Started on	Monday, 17 February 2025, 9:07 AM
State	Finished
Completed on	Monday, 17 February 2025, 9:45 AM
Time taken	37 mins 51 secs
Marks	4.00/5.00
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

Write a Python program to find sequences of Upper case letters joined with a underscore.

For example:

Input	Result
COMPU_TER	Found a match!

Answer: (penalty regime: 0 %)

```
n=input()
if n[0].isupper():
    print("Found a match!")
else:
    print("Not matched!")
```

	Input	Expected	Got	
✔	COMPU_TER	Found a match!	Found a match!	✔
✔	saveetha engineering	Not matched!	Not matched!	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

The provided code stub reads two strings from STDIN, a and b. Add code to print three lines where:

1. The first line contains the concatenation of the two strings.
2. The second line contains the repetition of the first string 3 times

Note: Get the values in float

For example:

Input	Result
Good Morning	GoodMorning GoodGoodGood

Answer: (penalty regime: 0 %)

```
a=input()
b=input()
print(a+b)
print(a*3)
```

	Input	Expected	Got	
✔	Good Morning	GoodMorning GoodGoodGood	GoodMorning GoodGoodGood	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Incorrect

Mark 0.00 out of 1.00

Let's dive into the interesting topic of regular expressions! You are given some input, and you are required to check whether they are valid mobile numbers.

A valid mobile number is a ten digit number starting with a $7, 8$ or 9.

Concept

A valid mobile number is a ten digit number starting with a $7, 8$ or 9.

Regular expressions are a key concept in any programming language. A quick explanation with Python examples is available here. You could also go through the link below to read more about regular expressions in Python.

Input Format

The first line contains an integer N, the number of inputs.
N lines follow, each containing some string.

Constraints

$1 \leq N \leq 10$

$2 \leq len(Number) \leq 15$

Output Format

For every string listed, print "*YES*" if it is a valid mobile number and "*NO*" if it is not on separate lines. Do not print the quotes.

For example:

Input	Result
2 9587456281 1252478965	YES NO

Answer: (penalty regime: 0 %)

```
import re
for i in range(int(input())):
    N=input().strip()
    if N.isalnum() and len(N)==10:
        if bool(re.search(r'(.*[7-9]){2,}',N)) and bool(re.search(r'(.*[7-9]){3,}',N)):
            if re.search(r'.*(.).*\1+.*',N):
                print("NO")
            else:
                print("YES")
        else:
            print("NO")
    else:
        print("NO")

```

	Input	Expected	Got	
✖	2 9587456281 1252478965	YES NO	NO NO	✖

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 provided code stub reads and integer, n, from STDIN. For all non-negative integers $i < n$, print i^3

Example

$n = 3$

The list of non-negative integers that are less than $n = 3$ is $[0, 1, 2]$. Print the square of each number on a separate line.

```
0
1
4
```

Input Format

The first and only line contains the integer, n.

Constraints

$1 \leq n \leq 20$

Output Format

Print n lines, one corresponding to each i.

For example:

Input	Result
3	0 1 8

Answer: (penalty regime: 0 %)

```
n=int(input())
for i in range(n):
    print(i**3)
```

	Input	Expected	Got	
✔	3	0 1 8	0 1 8	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

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

Without using any build-in methods, try to print the numbers in reverse order

Example

n= 1234

Print the string 4321

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
321	123

Answer: (penalty regime: 0 %)

```
n=int(input())
while(n!=0):
    print(n%10,end="")
    n=n//10
```

	Input	Expected	Got	
✔	321	123	123	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.