of 11 cricket players (in centimeters). Your task is to identify the tallest player, who will be selected as the captain of the team.

Input Format:
The first line of input will contain 11 integers, each representing the height of a player (in centimeters), each separated by a space.

Output Format
The output should be the height (in centimeters) of the tallest player.

Sample Test Cases +

captainof...

Submit

```
1 height=list(map(int,input().split()))
2 tallest=max(height)
3 print(tallest)
```

Terminal Test cases