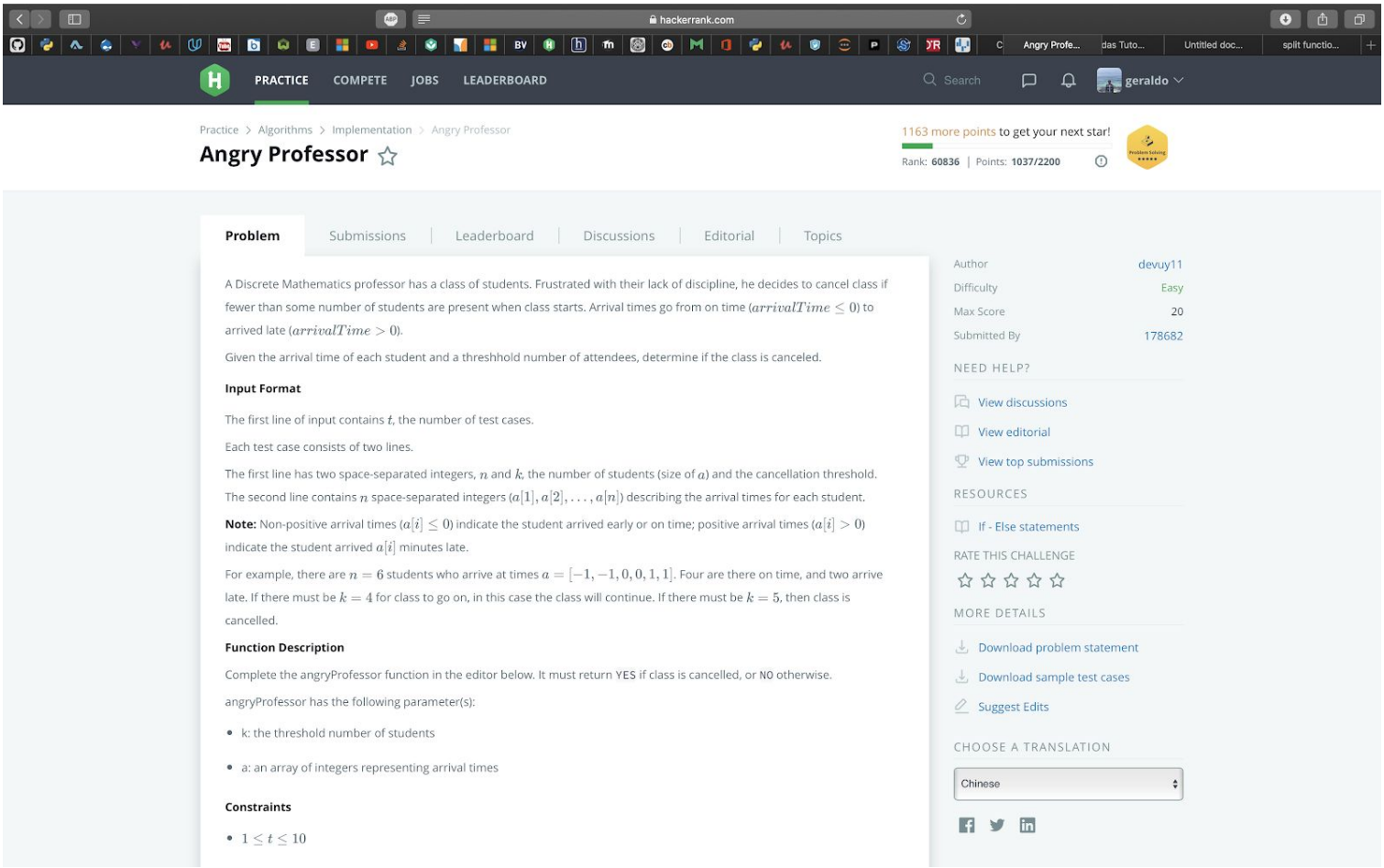
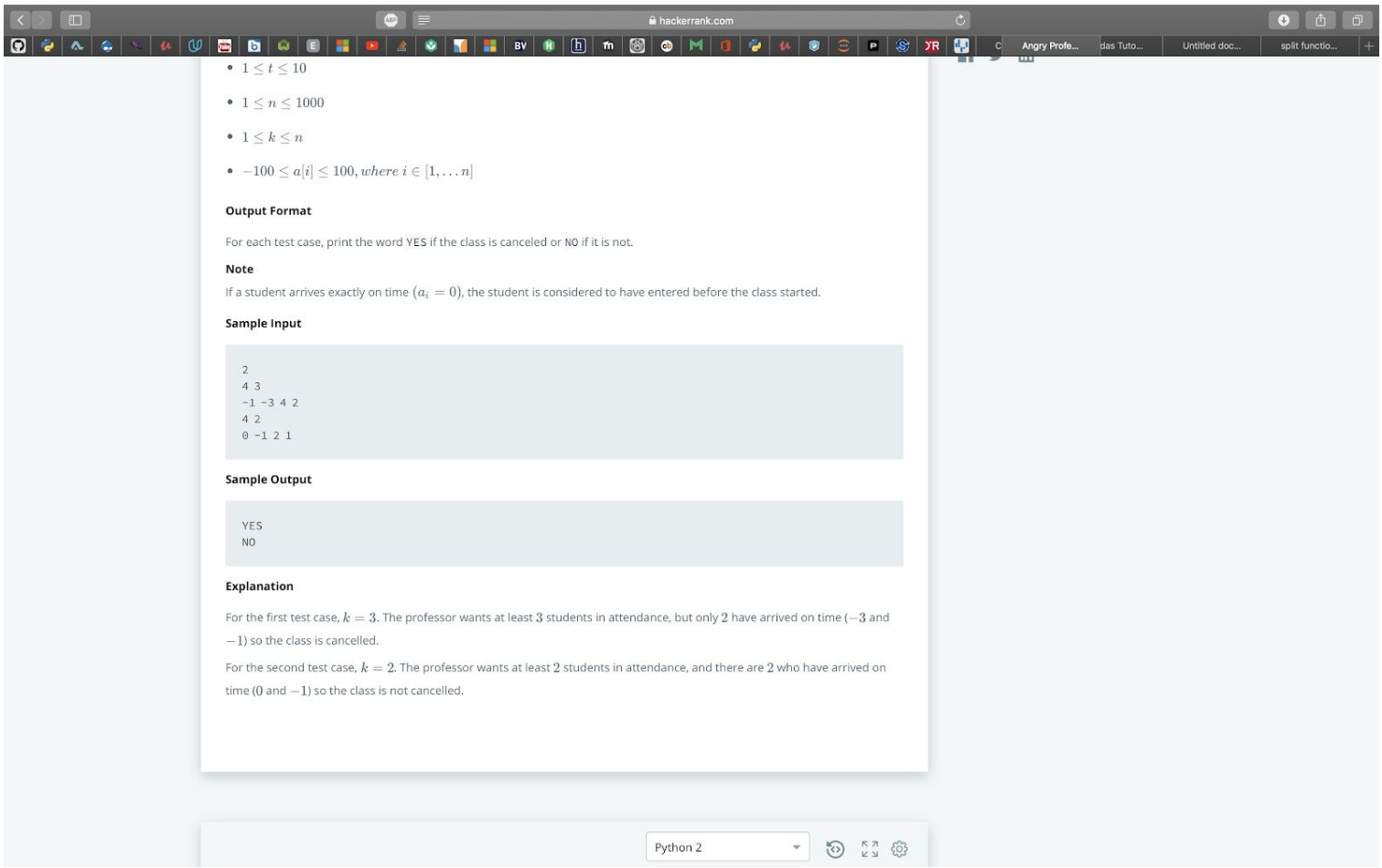
Geraldo Braho
Comp 5327 Advanced Algorithm

