

WAYS OF WRITING AN ALGORITHM

#algorithms #programmer



Yogeswaran Jan 5, 2019 · 2 min read

There are various ways of writing an algorithm. Today, I'm going to explain 3 ways of writing an **algorithm**.



4



4



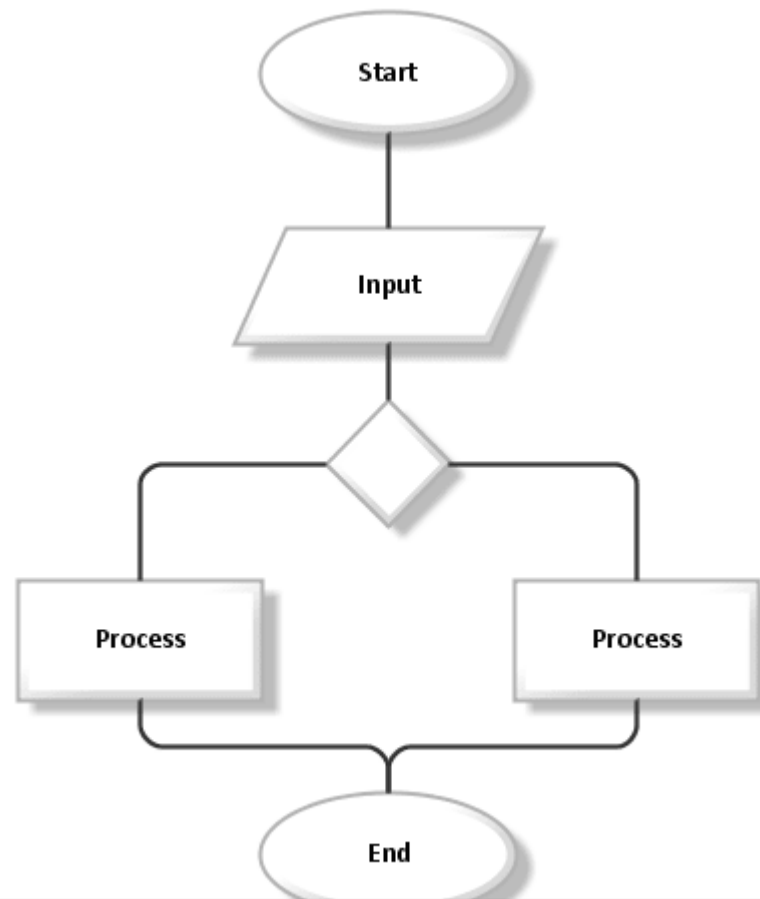
5



An algorithm can be written in many ways. It can be written in simple English but this method also has some demerits. Natural language can be ambiguous and therefore lack the characteristic of being definite. Each step of an algorithm should be clear and shouldn't have more than one meaning. English language-like algorithms are not considered good for most of the tasks.

2. Flowchart

Flowcharts pictorially depict a process. They are easy to understand and are commonly used in the case of simple problems.



4

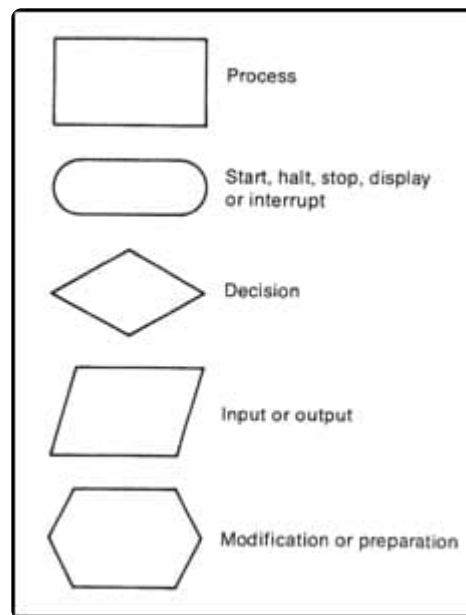


4



5





3. Pseudocode

The pseudocode has an advantage of being easily converted into any programming language. This way of writing algorithm is most acceptable and most widely used. In order to write a pseudocode, one must be familiar with the conventions of writing it.

1. Single line comments start with //

2. Multi-line comments occur between /* and */

3. Blocks are represented using brackets. Blocks can be used to represent compound statements or the procedures.

{



4



4



5



4. Statements are delimited by semicolon.

5. Assignment statements indicates that the result of evaluation of the expression will be stored in the variable.

`< variable > = < expression >`

6. The boolean expression 'x > y' returns true if x is greater than y, else returns false.

7. The boolean expression 'x < y' returns true if x is less than y, else returns false.

8. The boolean expression 'x <= y' returns true if x is less than or equal to y, else returns false.

9. The boolean expression 'x >= y' returns true if x is greater than or equal to y, else returns false.

10. The boolean expression 'x != y' returns true if x is not equal to y, else returns false.

11. The boolean expression 'x == y' returns true if x is equal to y, else returns false.

12. The boolean expression 'x AND y' returns true if both conditions are true, else returns false.

13. The boolean expression 'x OR y' returns true if any of the conditions is true, else returns false.

14. The boolean expression 'NOT y' returns true if the result of x evaluates to false, else returns false.

15. `if< condition >then< statement >`

16. This condition is an enhancement of the above 'if' statement. It can also handle the case where the condition isn't satisfied.

`if< condition >then< statement1 >else< statement2 >`

17. `switch case (C or C++)`



4



4



5



```
:< condition 1 >: < statement 1 >
```

```
.....
```

```
.....
```

```
.....
```

```
:< condition n >: < statement n >
```

```
:default: < statement n+1 >
```

```
}
```

18. while loop

```
while< condition >do {
```

```
statements
```

```
}
```

19. do-while loop

```
repeat
```

```
statements
```

```
until< condition >
```

20. for loop

```
for variable = value1 to value2 {
```

```
statements
```

```
}
```

21. input instruction

```
Read
```



4



4



5



23. The name of the algorithm is < name > and the arguments are stored in the < parameter list >

Algorithm< name > (< parameter list >)

Note: Number 6 to Number 11 use relational operator, Number 12 to 14 uses logical operator, and Number 15 uses conditional operator.

Join [My Telegram Community](#) so you won't miss out any articles.

Posted on Jan 5, 2019 by:



Yogeswaran

@yogeswaran79

Data Scientist & AI

Follow



Discussion

Subscribe



Add to the discussion

[Code of Conduct](#) • [Report abuse](#)



4



4



5



Part 2: Sorting Algorithm

#algorithms #sorting #computerscience #python

Part 1: Sorting Algorithm

#algorithms #sorting #beginners #computerscience

Concept Every Programmer should know

#concept #programmer



4



4



5

