Started on	Tuesday, 8 July 2025, 1:19 PM
State	Finished
Completed on	Tuesday, 8 July 2025, 1:30 PM
Time taken	11 mins 31 secs
Marks	4.00/5.00
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

Find the simple interest by getting the principal, rate and time value from the user

simple interest = (principal*rate*time)/100

Note: Time must be in year so convert 9 months to year format

For example:

Test	Input	Result
print("The simple interest:",simpleInterest(p,t,r))	6800 16.66 9/12	The simple interest: 849.66

Answer: (penalty regime: 0 %)

```
def simpleInterest(p,t,r):
    simple=(p*t*r)/100
    return simple
p=eval(input())
t=eval(input())
r=eval(input())
```

	Test	Input	Expected	Got	
✔	print("The simple interest:",simpleInterest(p,t,r))	6800 16.66 9/12	The simple interest: 849.66	The simple interest: 849.66	✔
✔	print("The simple interest:",simpleInterest(p,t,r))	3000 6.25 1	The simple interest: 187.5	The simple interest: 187.5	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Write a program that has a dictionary of names of students and their marks in five subjects. Create another dictionary from this dictionary that has the name of the students and their total marks. Find out the topper and the score.

For example:

Input	Result
{'Alice':[87,94,92,88,94], 'Bob':[87,67,78,75,83], 'Eve':[91,93,85,86,81]}	{'Alice': 455, 'Bob': 390, 'Eve': 436} Topper is: Alice with marks = 455

Answer: (penalty regime: 0 %)

```
marks = eval(input())
total = 0
total_marks = marks.copy()
for key,val in marks.items():
    total = sum(val)
    total_marks[key] = total
print(total_marks)
max = 0
topper = ''
for key,val in total_marks.items():
    if val>max:
        max = val
        topper = key
print("Topper is: ", topper, "with marks = ",max)

```

	Input	Expected	Got	
✔	{'Alice':[87,94,92,88,94], 'Bob':[87,67,78,75,83], 'Eve':[91,93,85,86,81]}	{'Alice': 455, 'Bob': 390, 'Eve': 436} Topper is: Alice with marks = 455	{'Alice': 455, 'Bob': 390, 'Eve': 436} Topper is: Alice with marks = 455	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Not answered

Mark 0.00 out of 1.00

Write python code to create a class Triangle and initialiaze the attributes(base and height) using default constructor and calculate the area of the triangle using user defined function.

For example:

Input	Result
5 14	Area of triangle: 35.0

Answer: (penalty regime: 0 %)

```
1
```

	Input	Expected	Got	
✖	5 14	Area of triangle: 35.0	Area of triangle: 65	✖

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

The included code stub will read an integer, n, from STDIN.

Without using any string methods, try to print the following:

$123\cdots n$

Note that "$\cdots$" represents the consecutive values in between.

Example

$n = 5$

Print the string 12345.

Input Format

The first line contains an integer n.

Constraints

$1 \leq n \leq 150$

Output Format

Print the list of integers from 1 through n as a string, without spaces.

For example:

Input	Result
3	123

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
✔	3	123	123	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

You are given a string S.
Your task is to find out whether S is a valid regex or not.

Input Format

The first line contains integer T, the number of test cases.
The next T lines contains the string S.

Constraints

$0 < T < 100$

Output Format

Print "True" or "False" for each test case without quotes.

For example:

Input	Result
2 .*\+ .*+	True False

Answer: (penalty regime: 0 %)

```
import re
n=int(input())
for i in range(n):
    s=input()
    try:
        p=re.compile(s)
        if p:
            print('True')
    except re.error:
        print('False')
```

	Input	Expected	Got	
✔	2 .*\+ .*+	True False	True False	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.