Please solve the following questions:

- https://www.hackerrank.com/challenges/angry-professor
- https://www.hackerrank.com/challenges/sherlock-and-squares
- https://www.hackerrank.com/challenges/utopian-tree

Question 1 https://www.hackerrank.com/challenges/angry-professor





- $1 \leq t \leq 10$
- $1 \leq n \leq 1000$
- $1 \leq k \leq n$
- $-100 \leq a[i] \leq 100, where\ i \in [1, \ldots n]$

Output Format

For each test case, print the word `YES` if the class is canceled or `NO` if it is not.

Note

If a student arrives exactly on time $(a_i = 0)$, the student is considered to have entered before the class started.

Sample Input

```
2
4 3
-1 -3 4 2
4 2
0 -1 2 1
```

Sample Output

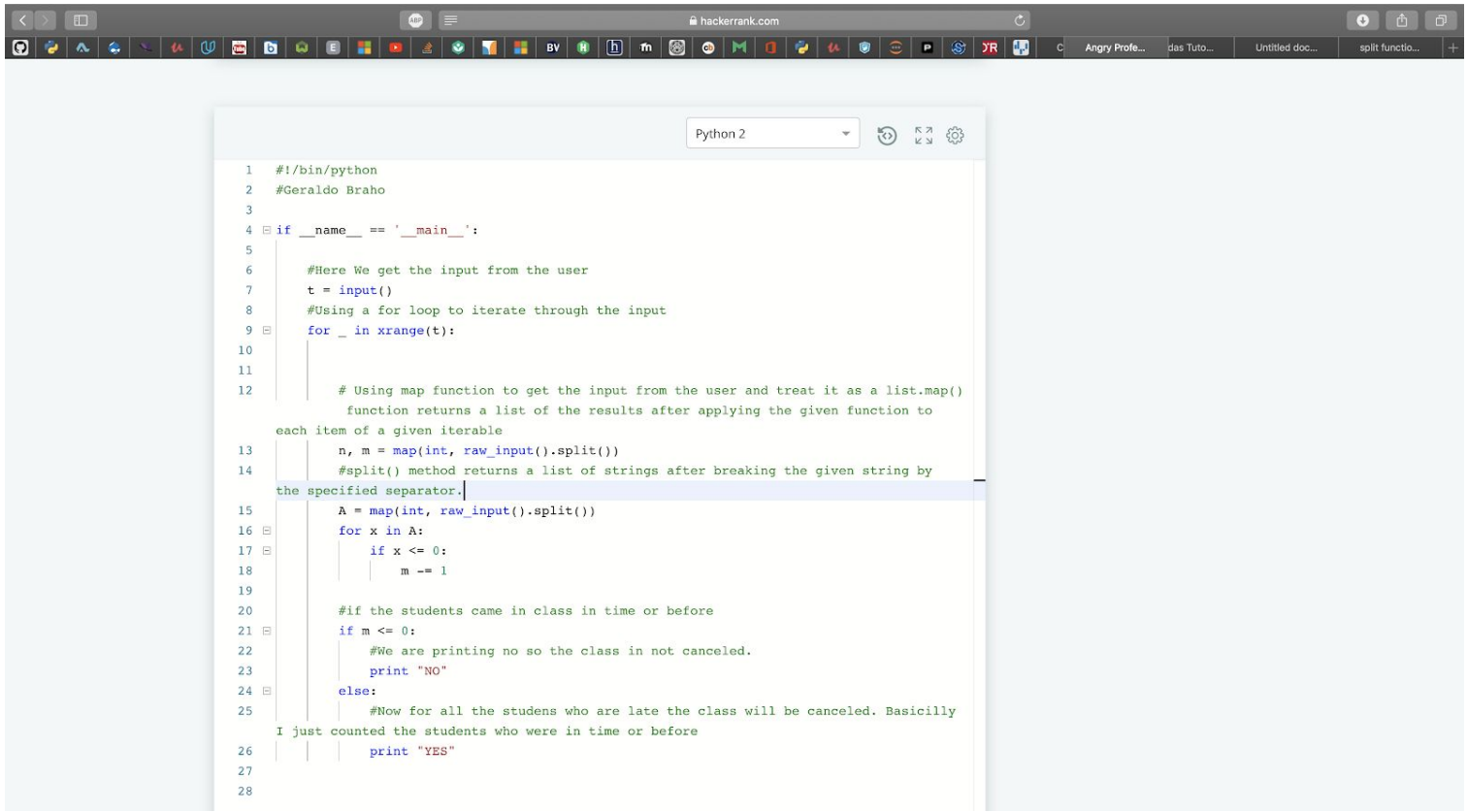
```
YES
NO
```

Explanation

For the first test case, $k = 3$. The professor wants at least 3 students in attendance, but only 2 have arrived on time (-3 and -1) so the class is cancelled.

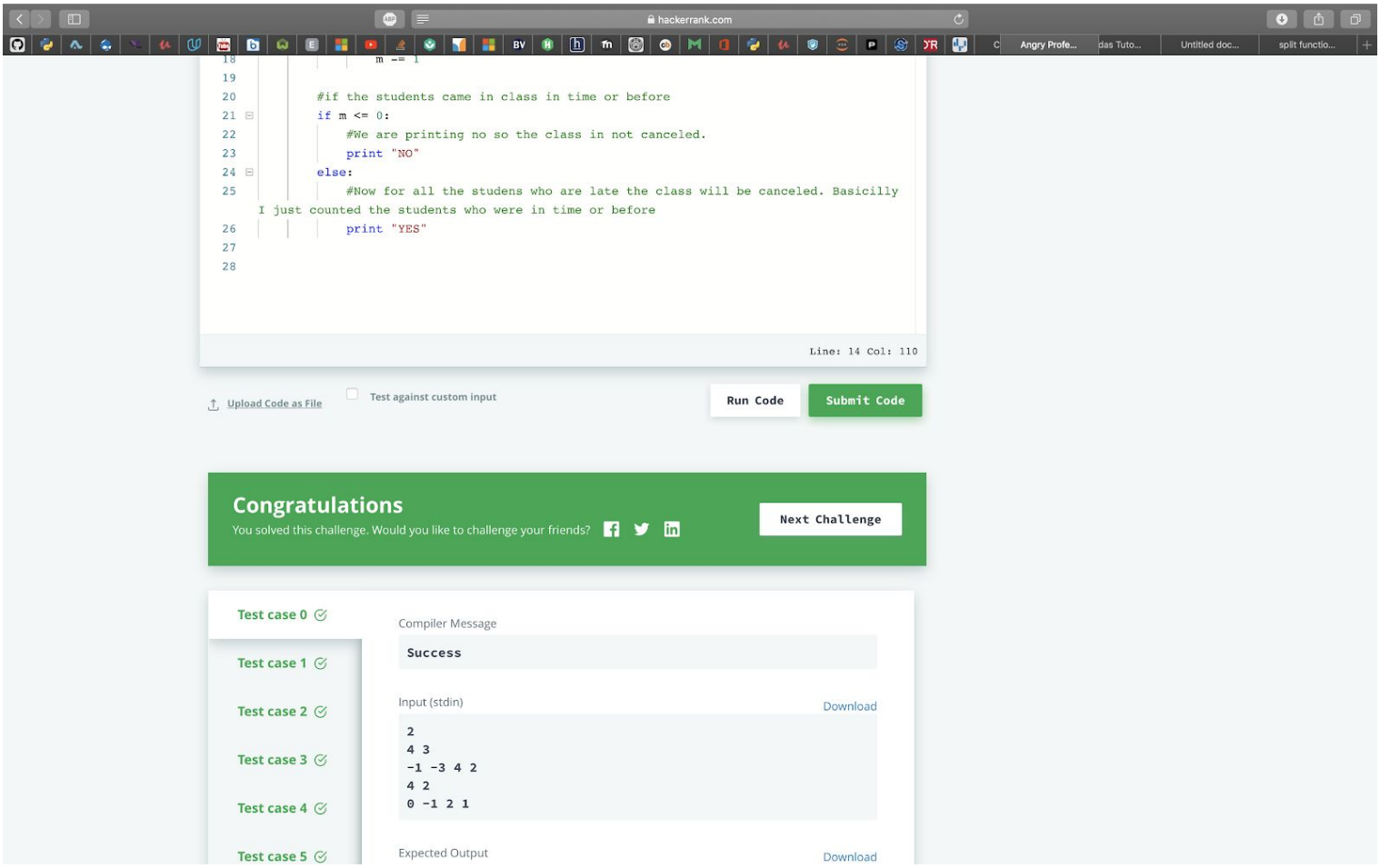
For the second test case, $k = 2$. The professor wants at least 2 students in attendance, and there are 2 who have arrived on time (0 and -1) so the class is not cancelled.

Python 2

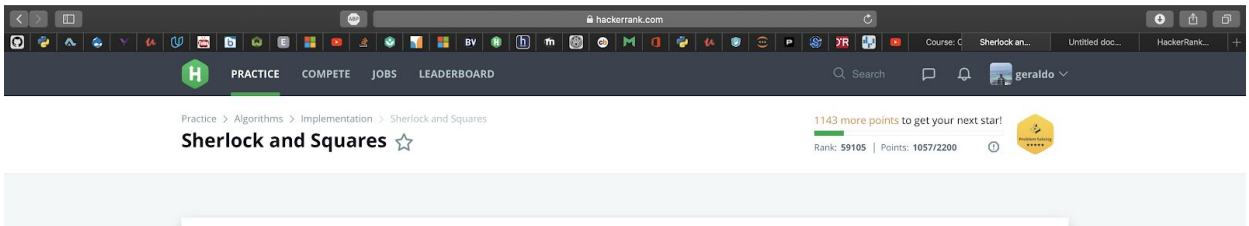


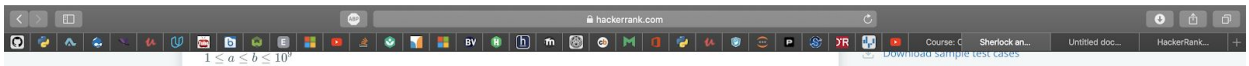
Python 2

```python
#!/bin/python
#Geraldo Braho

if __name__ == '__main__':

    #Here We get the input from the user
    t = input()
    #Using a for loop to iterate through the input
    for _ in xrange(t):



        # Using map function to get the input from the user and treat it as a list.map() function returns a list of the results after applying the given function to each item of a given iterable
        n, m = map(int, raw_input().split())
        #split() method returns a list of strings after breaking the given string by the specified separator.
        A = map(int, raw_input().split())
        for x in A:
            if x <= 0:
                m -= 1

        #if the students came in class in time or before
        if m <= 0:
            #We are printing no so the class in not canceled.
            print "NO"
        else:
            #Now for all the studens who are late the class will be canceled. Basicilly I just counted the students who were in time or before
            print "YES"


```



Question 2 https://www.hackerrank.com/challenges/sherlock-and-squares





$1 \leq a \leq b \leq 10^9$

Output Format

For each test case, print the number of square integers in the range on a new line.

Sample Input

```
2
3 9
17 24
```

Sample Output

```
2
0
```

Explanation

Test Case #00: In range $[3, 9]$, 4 and 9 are the two square integers.
Test Case #01: In range $[17, 24]$, there are no square integers.

