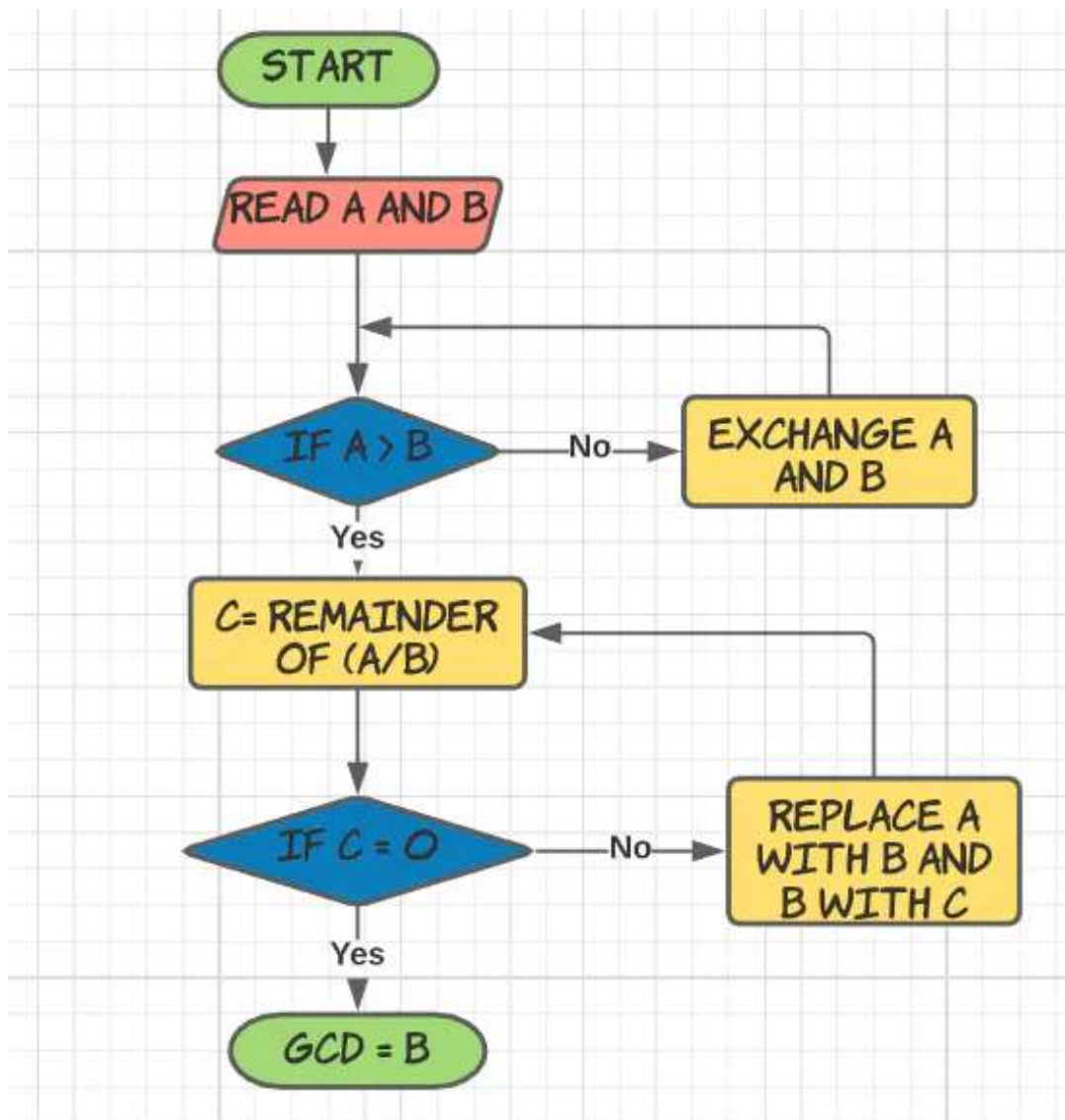


Flowchart and Pseudocode

Flowchart

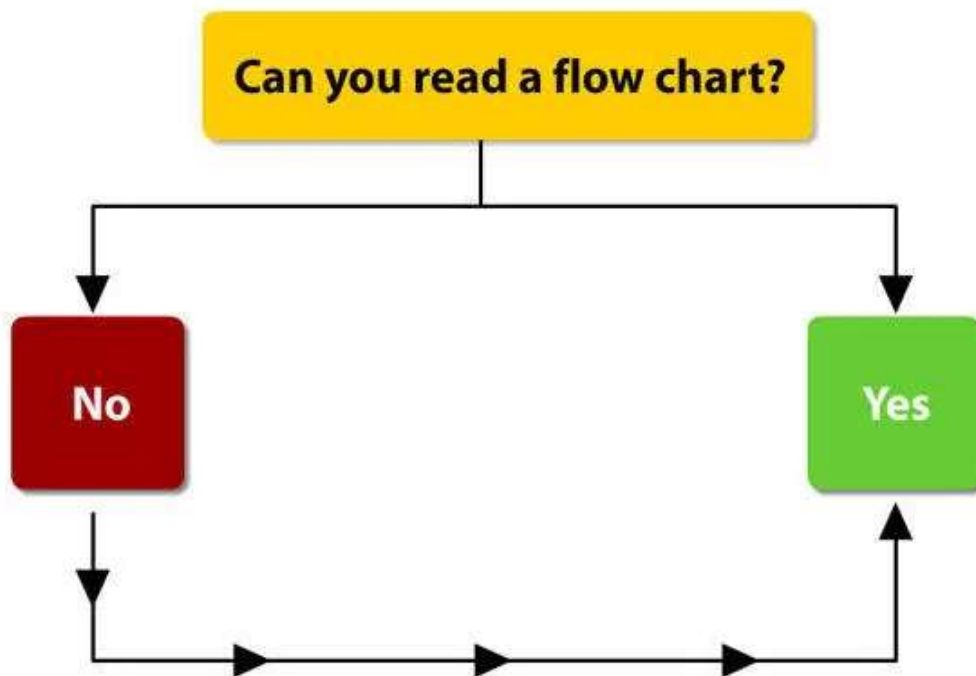
As we said, during writing code, coders often can get confused due to the complexity of various algorithms. As a result, it becomes difficult to shape the flow of the program. Flowcharts are interesting tools to help us out and to get a broad idea of our code. Let us understand them by making a flowchart of the GCD algorithm we discussed earlier.



Try and understand the flowchart

- After starting the program, we take two numbers as input from the user.
- The blue diamond-shaped box is what is called a decision box or a conditional (more on conditionals later). Here we find whether the first number is the greater one or not. If not, we exchange the numbers and compare them again.
- The yellow rectangle-shaped boxes are processing boxes. We find the remainder here and then again compare the remainder obtained with zero in the blue decision box.
- If the remainder does not come out to be zero, we repeat the process by changing the value of A and B as in the algorithm.
- After a certain number of repeated processes, the remainder becomes zero and the smaller number at that moment of time is our G.C.D.

Try tracing the flowchart for our example of 6 and 8. Now,



Pseudocode

Pseudocode is the third and the last step after writing an algorithm and drawing the flowchart which is suggested before we start actual coding. Pseudocode is an informal way of programming that does not require any strict programming language syntax or underlying technological considerations. It is used for creating an outline or a rough draft of the actual program. We will use the same G.C.D. example to maintain a flow.

Input a and b

```
if(a is greater than b)
    c = Remainder of a/b
else
    a=b and b=c
```

```
if(c is 0)
    Print b is the g.c.d.
else
    Go to step 1
```

You are advised to dry run([complete this link](#)) this pseudo code and try to visualize it. The pseudo-code will become more clear once we move further in the blog.