Complete Search
Complete Search

- Aka “brute force”
- Simply try every possibility
  - Won’t work for every problem!
  - Feasible only for small domains
Complete Search

- Example 1: Selection sort
  - For each position find the max value (from remainder of list)
- Example 2: Naïve GCD (greatest common divisor)
  - Try all $k$ from $\min(m,n)$ to 1
  - Check whether $k$ divides $m$ and $n$, if so we’ve found GCD
- Example 3: Primality test
  - Try all $k$ from 2 to $\sqrt{n}$
  - Check whether $k$ divides $n$, if so $n$ is not prime
Complete Search

- General tips:
  - Prune the search space if possible (don’t waste time checking impossible candidates)
  - Utilize symmetries
  - Pre-compute whatever you can
  - Try working backwards
  - Use a better data structure
Candy Division

Benny has just found out that he has $N$ candies in his pocket. He is always happy to share candies with his friends, because then he doesn’t have to eat them alone and get fat. There is one problem, though. He cannot invite any number of friends, since it would be impossible to divide candies evenly among people.

Given $N$, how many people can Benny invite to eat candies, so that everybody gets the same number of candies and all candies will be eaten?

**Input**

The input contains one line with one integer $N$ ($2 \leq N \leq 10^{12}$), denoting the number of candies in Benny’s pocket.

**Output**

Output one line with all possible number of friends Benny can invite. Output them in ascending order and separate by spaces.

**Sample Input 1**

```
30
```

**Sample Output 1**

```
0 1 2 4 5 9 14 29
```
import math

N = int(input())
sqrtN = int(math.floor(math.sqrt(N)) + 1)

nums = set()
for i in range(1, sqrtN):
    if N % i == 0:
        nums.add(i - 1)
        nums.add(N // i - 1)

print(' '.join([str(int(s)) for s in sorted(nums)]))
In-class problem

Falling Mugs

Susan is making high-speed videos of falling coffee mugs. When analyzing the videos she wants to know how big the mugs are, but unfortunately they all got destroyed during filming. Susan knows some physics though, so she has figured out how far they moved between different video frames. The camera was filming at a speed of 70 frames per second, which means that at frame $n$, counted from when the mug was released, the number of millimeters it has moved is $d = n^2$. The counting of the frames starts at 0.

Susan thinks a certain mug is $D$ millimeters high, but to verify this she needs to find two frames between which the mug has moved exactly this distance. Can you help her do this?

**Input**

The input file contains the single positive integer $D \leq 200000$, the distance to be measured.

**Output**

Output two non-negative integers $n_1$ and $n_2$, the numbers of the frames that Susan should compare. They should fulfill $n_2^2 - n_1^2 = D$. If no two such integers exist, instead output “impossible”.

<table>
<thead>
<tr>
<th>Sample Input 1</th>
<th>Sample Output 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>9 13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample Input 2</th>
<th>Sample Output 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>impossible</td>
</tr>
</tbody>
</table>