

01 | Link : <https://edabit.com/challenge/x5o7jTvzXjujvrh6t>

โจทย์

Iterated Square Root

Published by [Jerry Sun](#) in [Python](#) ▾

complete loops math numbers

The iterated square root of a number is the number of times the square root function must be applied to bring the number **strictly under 2**.

Given an integer, return its iterated square root. Return `"invalid"` if it is negative.

Examples

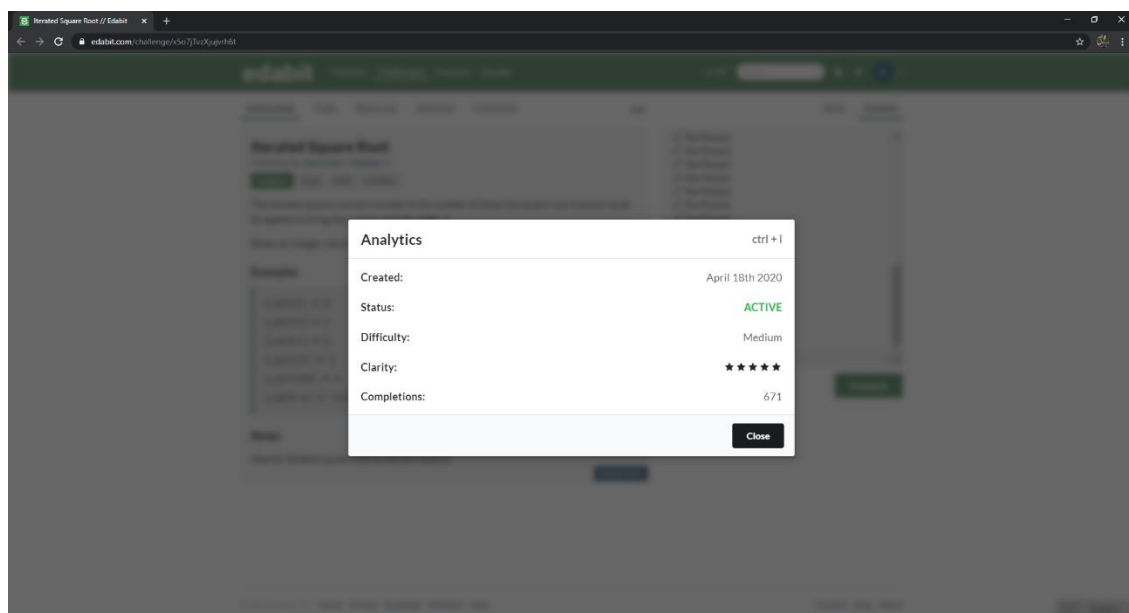
```
i_sqrt(1) → 0
i_sqrt(2) → 1
i_sqrt(7) → 2
i_sqrt(27) → 3
i_sqrt(256) → 4
i_sqrt(-1) → "invalid"
```

Notes

Idea for iterated square root by Richard Spence.

[SUGGEST EDIT](#)

ระดับความยาก



อธิบายโจทย์

สร้างฟังก์ชันให้หาการทำซ้ำของ Square Root ว่ามีการทำงานทั้งหมดกี่ครั้ง จนกว่าจะน้อยกว่า 2 และ ถ้ามีค่าเป็นลบให้รีเทิร์น "invalid" กลับไป

อธิบายการทำงานของโปรแกรม

ฟังก์ชัน `i_sqrt` คือการนับรอบการทำงานของ การทำซ้ำของ Square Root โดยที่มีเงื่อนไขให้ ตัวแปร `n` มากกว่า 0 ให้โปรแกรมตรวจสอบว่า `n` มีค่ามากกว่า 0 หรือไม่ถ้ามากกว่าให้ทำ ในบรรทัดถัดไปคือ ตัวแปร `count = 0` คือการกำหนดตัวแปร `count` เพื่อให้เก็บการทำซ้ำของฟังก์ชัน `i_sqrt` โดยที่มีการใช้ While Loop มาช่วยเก็บการทำซ้ำ while `n >= 2` คือ ให้วน Loop การทำงานจนกว่า `n` จะเท่ากับ 2 โดยมีการนำตัวแปร `count` มาบวกรอบในการทำของ While Loop โดยที่ใช้เงื่อนไข `n = sqrt(n)` เพื่อให้ `n` ทำงานกับฟังก์ชันทางคณิตศาสตร์ `sqrt` ฟังก์ชันทางคณิตศาสตร์ จนกว่าจะทำรากที่ 2 จนมีค่าน้อยกว่า 2 จึงหลุดจาก While Loop แล้ว return ค่า `count` กลับไปที่ฟังก์ชัน `i_sqrt` แต่ถ้าตัวแปร `n` มีค่าน้อยกว่า 0 ให้ return "invalid" กลับไป ที่ฟังก์ชัน `i_sqrt`

หน้าจอของผล run โปรแกรม

OUTPUT TERMINAL SQL CONSOLE DEBUG CONSOLE PROBLEMS

```
PS C:\Users\HP\Desktop\Edabit UnitTest> python3 '..\01 Iterated Square Root.py'
0
1
2
3
4
invalid
PS C:\Users\HP\Desktop\Edabit UnitTest> []
```

หน้าจอของผล run. Unit Test

```
OUTPUT  TERMINAL  SQL CONSOLE  ...  1: powershell  +  [ ]  [ ]  ^  x

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP\Desktop\Edabit UnitTest> python3 .\Unittest01.py
0
1
2
3
4
invalid

-----
Ran 0 tests in 0.000s

OK
PS C:\Users\HP\Desktop\Edabit UnitTest> █
```

Code Program

```
from math import sqrt
def i_sqrt(n):
    if n > 0:
        count = 0
        while n >= 2:
            count += 1
            n = sqrt(n)
        return count
    return 'invalid'

print(i_sqrt(1))
print(i_sqrt(2))
print(i_sqrt(7))
print(i_sqrt(27))
print(i_sqrt(256))
print(i_sqrt(-1))
```

Code Unit Test

```
import unittest
import IteratedSquareRoot01

class Test(unittest.TestCase):
    def i_sqrttest (self):
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(1))
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(2))
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(7))
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(27))
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(256))
        self.assertEqual(IteratedSquareRoot01.i_sqrttest(-1))

if __name__ == '__main__':
    unittest.main()
```