**ALGORITHM**

An algorithm is a process or set of instruction to perform specific operation or some other problem-solving operations especially by a computer.

**CHARACTERISTICS OF AN ALGORITHM:**

1. **INPUT:** An algorithm has some input values. We can pass 0 or some input value to an algorithm.
2. OUTPUT: We will get 1 or more output at the end of an algorithm.
3. **UNAMBIGUITY:** An algorithm should be unambiguous which means that the instructions in an algorithm should be clear and simple.
4. **FINITENESS:** An algorithm should have finiteness. Here, finiteness means that the algorithm should contain a limited number of instructions, i.e. the instructions should be countable.
5. **EFFECTIVENESS:** An algorithm should be effective as each instruction in an algorithm affects the overall process.
6. **LANGUAGE INDEPENDENT:** An Algorithm must be language independent so that the instructions in an algorithm can be implemented in any programming language.

**DATAFLOW OF AN ALGORITHM**

1. **PROBLEM:** A problem can be a real-world problem or any instance from the real-world problem for which we need to create a program or set of instructions. The set of instructions is known as an algorithm.
2. **ALGORITHM:** An algorithm will be designed for a problem which is a step by step procedure.
3. **INPUT:** After designing an algorithm, the required and the desired inputs are provided to the algorithm.
4. **PROCESSING UNIT:** The input will be given to processing unit, and the processing unit will produce the desired output.
5. **OUTPUT:** The output is the outcome or the result of the program.

**WHY DO WE NEED ALGORITHMS?**

1. **SCALABILITY:** It helps us to understand the scalability. When we have a big real-world problem, we need to scale it down into small-small steps to easily analyse the problem.
2. **PERFORMANCE:** The real-world is not easily broken down into smaller steps. If the problem can be easily broken into smaller steps means that the problem is feasible.

**FACTORS OF AN ALGORITHM**

1. **MODULARITY:** If any problem is given and we can break that problem into small-small modules or small-small steps, which is a basic definition of an algorithm, it means that this feature has been perfectly designed for the algorithm.
2. **CORRECTNESS:** The correctness of an algorithm is defined as when the given inputs produce the desired output.
3. **MAINTAINABILITY:** It means that algorithm should be designed in a very simple structured way so that when we redefine the algorithm, no major change will be done in the algorithm.
4. **FUNCTIONALITY:** It considers various logical steps to solve the real-world problem.
5. **ROBOUSTNESS:** It means that how an algorithm can clearly define our problem.
6. **USER-FRIENDLY:** If the algorithm is not user-friendly, then the designer will not be able to explain it to the programmer.
7. **SIMPLICITY:** If the algorithm is simple then it is easy to understand,
8. **EXTENSIBILITY:** If any other algorithm designer or programmer wants to use your algorithm then it should be extensible.