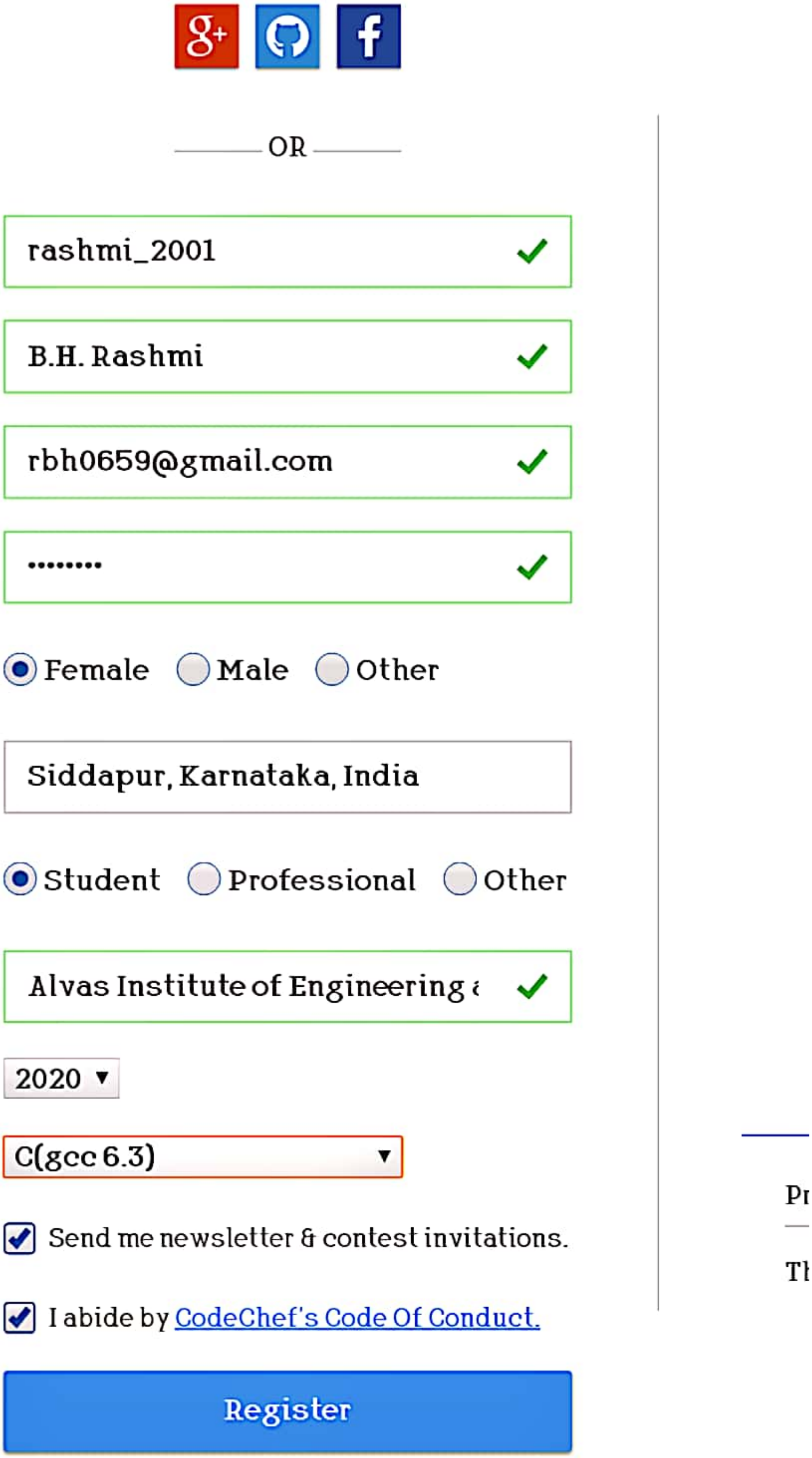


rashmi_2001 ✓

B.H. Rashmi ✓

rbh0659@gmail.com ✓

•••••••• ✓

● Female ○ Male ○ Other

Siddapur, Karnataka, India

● Student ○ Professional ○ Other

Alvas Institute of Engineering ✓

2020 ▼

C(gcc 6.3) ▼

☑ Send me newsletter & contest invitations.