Download Sample test cases

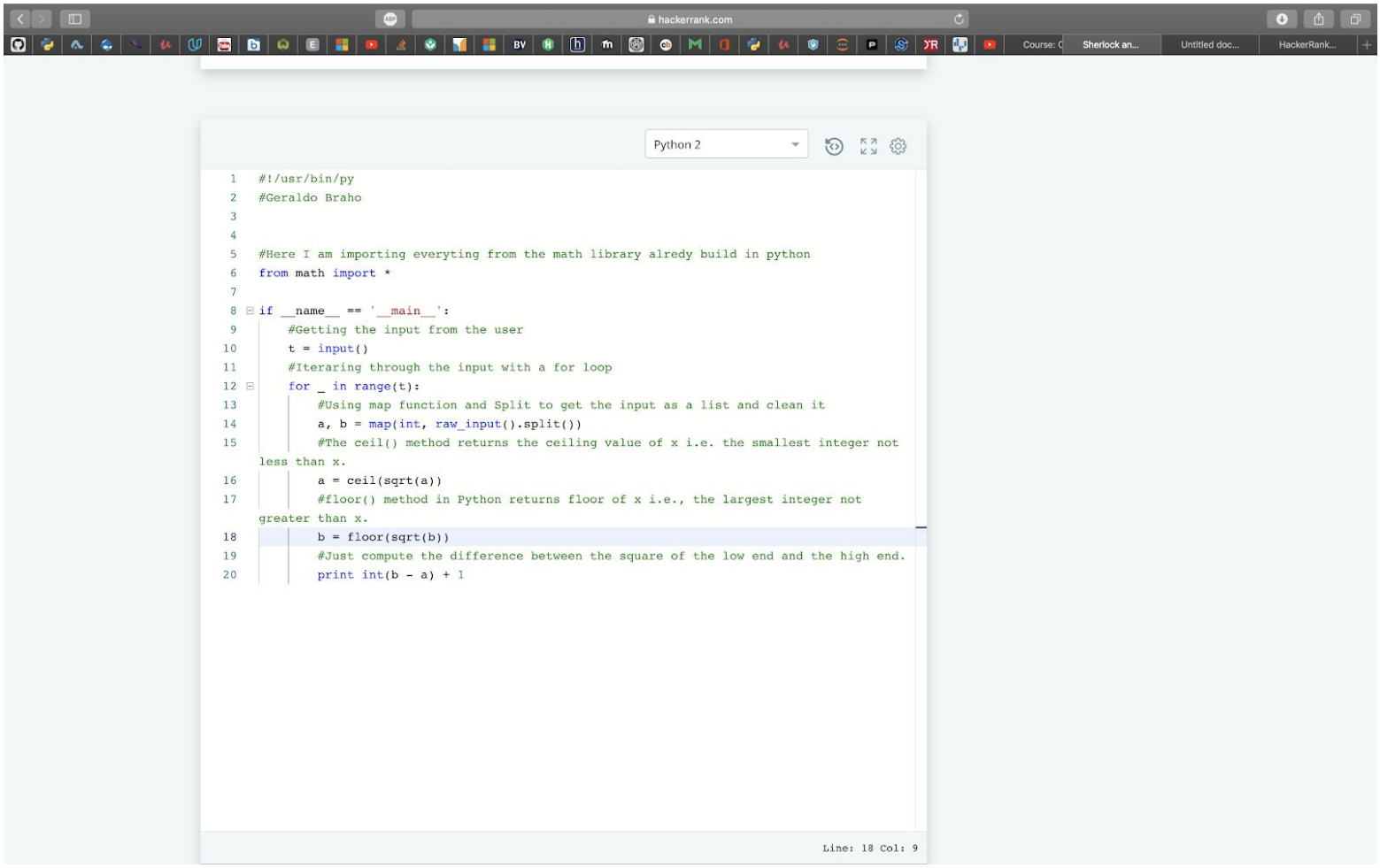
Suggest Edits

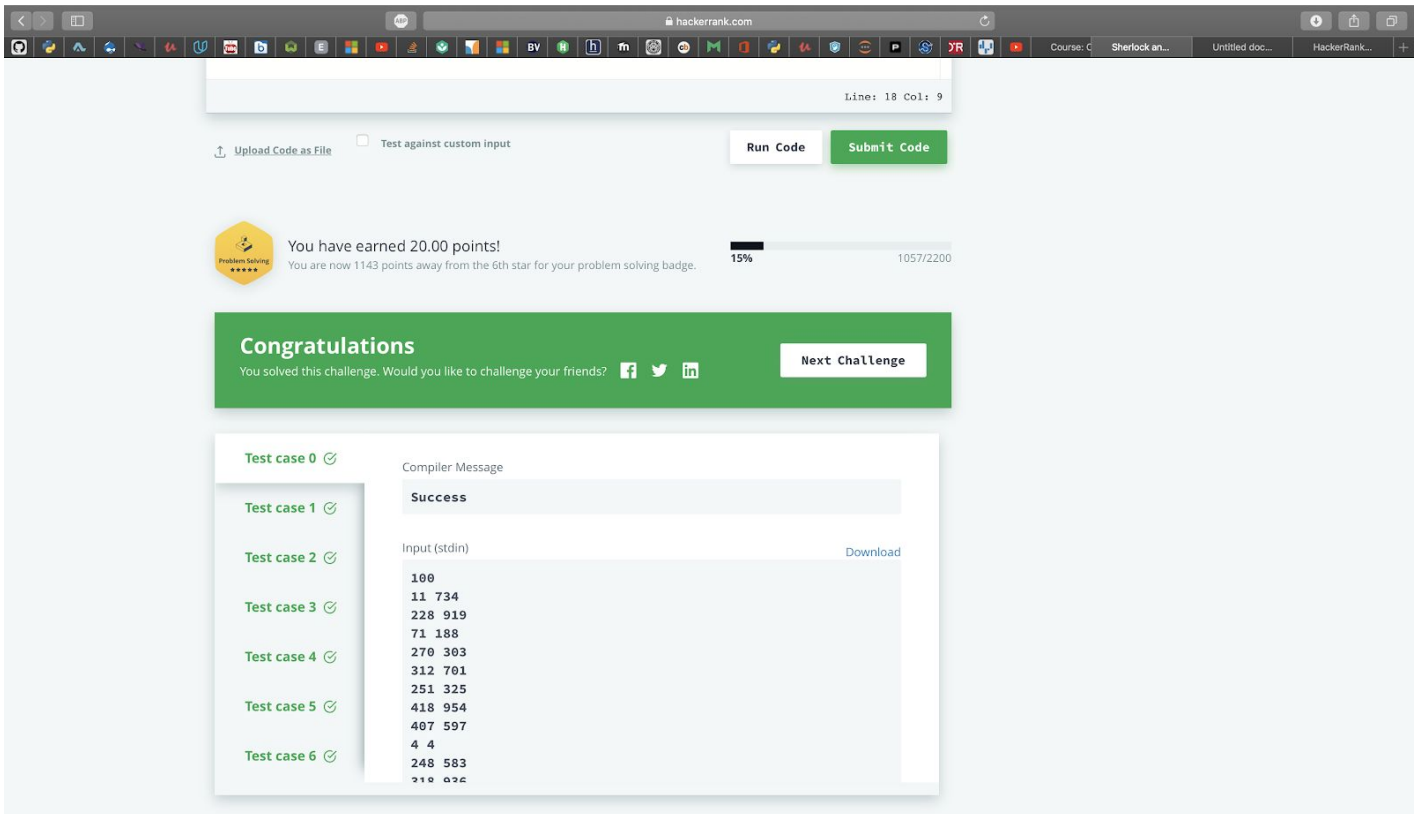
CHOOSE A TRANSLATION

Chinese

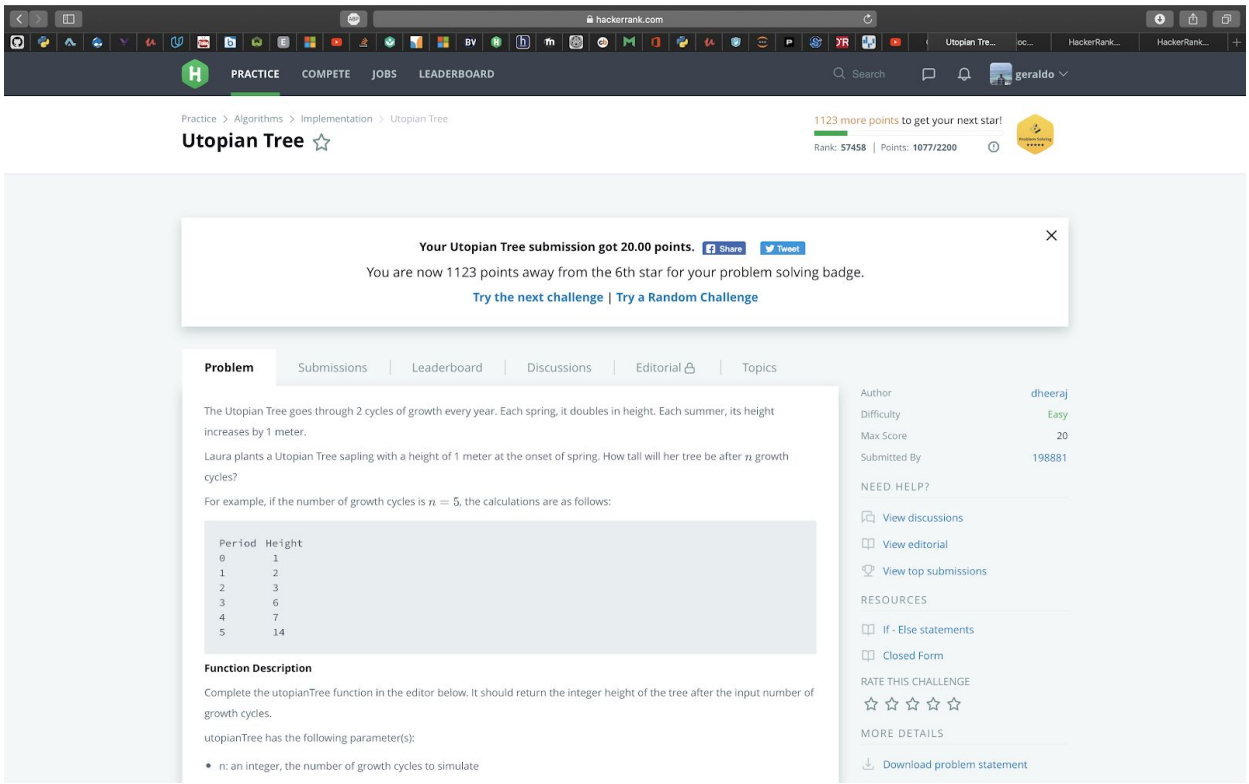
Python 2

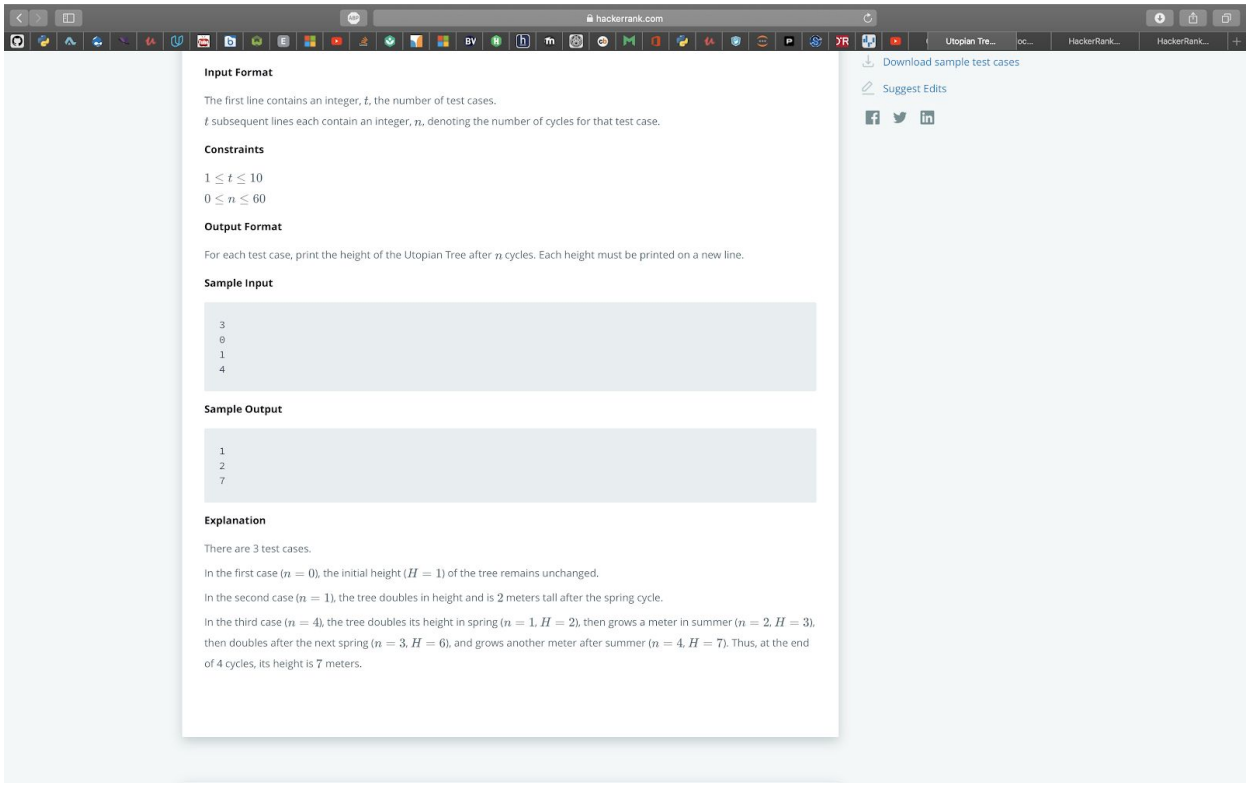
```
#!/usr/bin/py
#Geraldo Braho


#Here I am importing everyting from the math library alredy build in python
from math import *

if __name__ == '__main__':
    #Getting the input from the user
    t = input()
    #Iteraring through the input with a for loop
    for _ in range(t):
        #Using map function and Split to get the input as a list and clean it
```


hackerrank.com
Python 2
#!/usr/bin/py
#Geraldo Braho
#Here I am importing everyting from the math library alredy build in python
from math import *
if __name__ == '__main__':
