Chapter-3

ALGORITHM

Algorithm is something new? ✔ ✘

Muhammad al-Khwarizmi

You may have heard about a word **"Algorithm"** in your ***Maths*** or ***Computer*** class before.

But have you wondered

- **Why** it is called as "Algorithm".
- Also **what** is an "Algorithm"?

Why Algorithm- We often relate a word "Algorithm" or "Algo" with computer science and think of an algorithm as something new.

But the term actually originated some 900 years back. The word "Algorithm" comes from a **Persian Mathematical Genius** "Muhammad ibn Musa al-khwarizmi", popularly known as "al-khwarizmi".

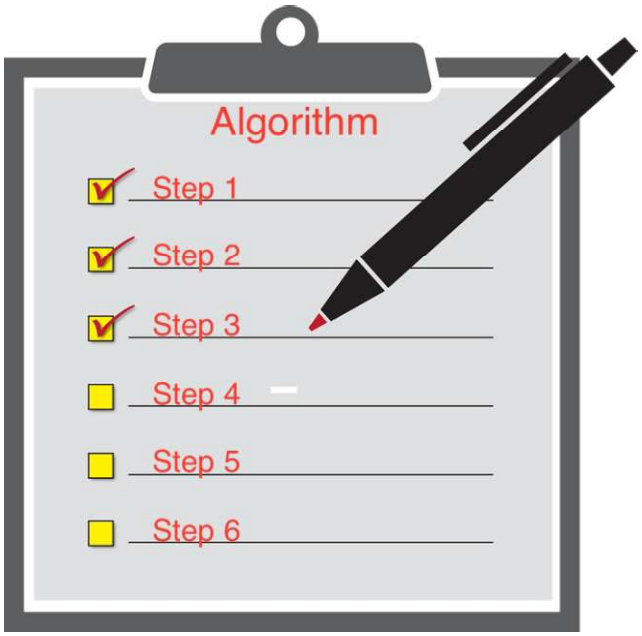
He is famous for introducing **"Hindu-Arabic numerals"** and concept of **"Algebra"** into European mathematics. His name when Latinised in the title of his book became

al-Khwarizmi => algoritmi => algorithm

What is an Algorithm: After you understand the origin of the word, the obvious question you might ask is about Algorithm itself. What is an Algorithm and how does it help us?

Algorithm = Set of steps to solve a particular problem

In a simple language, an algorithm is a detailed *step-by-step* instruction set or formula for solving a problem or completing a *task*.



You may think of algorithm only in the context of computer science or coding. But algorithm is everywhere. Knowingly or Unknowingly you are dealing with algorithms in your daily life.

- The process of baking a cake
- A recipe for making food
- Your morning routine
- Making a lemonade

At this point, you might be confused with the above examples.

You might think that "How making a lemonade" is an algorithm. Don't worry, let's see ***how*** !

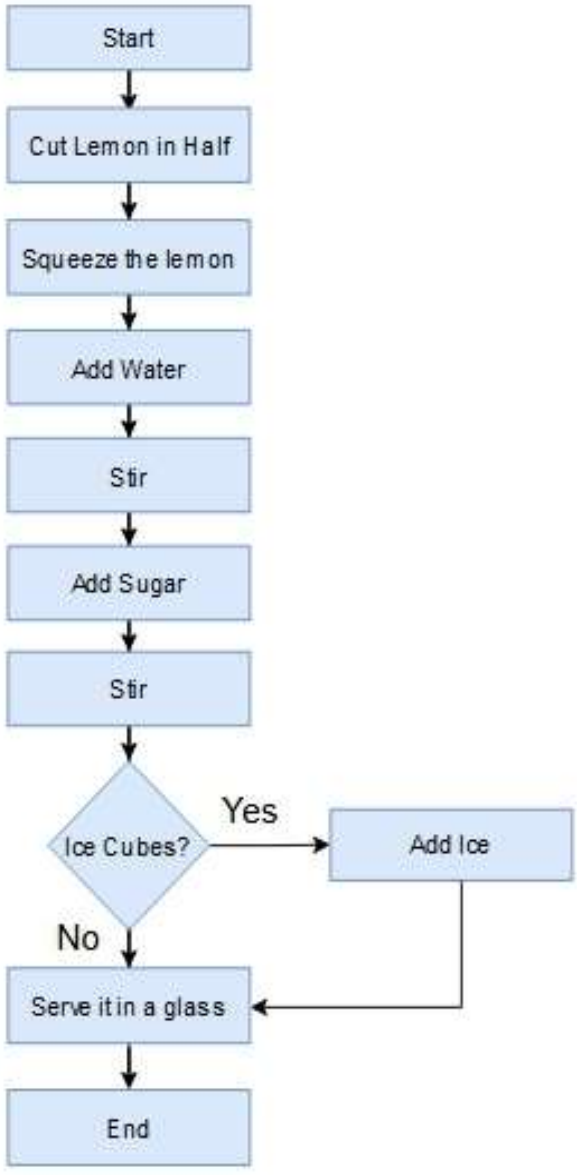
Making lemonade:

Let's write an algorithm (step-by-step instruction) that is involved in the process of making lemonade

In General English Language -

1. Cut Lemon in half
2. Squeeze Lemon in a container
3. Add water to the container
4. Stir the container
5. Add Sugar to the container
6. Stir the container again
7. Ice cubes?
    a. No - Go to step 8
    b. Yes
        i. Add ice and then go to step 8
8. Serve it in a Glass

In Flow Chart Form →



Flow Chart

Calm down! We will learn more about Flow Charts in Future

Write algorithm for following scenarios in General English Language	
Scenario	**Solution**
Opening the Window	
Tie your Shoelace	

Write algorithm for following scenarios in General English Language	
Scenario	**Solution**
Baking Cake	
Brushing Teeth	

Code Algorithm:

By now, you are familiar with Algorithms. You have also written them for daily general purpose use cases.

Let us now deep dive into Algorithms from Computer Science perspective.

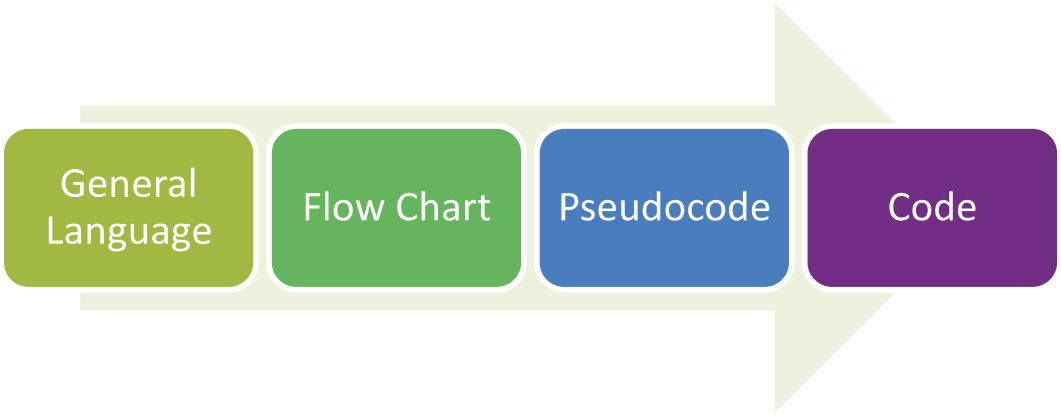
Recall Code or Coding – A computer program (set of instructions) written in a specific computer language which tells a computer what to do or solve a particular problem !

Why do we write code ?

To solve a particular use case scenario. These scenarios can be

- Chatting with a distant relative – WhatsApp
- To play online cricket – Cricket League
- To shop online – Myntra
- To book online railway ticket – IRCTC
- To book online bus ticket – redBus
- To transfer money – Paytm
- To book movie ticket – BookMyShow
- Many more ....

A good way to code a particular use case is to break it and follow below pattern



General Language:

This is a process where you write an algorithm in any general human understandable language. This is not necessarily to be in English language. This is where you should spend some time figuring out the process and all the different cases your algorithm may need to handle

Flow Chart:

This is a process very similar to General Language for writing an algorithm. The difference is in the representation. A flowchart is a graphical representation of an algorithm. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting".

Pseudocode:

In this you break your usecase into very specific steps, still in human understandable language but in informal way. In other words, Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations

Coding:

Finally, you write out each step of your algorithm in your coding language of choice. This is where you have to follow strict programming language syntax or underlying technology considerations

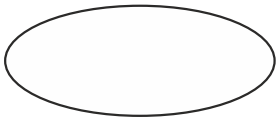
Let us now learn Flow Chart and Pseudocode Representation

Flowchart in detail

Flowchart **= Pictorial representation of an algorithm using** ***symbols***

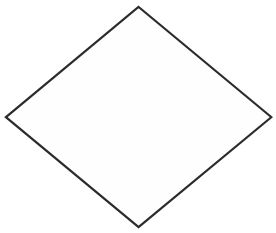
Symbols – There are few commonly defined symbols which are used in flowchart across world

Start or End: This symbol is used to describe the beginning or the ending of the workflow



Process: This symbol is used to describe the process, information, action, function or a step

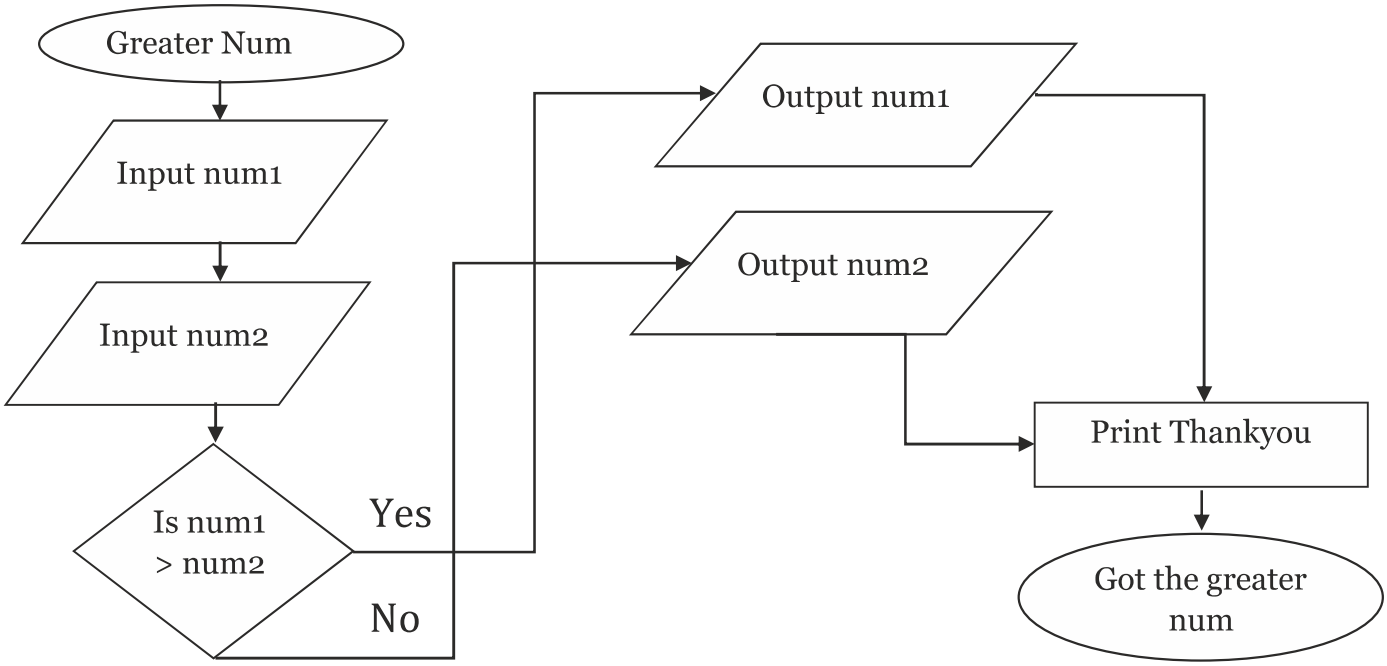
Decision: This symbol is used to describe a step where a decision to be made



Data: This symbol is used to represent the input or output of data

Flowline: This symbol is used to represent the order of workflow

Example: Flow chart representation to check which number among num1 & num2 is greater.



What would be the flow in accordance with this flowchart.

If ,

Num1 = 5

Num2 = 7

Pseudocode in detail

Pseudocode = is an informal way of programming or coding

In the earlier example,

We created a flowchart to explain the process of finding which number between num1 and num2 is greater.

Let us create an algorithm now using Pseudocode.

```
# Find the greater number between num1 and num2
Get Num1
Get Num2
If Num1 > Num2
    Print (Num1 is Greater)
Else
    Print (Num2 is Greater)
Print(We got the result)
```

Fun Facts:

You can choose **any or all the ways** to represent an algorithm before actually coding it. Algorithm helps to identify the design or defects at the earlier stage of development.

Examples

In this section, we will use our learning and create algorithms for various problems (use cases).

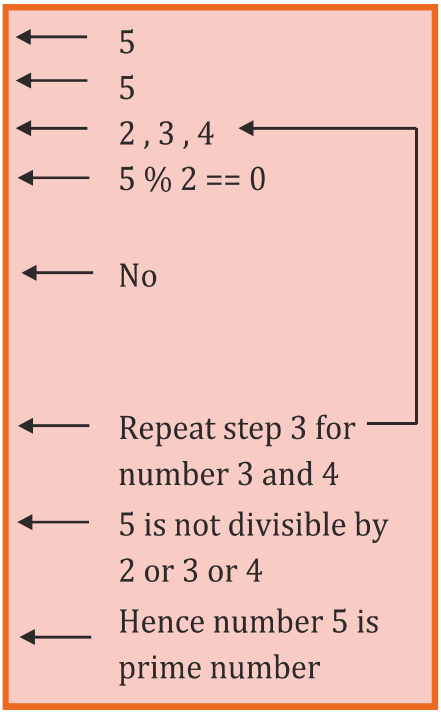
Use case 1: **Prime number** – In this use case, a user is asked to input a number between 1 and 100. Computer program should check whether the number is prime or not.

Prime number = a number that is divisible only by itself and 1

Using General Language (English)

1. Enter a number between 1 and 100
2. User enters a number
3. In the range of 2 and one less number entered by user
    - Check if number is fully divisible (i.e reminder == 0) by the number in range
        - If Yes,
            - Output that the number is not a prime
        - Else,
            - Increment the range and repeat step 3
4. If in step 3, the user number was never a divisible by any number between 2 and one less number entered by user.
    - Output that the number is prime

1. Enter a number between 1 and 100
2. User enters a number
3. In the range of 2 and one less number entered by user
    - Check if number is fully divisible (i.e reminder == 0) by the number in range
    - If Yes,
        - Output that the number is not a prime and do not run other steps
    - Else,
        - Increment the range and repeat step 3
4. If in step 3, the user number was never a divisible by any number between 2 and one less number entered by user.
    - Output that the number is prime



Using Pseudocode (Informal but human readable text)

1. Number = input("Enter a number between 1 and 100")
2. For counter =2 to (Number – 1)
    a. If Number % counter == 0
        i. Number is not prime
        ii. Exit
    b. Else
        i. Repeat step 2 with counter + 1
3. Number is prime

4

2 , 3

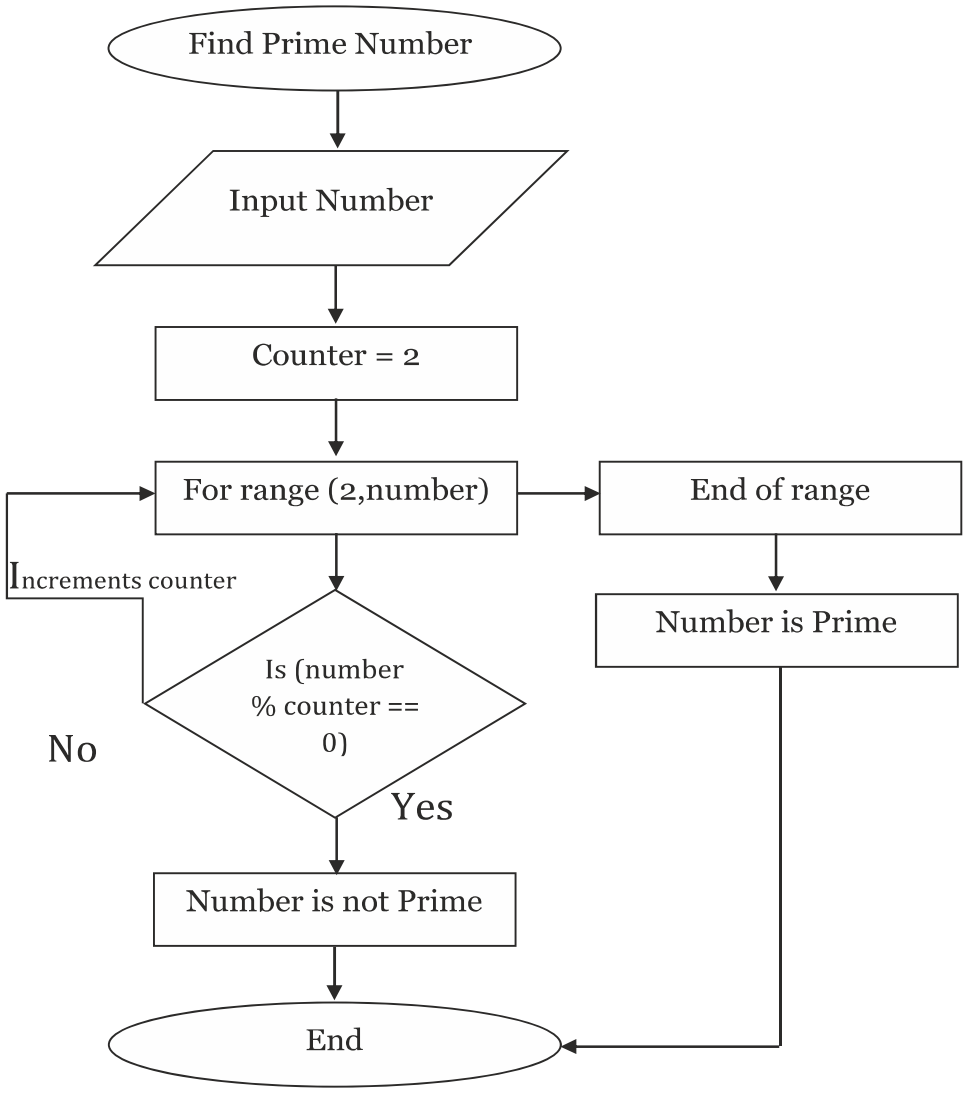
4 % 2 == 0

Yes, Number is not prime

Exit

*Repeat for number = 5 *

Flowchart – Now using flow chart



Solve this flowchart

Practice Use cases

In the previous use case example, we wrote algorithm using "General Language", "Pseudocode" and "Flowchart".

Similarly practice the below use cases

Use case 2: **Add three numbers** – In this use case, ask user to enter three numbers. After input, add all the numbers. Show the result at the end.

Use case 3: **Check if any vowel exist in an input name** – Vowels are "a , e , i , o , u". Ask user to enter first name. Check if any vowel exists in the name. If yes, then inform user. If no, then also inform user.

Use case 4: **Lamp is working or not** – Assume a scenario in which you are entering into your store room. You switch on the button of a lamp. Either the lamp will light or it will not. Write an algorithm.

Use case 5: **Leave application** – Write and submit a leave application to your class teacher. Reason for application can be "taking a break on Saturday to go shopping". Depending whether the application is approved or rejected, you will either be going on shopping or not. Write an algorithm covering all these conditions.

Use case 6: **Swapping Value** – In this use case, ask user to enter two numbers. Using arithmetic operations, you have to swap these numbers. For example if first number is A = 2 and second number is B = 5 then after swap the values should be A = 5 and B = 2.

Use case 7: **Choose clothing for party** – You have formal suit and ethnic wear options to choose for a party in the evening. If the weather is warm, then you would want to choose an ethnic wear. But if the weather is cold, then you would want to choose formal suit. Find the weather and choose the clothing accordingly. Write an algorithm covering these conditions.

Use case 8: **What's the value of T** – Fill the below table for **T.** With starting value of "**a=1** and **b=3**" after traversing the flow chart. What is the final value of "**a** and **b**" at which the algorithm stops.

a	b	T
1	3