☑ I abide by CodeChef's Code Of Conduct.

## B.H. Rashmi

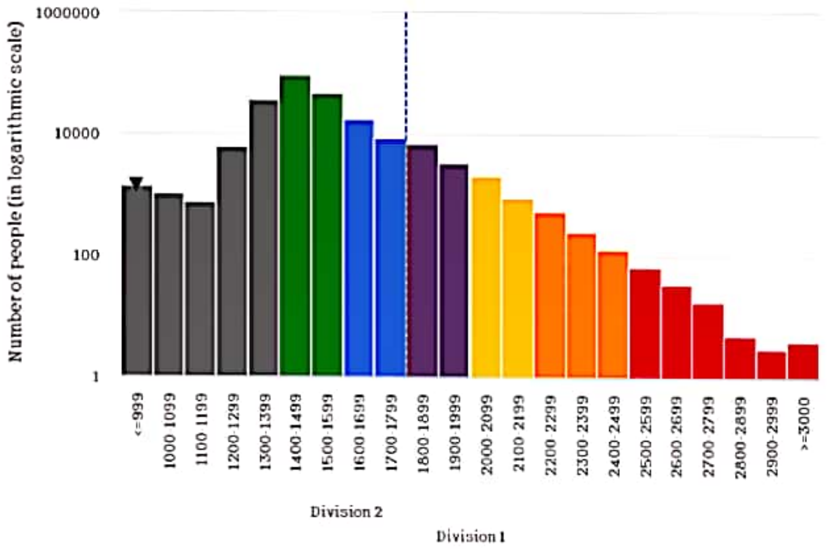
| | |
|---|---|
| Username: | rashmi_2001 |
| Country: | India |
| State: | Karnataka |
| City: | Siddapur |
| Student/Professional: | Student |
| Institution: | Alvas Institute of Engineering and Technology Karnataka, India |
| Teams List: | List of teams by B.H. Rashmi |
| Team Invites: | Click here to check team invites. 0 |

## Rating Graphs

CodeChef | Long Challenge | Cook Off | Lunch Time

3500
3000
2500
2000
1500
1000
Rating
No Participation
2011
2012
2013
2014
2015
2016
2017
Year

