Infinite Grid

You are standing on an infinitely large grid with many rows and columns.

Each cell (x, y) in the grid contains the integer x * y (the product of the row number and column number). Initially, you are standing at (1, 1). In one move, you can move from cell (x, y) to either

(x, y + 1) or (x + 1, y) cell.

You are given an integer N, find the minimum number of moves needed to reach a cell that contains the value N.

Constraints:

$2 <= N <= 10^{12}$

N is an integer

Input:

N

Note that the input value might not fit into the 32-bit integer data type.

Output:

Print the minimum number of moves needed to reach a square that contains the integer N

Sample Testcases:

Input: 10

Output: 5

Explanation: Cell (2,5) can be reached in five moves. We cannot reach any cell that contains 10 in less than five moves.

Input: 50

Output: 13

Input: 10000000019

Output: 10000000018

Problem Statistics

Author: Muhesh Kumar B

Solved By: 3

View leaderboard

Theme: Terminal Language Python Font size: 18

```python

import math
n = int(input())
point = (1,1)
factors = []
for i in range(1,int(math.sqrt(n)+1)):
    if n%i==0:
        factors.append([i,n//i])
def steps_to(pair):
    steps = pair[0]+pair[1]-2
    return steps
steps_req = n
for pair in factors:
    if steps_to(pair)<steps_req:
        steps_req = steps_to(pair)
print(steps_req)

```

Use custom I/O

Run Code | Save Code | Pause Test

Status:

```
Success your code has passed all test cases!!
```