

Q.1.Say "Hello, World!"

With Python

Here is a sample line of code that can be executed in Python:

```
print("Hello, World!")
```

You can just as easily store a string as a variable and then print it to stdout:

```
my_string = "Hello, World!"
```

```
print(my_string)
```

The above code will print Hello, World! on your screen. Try it yourself in the editor below!

Input Format

You do not need to read any input in this challenge.

Output Format

Print Hello, World! to stdout.

Sample Output

Hello, World!

```
In [1]: print('Hello,World!')
```

Hello,World!

```
In [2]: str='Hello,World! '
print(str)
```

Hello,World!

Q.2.Python If-Else

Task

Given an integer, , perform the following conditional actions:

If is odd, print Weird

If is even and in the inclusive range of to , print Not Weird

If is even and in the inclusive range of to , print Weird

If is even and greater than , print Not Weird

Input Format

A single line containing a positive integer, .

Constraints

$1 \leq n \leq 100$

Output Format

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

Sample Input 0

3

Sample Output 0

Weird

Sample Input 1

Sample Output 1

Not Weird

```
In [4]: n=int(input())
        if n %2 != 0:
            print('Weird')
        elif n%2 == 0 and n in range(2,6):
            print('Not Weird')
        elif n%2 == 0 and n in range(6,21):
            print('Weird')
        else:
            print('Not Weird')
```

3

Weird

```
In [5]: n=int(input())
        if n %2 != 0:
            print('Weird')
        elif n%2 == 0 and n in range(2,6):
            print('Not Weird')
        elif n%2 == 0 and n in range(6,21):
            print('Weird')
        else:
            print('Not Weird')
```

24

Not Weird

Q.3.Arithmetic Operators

Task

Read two integers from STDIN and print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).

3. The third line contains the product of the two numbers.

Input Format

The first line contains the first integer, . The second line contains the second integer, .

Output Format

Print the three lines as explained above.

Sample Input 0

```
3
2
```

Sample Output 0

```
5
1
6
```

```
In [6]: print('input')
a = int(input())
b = int(input())
print('output')
print(a+b)
print(a-b)
print(a*b)
```

```
input
3
2
output
5
1
6
```

```
In [ ]:
```

4.Python: Division

Task

Read two integers and print two lines. The first line should contain integer division, `//` . The second line should contain float division, `/` . You don't need to perform any rounding or formatting operations.

Input Format

The first line contains the first integer, `a` . The second line contains the second integer, `b` .

Output Format

Print the two lines as described above.

Sample Input 0

```
4
3
```

Sample Output 0

```
1
1.333333333333
```

```
In [7]: print('Input')
a = int(input())
b = int(input())
print('Output')
print(a//b)
print(a/b)
```

```
Input
4
3
Output
1
1.3333333333333333
```

```
In [ ]:
```

5.Loops

Task

Read an integer n . For all non-negative integers i , print i^2 . See the sample for details.

Input Format

The first and only line contains the integer, n .

Output Format

Print lines, one corresponding to each i .

Sample Input 0

5

Sample Output 0

0
1
4
9
16

```
In [8]: print('Input')
n = int(input())
print('Output')
for i in range (0,n):
    print(i**2)
```

Input

5

Output

0

1

4

9
16

In []:

6. Write a function

We add a Leap Day on February 29, almost every four years. The leap day is an extra, or intercalary day and we add it to the shortest month of the year, February.

In the Gregorian calendar three criteria must be taken into account to identify leap years:

```
The year can be evenly divided by 4, is a leap year, unless:  
The year can be evenly divided by 100, it is NOT a leap year, unless:  
The year is also evenly divisible by 400. Then it is a leap year.
```

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. Source

Task

You are given the year, and you have to write a function to check if the year is leap or not. Note that you have to complete the function and remaining code is given as template.

Input Format

Read y, the year that needs to be checked.

Output Format

Output is taken care of by the template. Your function must return a boolean value (True/False)

Sample Input 0

1990

Sample Output 0

False

Explanation 0

1990 is not a multiple of 4 hence it's not a leap year

```
In [15]: def is_leap(year):  
    leap = False  
    if year % 4 == 0:  
        return True  
    elif year % 400 == 0:  
        return True  
    else:  
        leap  
  
    return leap  
  
year = int(input())  
print(is_leap(year))
```

2004
True

```
In [16]: def is_leap(year):  
    leap = False  
    if year % 4 == 0:  
        return True  
    elif year % 400 == 0:  
        return True  
    else:  
        leap  
  
    return leap  
  
year = int(input())  
print(is_leap(year))
```

1998
False

In []:

7.Print Function

The included code stub will read an integer, , from STDIN.Without using any string methods, try to print the following:

Note that "" represents the consecutive values in between.

Example

Print the string .

Input Format

The first line contains an integer .

Output Format

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

Sample Input 0

3

Sample Output 0

123

```
In [2]: print('input')
n = int(input())
print('Output')
for i in range (0,n):
    print(i+1,end="")
```

input

5

Output

12345

In []:

