



Detect Floating Point Number ★

78/115 challenges solved

Rank: 43216 | Points: 1325



Your Detect Floating Point Number submission got 20.00 points.

[Share](#)[Post](#)[Try the next challenge](#) | [Try a Random Challenge](#)[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

Check [Tutorial](#) tab to know how to solve.

You are given a string ***N***.

Your task is to verify that ***N*** is a floating point number.

In this task, a valid float number must satisfy all of the following requirements:

> Number can start with +, - or . symbol.

For example:

✓ +4.50

✓ -1.0

✓ .5

✓ -.7

✓ +.4

✗ -+4.5

> Number must contain at least **1** decimal value.

For example:

✗ 12.

✓ 12.0

> Number must have exactly one . symbol.

> Number must not give any exceptions when converted using `float(N)`.

Input Format

The first line contains an integer ***T***, the number of test cases.

The next ***T*** line(s) contains a string ***N***.

Constraints

- $0 < T < 10$

Output Format

Output True or False for each test case.

Sample Input 0

```
4
4.000
-1.00
+4.54
SomeRandomStuff
```

Sample Output 0

False
True
True
False

Explanation 0

4.000: 0 is not a digit.

-1.00: is valid.

+4.54: is valid.

SomeRandomStuff: is not a number.

[Change Theme](#)

Language

Python 3



```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 t = int(input())
5 pattern = r"^[+-]?[0-9]*\.[0-9]+$"
6 for _ in range(t):
7     s = input()
8     try:
9         if re.fullmatch(pattern, s):
10             float(s)
11             print(True)
12         else:
13             print(False)
14     except ValueError:
15         print(False)
16
```

EMACS

Line: 16 Col: 1

[Upload Code as File](#)☐

Test against custom input

[Run Code](#)[Submit Code](#)

You have earned 20.00 points!

78/115 challenges solved.

68%



Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Compiler Message

Test case 1

Success

Test case 2

Input (stdin)

Download

Test case 3

1 4

Test case 4

2 4.000

Test case 5

3 -1.00

Test case 6

4 +4.54

5 SomeRandomStuff

Expected Output

Download