**Q.1 :** Given an integer, , 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 <= n <= 100

**Output Format**

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

**Answer :** if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input().strip())

    if n % 2 == 1:

        print('Weird')

    elif n % 2 == 0 and 2 <= n <= 5:

        print('Not Weird')

    elif n % 2 == 0 and 6 <= n <= 20:

        print('Weird')

    elif n % 2 == 0 and n > 20:

        print('Not Weird')

**Output** : Input (stdin)

* **3**

Your Output (stdout)

* **Weird**

Expected Output

* **Weird**

**Q.2 :** The provided code stub reads two integers from STDIN, a and b. Add code to 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.

**Example**  
a = 3  
b = 5

Print the following:

8

-2

15

**Input Format**

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

**Constraints**

1 <= a <= 10^10

1 <= b <= 10^10

**Output Format**

Print the three lines as explained above.

**Answer :** if \_\_name\_\_ == '\_\_main\_\_':

    a = int(input())

    b = int(input())

print(a + b)

print(a - b)

print(a \* b)

**Output : Input (stdin)**

* **3**
* **2**

**Your Output (stdout)**

* **5**
* **1**
* **6**

**Expected Output**

* **5**
* **1**
* **6**

**Q.3 :** The provided code stub reads two integers, a and b, from STDIN.

Add logic to print two lines. The first line should contain the result of integer division, a // b. The second line should contain the result of float division, a / b.

No rounding or formatting is necessary.

**Example**  
a = 3  
b = 5

* The result of the integer division 3 // 5 = 0.
* The result of the float division is 3 / 5 = 0.6.

Print:

0

0.6

**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.

**Answer :** if \_\_name\_\_ == '\_\_main\_\_':

    a = int(input())

    b = int(input())

print(a // b)

print(a / b)

**Output :** Input (stdin)

* **4**
* **3**

Your Output (stdout)

* **1**
* **1.3333333333333333**

Expected Output

* **1**
* **1.33333333333**

**Q.4 :** The provided code stub reads an integer, n, from STDIN. For all non-negative integers i < n, print i^2.

**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 <= n <= 20

**Output Format**

Print n lines, one corresponding to each i.

**Answer :** if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input())

    for i in range(n):

        if i < n:

            print(i\*\*2)

**Output :** Input (stdin)

* **5**

Your Output (stdout)

* **0**
* **1**
* **4**
* **9**
* **16**

Expected Output

* **0**
* **1**
* **4**
* **9**
* **16**

**Q.5 :** An extra day is added to the calendar almost every four years as February 29, and the day is called a *leap day*. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used 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](http://www.timeanddate.com/date/leapyear.html)

**Task**

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True, otherwise return False.

Note that the code stub provided reads from STDIN and passes arguments to the is\_leap function. It is only necessary to complete the is\_leap function.

**Input Format**

Read year, the year to test.

**Constraints**

1900 <= year <= 10^5

**Output Format**

The function must return a Boolean value (True/False). Output is handled by the provided code stub.

**Answer :** def is\_leap(year):

    if year % 400 == 0:

        return True

    elif year % 100 == 0:

        return False

    elif year % 4 == 0:

        return True

    return False

year = int(input())

print(is\_leap(year))

**Output :** Input (stdin)

* **1990**

Your Output (stdout)

* **False**

Expected Output

* **False**