Started on	Wednesday, 12 March 2025, 9:37 AM
State	Finished
Completed on	Wednesday, 12 March 2025, 10:01 AM
Time taken	23 mins 50 secs
Marks	4.00/5.00
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

CSS colors are defined using a hexadecimal (*HEX*) notation for the combination of Red, Green, and Blue color values (*RGB*).

Specifications of HEX Color Code

- It must start with a '#' symbol.
- It can have 3 or 6 digits.
- Each digit is in the range of 0 to F. $(1,2,3,4,5,6,7,8,9,0,A,B,C,D,E$ and $F)$.
- $A-F$ letters can be lower case. $(a,b,c,d,e$ and f are also valid digits).

Examples

```
Valid Hex Color Codes
#FFF
#025
#F0A1FB

Invalid Hex Color Codes
#fffabg
#abcf
#12365erff
```

You are given N lines of *CSS* code. Your task is to print all valid *Hex Color Codes*, in order of their occurrence from top to bottom.

Input Format

The first line contains N, the number of code lines.
The next N lines contains *CSS* Codes.

Constraints

$0 < N < 50$

Output Format

Output the color codes with '#' symbols on separate lines.

Explanation

#BED and *#Cab* satisfy the Hex Color Code criteria, but they are used as selectors and not as color codes in the given CSS.

Hence, the valid color codes are:

#FfFdF8
#aef
#f9f9f9
#fff
#ABC
#fff

Note: There are no comments (*// or /* */*) in CSS Code.

For example:

Input	Result

11
#BED
{
    color: #FfFdF8; background-color:#aef;
    font-size: 123px;
    background: -webkit-linear-gradient(top, #f9f9f9, #fff);
}
#Cab
{
    background-color: #ABC;
    border: 2px dashed #fff;
}
	#FfFdF8
	#aef
	#f9f9f9
	#fff
	#ABC
	#fff

Answer: (penalty regime: 0 %)

```
import re

T = int(input())
in_css = False
for _ in range(T):
    s = input()
    if '{' in s:
        in_css = True
    elif '}' in s:
        in_css = False
    elif in_css:
        for color in re.findall('#[0-9a-fA-F]{3,6}', s):
            print(color)
```

	Input	Expected	Got	

✓	11
	#BED
	{
	    color: #FfFdF8; background-color:#aef;
	    font-size: 123px;
	    background: -webkit-linear-gradient(top, #f9f9f9, #fff);
	}
	#Cab
	{
	    background-color: #ABC;
	    border: 2px dashed #fff;
	}
		#FfFdF8
		#aef
		#f9f9f9
		#fff
		#ABC
		#fff
			#FfFdF8
			#aef
			#f9f9f9
			#fff
			#ABC
			#fff
				✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Given an integer, n, perform the following conditional actions:

- If n is odd, print Weird
- If n is even and in the inclusive range of 2 to 5, print Not Weird
- If n is even and in the inclusive range of 6 to 20, print Weird
- If n is even and greater than 20, print Not Weird

Input Format

A single line containing a positive integer, n.

Constraints

- $1 \leq n \leq 100$

Output Format

Print Weird if the number is weird. Otherwise, print Not Weird.

For example:

Input	Result
3	Weird

Answer: (penalty regime: 0 %)

```
n = int(input().strip())
if(n%2==0):
    if(n>=2 and n<=5):
        print("Not Weird")
    elif(n>=6 and n<=20):
        print("Weird")
    elif(n>=20):
        print("Not Weird")
else:
    print("Weird")
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Incorrect

Mark 0.00 out of 1.00

construct a Python Program to find Factorial of the number '3'

Answer: (penalty regime: 0 %)

```
num=int(input())

from math:
    import fact:
        return()

print()
```

Syntax Error(s)

```
  File "__tester__.python3", line 3
    from math:
             ^
SyntaxError: invalid syntax
```

Incorrect

Marks for this submission: 0.00/1.00.

Question **4**

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 **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 %)

```
a=input()
b=a[::-1]
print(b)
```

	Input	Expected	Got	
✓	321	123	123	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.