NEW

Q Search

jigarpandya_ce ⌄

All Contests > 22_PPS1_Lab5 > The Euclidean Algorithm

The Euclidean Algorithm   🔒 locked

Problem	Submissions	Leaderboard	Discussions

Find GCD of two posititve numbers using The Euclidean Algorithm.

Recall that the Greatest Common Divisor (GCD) of two numbers m and n is the largest number that divides both m and n.

The Euclidean Algorithm is a technique for quickly finding the GCD of two integers.

Define function named *find_gcd* which takes two numbers and returns a number. Use euclidean algorithm to implement *find_gcd* function. Call *find_gcd* function from main with input numbers as arguments and print number returned by *find_gcd* function.

The Euclidean Algorithm for finding GCD(A,B) is as follows:

```
If A = 0 then GCD(A,B)=B, since the GCD(0,B)=B, and we can stop.
If B = 0 then GCD(A,B)=A, since the GCD(A,0)=A, and we can stop.
Write A in quotient remainder form (A = B·Q + R)
Find GCD(B,R) using the Euclidean Algorithm since GCD(A,B) = GCD(B,R)
```

Example:

Find the GCD of 270 and 192

```
A=270, B=192
A ≠0
B ≠0
Use long division to find that 270/192 = 1 with a remainder of 78. We can write this as: 270 = 192 * 1 +78
Find GCD(192,78), since GCD(270,192)=GCD(192,78)

A=192, B=78

A ≠0
B ≠0
Use long division to find that 192/78 = 2 with a remainder of 36. We can write this as:
192 = 78 * 2 + 36
Find GCD(78,36), since GCD(192,78)=GCD(78,36)

A=78, B=36

A ≠0
B ≠0
Use long division to find that 78/36 = 2 with a remainder of 6. We can write this as:
78 = 36 * 2 + 6
Find GCD(36,6), since GCD(78,36)=GCD(36,6)

A=36, B=6

A ≠0
B ≠0
Use long division to find that 36/6 = 6 with a remainder of 0. We can write this as:
36 = 6 * 6 + 0
Find GCD(6,0), since GCD(36,6)=GCD(6,0)

A=6, B=0
```

```
A ≠0
B =0, GCD(6,0)=6
```

For more details on how Euclidean algorithm works, please refer following link:

https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm

Input Format

Two positive integers (*m* and *n*) separated by space.

Constraints

1 <= *m*, *n* <= 10000000000000000000

Output Format

One integer - GCD of of numbers in input.

Sample Input 0

```
270 192
```

Sample Output 0

```
6
```

Sample Input 1

```
91 93
```

Sample Output 1

```
1
```

Sample Input 2

```
44 22
```

Sample Output 2

```
22
```

Sample Input 3

```
97 91
```

Sample Output 3

```
1
```

Sample Input 4

```
33 66
```

Sample Output 4

33

Sample Input 5

1 2

Sample Output 5

1

Sample Input 6

1 1

Sample Output 6

1

Sample Input 7

3243434443434344838 9897347893459873458

Sample Output 7

6

Submissions: 139
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

⬇ Download problem statement

⬇ Download all test cases

✎ Suggest Edits

Collapse

Admin Options

✏ Edit Challenge

View Submissions

```
#include <stdio.h>
#include <string.h>
```

```
 3   #include <math.h>
 4   #include <stdlib.h>
 5
 6▼  int main() {
 7
 8▼      /* Enter your code here. Read input from STDIN. Print output to STDOUT */
 9       return 0;
10   }
11
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |