

Introduction

16 August 2024 08:51

Lecture 1:

1. What is programming?
2. Algorithms (Introduction)
3. Flowchart (Practice & Assignments)
4. C++ (About & Why C++)
5. Basic Programs to start with.

Bonus:

Environment Setup & Executions

What is programming?

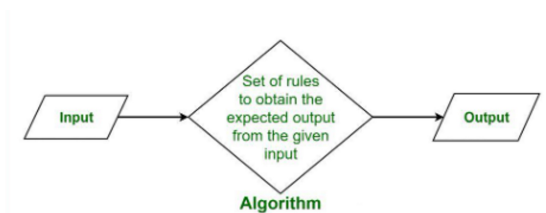
Programming is the process of writing instructions that a computer can understand and execute to perform specific tasks or solve problems. These instructions, called code, are written in programming languages, which are structured languages designed to communicate with computers.

- **Automation of Tasks:** Programming allows us to automate repetitive tasks, saving time and reducing errors.
- **Software Development:** Programming is used to create software applications, from simple apps to complex systems like operating systems, games, and web platforms.
- **Control Systems:** Programming is used to control machines, robots, and other devices, making it essential in industries like manufacturing and automation.

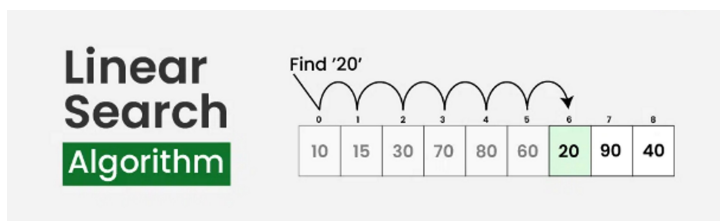
Algorithms (Introduction)

The word Algorithm means “a process or set of rules to be followed in calculations or other problem-solving operations”.

Therefore Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed in order to get the expected results.



Example: Linear Search



The algorithm for linear search can be broken down into the following steps:

- **Start:** Begin at the first element of the collection of elements.
- **Compare:** Compare the current element with the desired element.
- **Found:** If the current element is equal to the desired element, return true or index to the current element.
- **Move:** Otherwise, move to the next element in the collection.
- **Repeat:** Repeat steps 2-4 until we have reached the end of collection.
- **Not found:** If the end of the collection is reached without finding the desired element, return that the desired element is not in the array.

```
#include <bits/stdc++.h>
using namespace std;








int search(int arr[], int N, int x)
{
    for (int i = 0; i < N; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

// Driver code
int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int N = sizeof(arr) / sizeof(arr[0]);

    // Function call
    int result = search(arr, N, x);
    (result == -1)
        ? cout << "Element is not present in array"
        : cout << "Element is present at index " << result;

    return 0;
}
```

Flowchart (Practice & Assignments)

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

Guidelines for Developing Flowcharts

These are some points to keep in mind while developing a flowchart –

- Flowchart can have only one start and one stop symbol
- On-page connectors are referenced using numbers
- Off-page connectors are referenced using alphabets
- General flow of processes is top to bottom or left to right
- Arrows should not cross each other

Here is a flowchart to calculate the average of two numbers

Step 1: Start

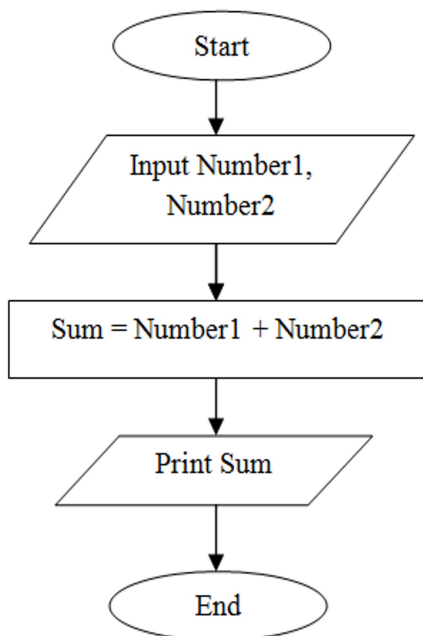
Step 2: Declare variables num1, num2 and sum.

Step 3: Read values for num1, num2.

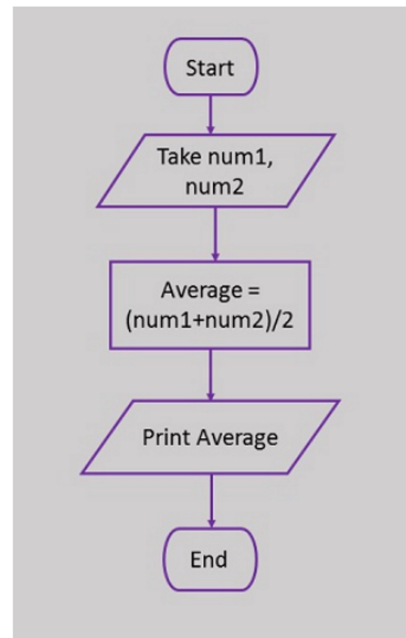
Step 4: Add num1 and num2 and assign the result to a variable sum.

Step 5: Display sum

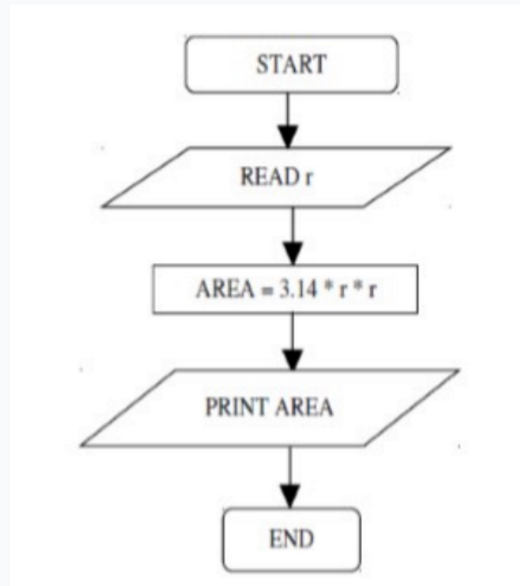
Step 6: Stop



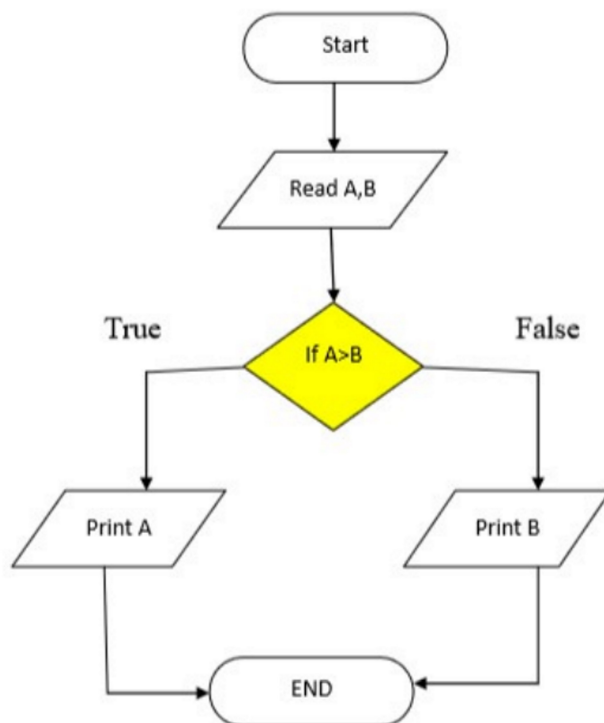
Here is a flowchart to calculate the average of two numbers



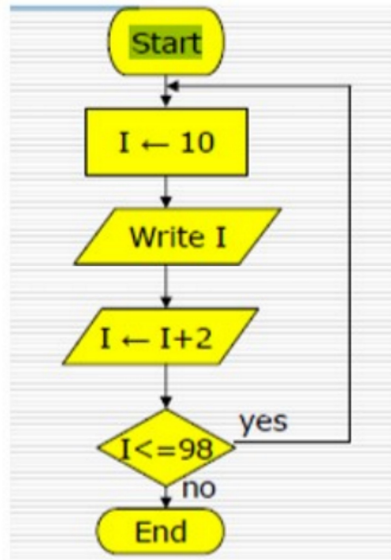
Problem 1 : Flowchart to calculate the area of circle.



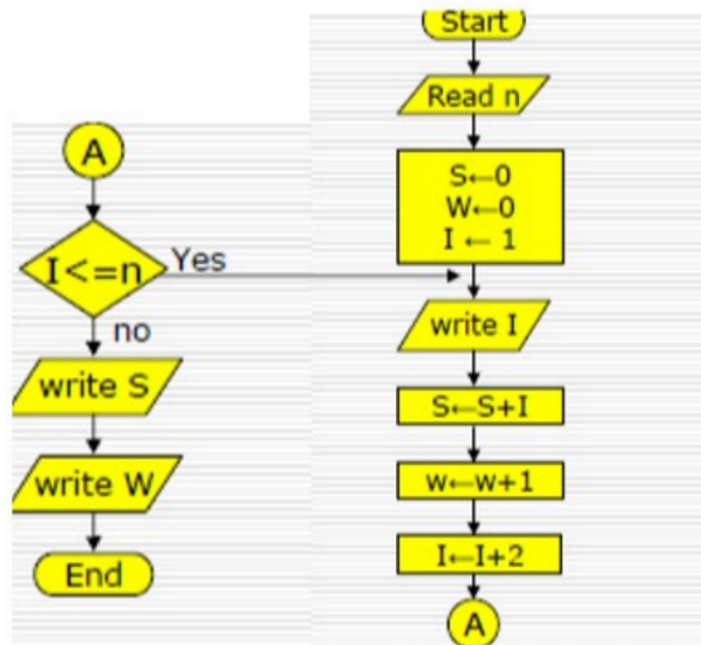
Problem 3: Flowchart to find the greatest from 2 numbers.



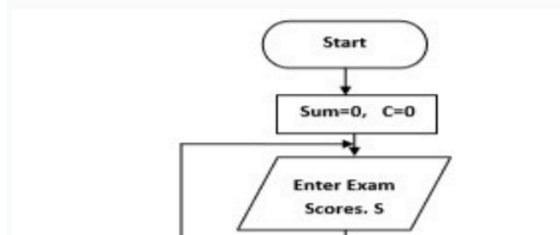
Problem 4: Flowchart to print the Even numbers between 9 and 100.



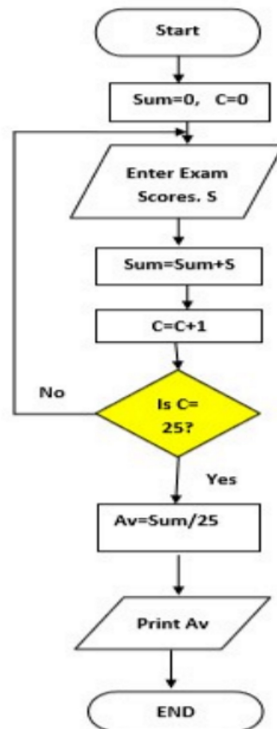
Problem 5: Flowchart for printing odd numbers less than a given number. It should also calculate their sum and count.



Problem 6 : Flowchart for the calculate the average from 25 exam scores.



Problem 6 : Flowchart for the calculate the average from 25 exam scores.



Assignments

1. Draw a flowchart to find the sum of first 100 natural numbers.
2. Draw a flow chart to find the largest of 3 numbers.
3. Draw a flowchart for check a given number is prime or not