## CodeChef Rating Distribution



0

★

CodeChef Rating

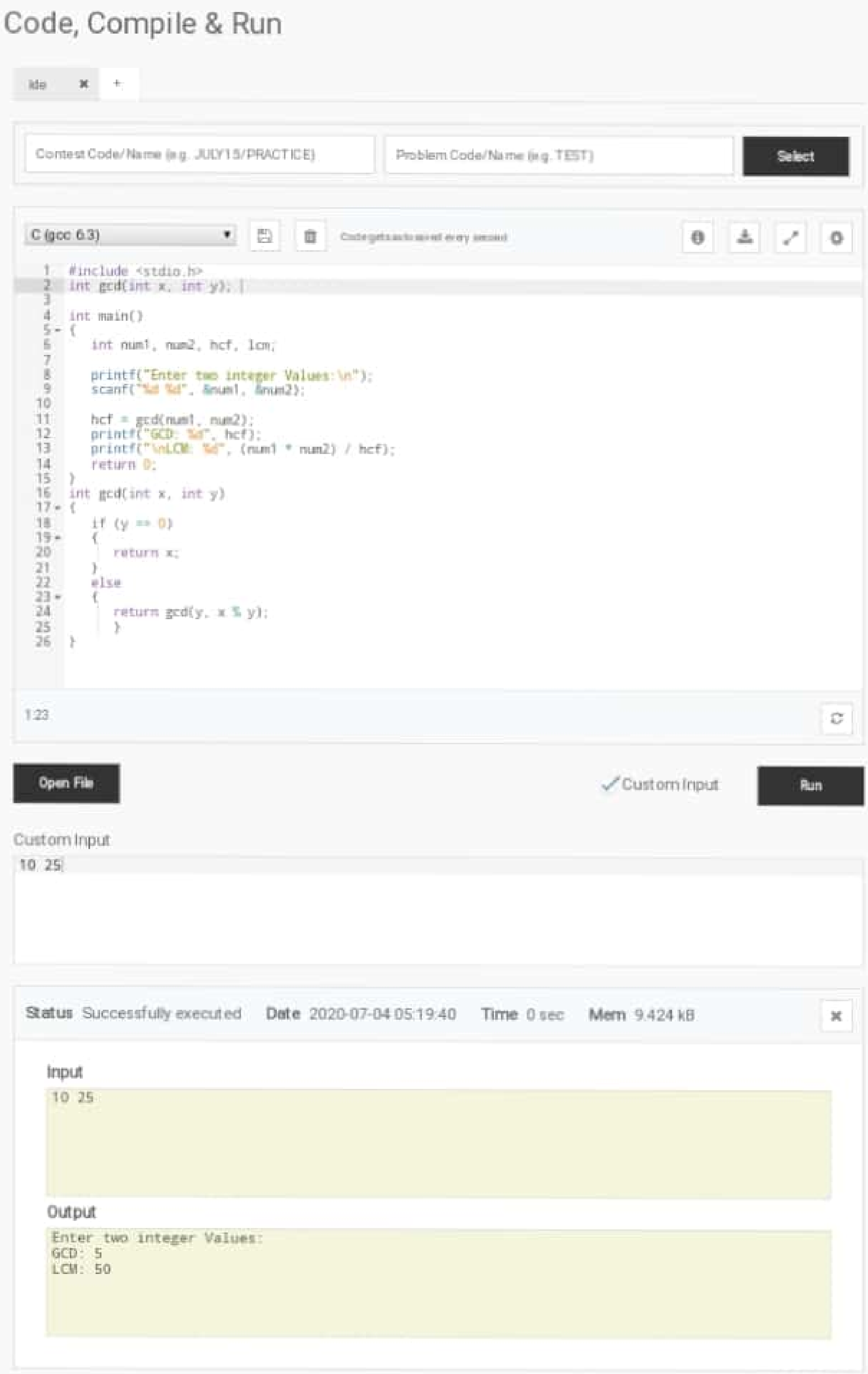
(Highest Rating 0)

| NA | NA |
|---|---|
| Global Rank | Country Rank |

| Contests | Rating | Global Rank | Country Rank |
|---|---|---|---|
| Long Challenge | 0 | NA | NA |
| Cook-off | 0 | NA | NA |
| Lunch Time | 0 | NA | NA |

### Recent Activity

| Date/Time | Problem | Result | Lang |
|---|---|---|---|
| No Recent Activity | | | |



# Code, Compile & Run

ide ✖ +
Contest Code/Name (e.g. JULY15/PRACTICE)
Problem Code/Name (e.g. TEST)
Select
C (gcc 6.3)
Code gets auto saved every second
1:23
Open File
✓Custom Input
Run

```c
#include <stdio.h>
int gcd(int x, int y);

int main()
{
   int num1, num2, hcf, lcm;

   printf("Enter two integer Values:\n");
   scanf("%d %d", &num1, &num2);

   hcf = gcd(num1, num2);
   printf("GCD: %d", hcf);
   printf("\nLCM: %d", (num1 * num2) / hcf);
   return 0;
}
int gcd(int x, int y)
{
   if (y == 0)
   {
      return x;
   }
   else
   {
      return gcd(y, x % y);
      }
}
```

Custom Input

```
10 25
```

**Status** Successfully executed **Date** 2020-07-04 05:19:40 **Time** 0 sec **Mem** 9.424 kB

Input

```
10 25
```

Output

```
Enter two integer Values:
GCD: 5
LCM: 50
```

Algorithm :[GCD]

step 1 :- Start

step 2 :- Drop any negative signs.

step 3 :- Assign the larger to the dividend, the number to the divisor.

step 4 :- compute the quotient & reminder

step 5 :- Reassign the divisor to be the dividend & the remainder to be the divisor

step 6 :- Repeat steps 3-4 units the remainder is zero

step 7 :- stop.