#Getting the input from the user
t = input()
#Iteraring through the input with a for loop
for _ in range(t):
#Using map function and Split to get the input as a list and clean it
a, b = map(int, raw_input().split())
#The ceil() method returns the ceiling value of x i.e. the smallest integer not less than x.
a = ceil(sqrt(a))
#floor() method in Python returns floor of x i.e., the largest integer not greater than x.
b = floor(sqrt(b))
#Just compute the difference between the square of the low end and the high end.
print int(b - a) + 1
Line: 18 Col: 9


hackerrank.com
Line: 18 Col: 9
Upload Code as File
Test against custom input
Run Code
Submit Code
You have earned 20.00 points!
You are now 1143 points away from the 6th star for your problem solving badge.
15%
1057/2200
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
100
11 734
228 919
71 188
270 303
312 701
251 325
418 954
407 597
4 4
248 583

Question 3 - https://www.hackerrank.com/challenges/utopian-tree





Input Format

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

t subsequent lines each contain an integer, n, denoting the number of cycles for that test case.

Constraints

$1 \leq t \leq 10$
$0 \leq n \leq 60$

Output Format

For each test case, print the height of the Utopian Tree after n cycles. Each height must be printed on a new line.

Sample Input

```
3
0
1
4
```

Sample Output

```
1
2
7
```

Explanation

There are 3 test cases.

In the first case ($n = 0$), the initial height ($H = 1$) of the tree remains unchanged.

In the second case ($n = 1$), the tree doubles in height and is 2 meters tall after the spring cycle.

In the third case ($n = 4$), the tree doubles its height in spring ($n = 1$, $H = 2$), then grows a meter in summer ($n = 2$, $H = 3$), then doubles after the next spring ($n = 3$, $H = 6$), and grows another meter after summer ($n = 4$, $H = 7$). Thus, at the end of 4 cycles, its height is 7 meters.

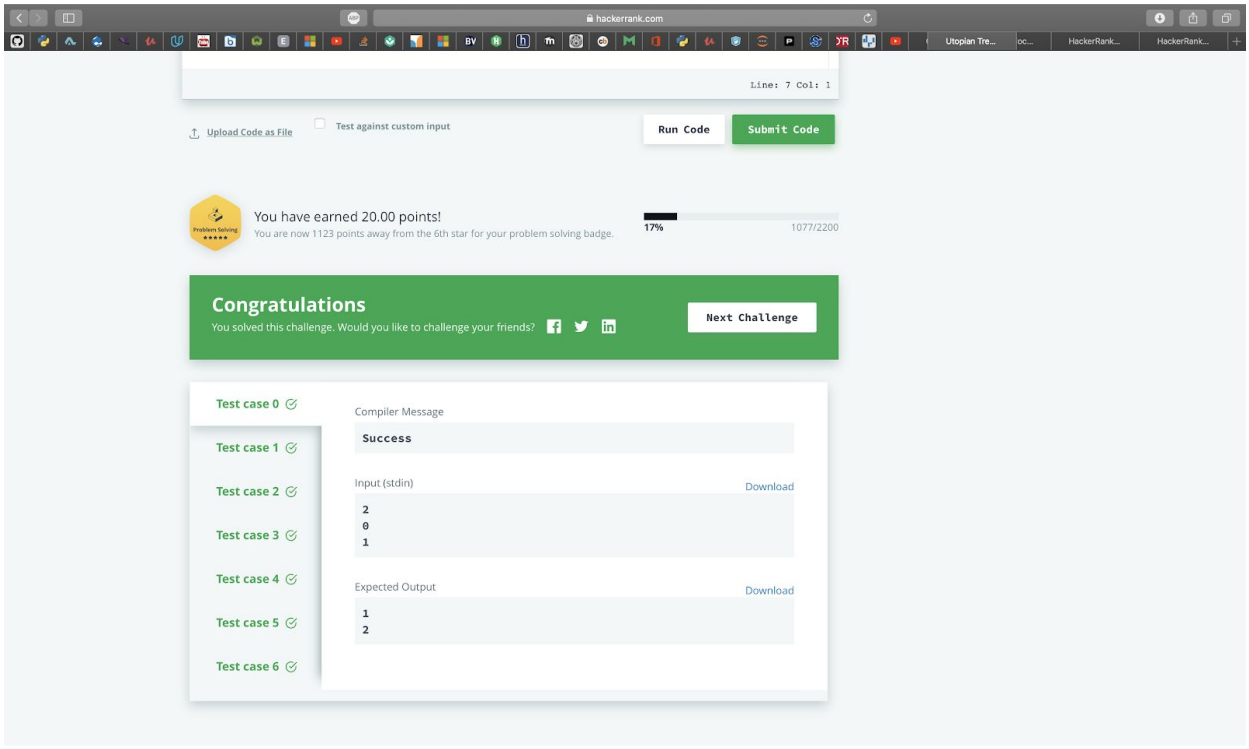
Download sample test cases

Suggest Edits

Python 3

```python
#I had overlooked the fact that they always started out by being 1 meter high, so on the back of an enveloped I assumed that they started out being x meters tall. However, as it turns out that maede it easier to find the pattern. I basically just jotted down the development for a few cycles which can be written as
# 0    x    x
# 1    2x    2x + 1
# 2    2(2x +1) = 4x + 2    4x + 3
# 3    2(4x + 3) = 8x + 6    8x + 7
# 4    2(8x +7) = 16x + 14    16x + 15

#where x is the starting height. Year zero is a possible input, which means that it did not have time to grow. Therefore I have just noted it as x.If we start out by looking at the yearly cycle (after the summer growth) we can see that the factor is always 1 larger than the constant. We can also see that the grow as an exponential of 2. Which means we can express it as h = 2^mx  2^m -1 Where m is the number of years it has grown which is equal to m = \lfloor n/2 \rfloor or integer division by 2 of the cycles. Since x = 1 is always true we can simplify the solution to h = 2^m  2^m -1 = 2^{m 1} -1 Then in order to get the correct height, we must check if the cycles are odd, if that is the case we need to multiply  the height with 2 as we have gone through another spring growth.

def utopianTree(n):
    height = 2**((n // 2) + 1) - 1
    return height if n % 2 == 0 else height*2


if __name__ == "__main__":
    t = int(input().strip())
    for a0 in range(t):
        n = int(input().strip())
        result = utopianTree(n)
        print(result)
```

Line: 7 Col: 1


hackerrank.com
Line: 7 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
You have earned 20.00 points!
You are now 1123 points away from the 6th star for your problem solving badge.
17%
1077/2200
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
2
0
1
Expected Output
Download
1
2