Algorithm (LCD):-

step 1 :- start

step 2 :- Make R/W low

step 3 :- Make RS=0; if data byte is command RS=1 ; if data byte is data (ASCII value)

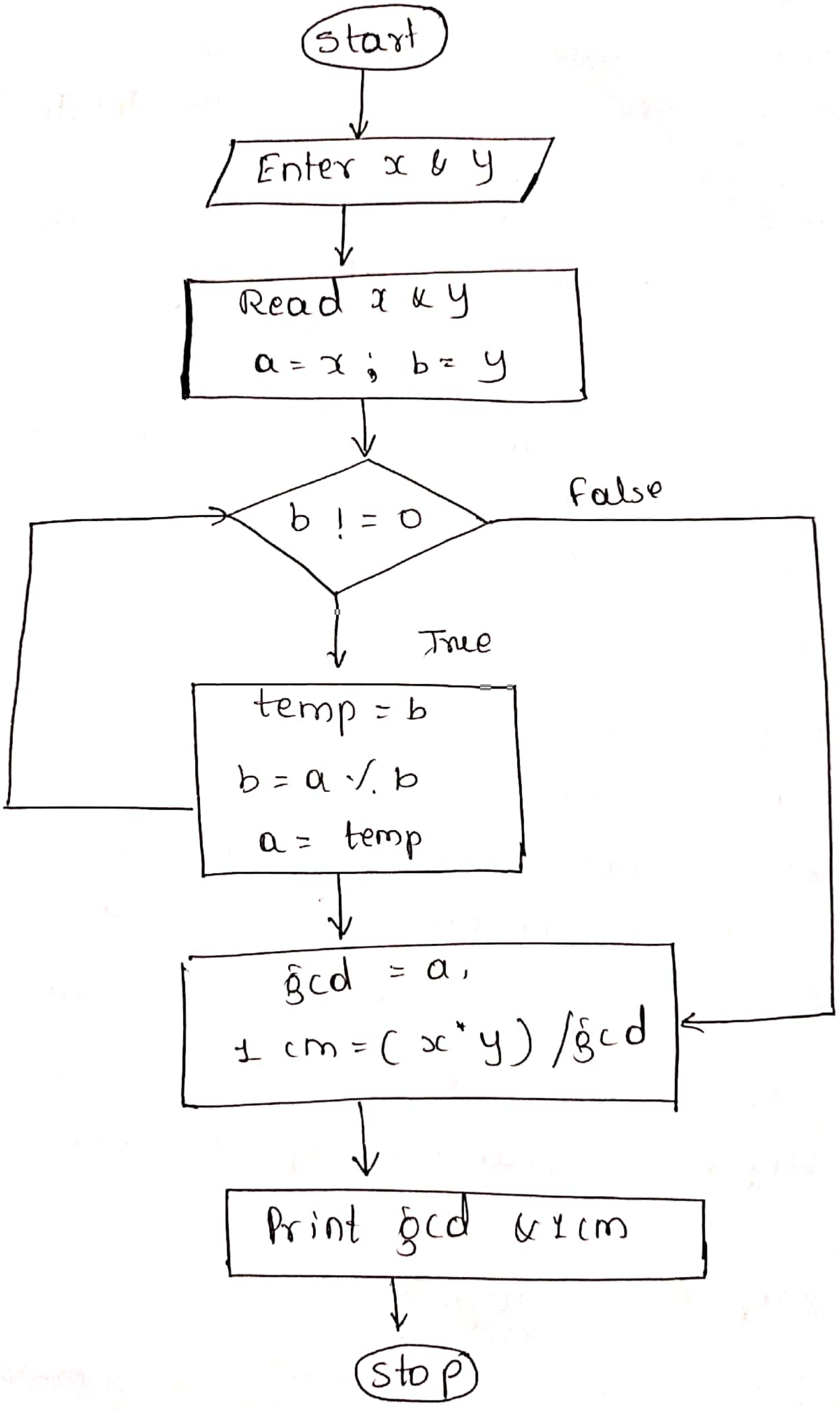
step 4 :- Place data byte on data register

step 5 :- Pulse E (High to low)

step 6 :- Repeat the steps to send another data byte

step 7 :- stop.

Flowchart


Start
Enter x & y
Read x & y
a = x ; b = y
b != 0
False
True
temp = b
b = a % b
a = temp
gcd = a,
lcm = ( x*y ) /gcd
Print gcd & lcm
stop