Computer Programming Fundamentals

Week No. 2: Algorithms

1) What is a Computer Program?
2) What does Programming means?
3) Algorithms
4) Pseudocode
5) Flowchart

Computer Program

A Program is a set of instructions compiled together in a file to perform some specific task by the CPU (Central Processing Unit). It is a series of binary numbers (0s and 1s) arranged in a sequence, which when given to the computer performs some task.

Computer is a dumb machine with expeditious computational speed. It can give quick results to many of the complex scientific calculations, but it can't perform a task on its own. A computer needs set of instructions to do some task. This set of instructions is contained in a computer program. Computer program is basically in binary language i.e. series of 0s and 1s. A large bunch of programs makes the computer functional without which the computer would be like a paralyzed machine.

You may think computer as an idiot person who does not know to cook. If you provide ingredients for cooking Pasta to that idiot person, you cannot expect a delicious dish. However, if you provide ingredients along with the full step-by-step recipe of cooking Pasta then you may expect a real Pasta from that idiot person.

Programming

Programming is telling the computer what to do. Programming is the process of writing an algorithm into a sequence of computer instructions. Or you can simply say it is the process of writing computer programs. We generally transform the solution of a specific problem into computer language. It is the only way through which we can create our own programs and can execute them on a computer. Programming requires skill, logical thinking and lots of experience.

Algorithms

A sequence of step by step instructions to solve a problem. It was created by the mathematician Mohammed Ibn-Musa Al-Khawarizmi. It is used to plan about anything in your life.

Example 1: Drink your Coffee

1. Take a coffee pad.
2. Put it in the coffee machine.
3. Check if the coffee machine is turned on. __If not, turn the machine on.
4. Check if the water filter is full enough. — If not, add water.
5. Put a coffee mug under the coffee dispenser.
6. Press the coffee button.
7. The coffee is being served. — Wait for the machine to indicate when the coffee is ready.
8. If the coffee is ready — Take the coffee mug out of the coffee machine.
9. Add milk and sugar. — If you wish.

Congratulations! Your "Drink Your Coffee" algorithm is ready! Let´s make a second algorithm.

Example 2: Wash Your Hands

1. Open the water tap.
2. Put soap on your hands.
3. Clean your hands with water.
4. Shut down the water tap.
5. Dry your hands.

Computer algorithm is a set of instructions that the computer must follow to perform a specific task. Computers are not smart enough to predict what to do. Therefore, the computer algorithm has to be fully detailed in order to get the desired result.

More than one algorithm can be set to solve the same problem. However, in writing an algorithm we seek, in addition to finding a solution, to find a better performance. For example, in order to find a job application file in stock of files that are sorted alphabetically, we can try many algorithms.

For example, if we are looking for "Mohammed Ali Saeed" file. One algorithm can be by start flipping through the files one by one till we find the desired file.

Another algorithm, will can jump to the middle of the files and check the name. If the name alphabetically before "Mohammed" then we will ignore the first part. If it is after we will ignore the other half. After that, we will divide the application files in the middle again and check the name to ignore another half of the problem. The second algorithm for sure more efficient and faster to find the desired file in less steps than the first one. If we have 1024 file, we will skip 512 files just in one step by jumping to the middle and ignoring one part. In the second step we will have only 256 files and so on.

Algorithms can be represented using **Pseudocode** or **Flowchart** in order to make it easier to be read, understood and converted to a program.

Pseudocode

Pseudocode is a plain language description of the steps in an algorithm or another system. Pseudocode is intended for human reading rather than machine reading. The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code, and that it is an efficient and environment-independent description of the key principles of an algorithm. It is commonly used in textbooks and scientific publications to document algorithms and in planning of software.

Some of the pseudocode vocabulary, that we are going to use, are

- **Start** (in the beginning of each pseudocode)
- **End** (at the end of each pseudocode)
- **Let** (to assign a value to a variable. Assignment operation may be indirect value but as a mathematical or logical expression)
- **Get** (to read a value for a variable from the user or a file, mostly used when no direct value is specified)
- **Print** (to output a value on a screen or a printer)
- **If, Then, Else** (to control the algorithm flows to work only on the specified situations)
- **Repeat, Until** (to repeat certain instructions multiple times)

Example 1:

Write a pseudocode to find and print the summation of 3 and 4.

Solution:

1. Start
2. Let x=3
3. Let y=4
4. Let z = x + y
5. Print z
6. End

Example 2:

Write a pseudocode to find and print the summation of any two numbers.

Solution:

1. Start
2. Get x
3. Get y
4. Let z = x + y
5. Print z
6. End

Note1: the mathematical signs that are used in pseudocode or programming in general are:

- For summation + sign
- For subtraction - sign
- For multiplication * sign
- For division / sign

Example 3:

Write a pseudocode to find and print the multiplication of two numbers.

Solution:

1. Start
2. Get x
3. Get y
4. Let z = x*y
5. Print z
6. End

Example 4:

Write a pseudocode to find and print the average of 3 numbers.

Solution:

1. Start
2. Get A
3. Get B
4. Get C
5. Let SUM = A+B+C
6. Let AVG = SUM/3
7. Print AVG
8. End

Example 5:

Write a pseudocode and to find and print the volume of any cube.

Solution:

1. Start
2. Get X
3. Let V = X*X*X
4. Print V
5. End

Note2: to compare two numbers in pseudocode or in most programming languages, we use the following signs:

Signs / Characters	The Purpose
<	Smaller than
>	Greater than
<=	Smaller than or equal
>=	Greater than or equal
==	Equal to
!=	Not Equal to

Example 6:

Write a pseudocode to find and print the greater number between two inputs.

Solution:

1. Start
2. Get X
3. Get Y
4. If X>Y Then
   Print X
   Else
   Print Y
5. End

Example 7:

Write a pseudocode to find and print the smaller number between two inputs.

Solution:

1. Start
2. Get X
3. Get Y
4. If X<Y Then
   Print X
   Else
   Print Y
5. End

Example 8:

Write a pseudocode to print the word " Hello" 10 times

Solution:

1. Start
2. Print "Hello"
3. Print "Hello"
4. Print "Hello"
5. Print "Hello"
6. Print "Hello"
7. Print "Hello"
8. Print "Hello"
9. Print "Hello"
10. Print "Hello"
11. Print "Hello"
12. End

Note3: assignment operation (=) in programming is a lot different than equal sign (=) in mathematical equations. In mathematical equations, the equal sign means the left part is equal to the right part. However, in programming, the assignment operation means that the result of the right part will be stored on the left part. Moreover, left part consists of a single variable and cannot contain any mathematical or logical operations. All the operations should be on the right part only. Therefore, in programming algorithms we can see a statement like the following:

X=X+1

This statement means that the value of the right part will be calculated and stored on the variable on the left as new value. If we assume that X=3 then the result of this operation will be calculating 3+1 which is 4 and then update the X value to be 4.

One important use of the above statement is called "Counter Variable", which is a variable that is used to keep track of anything that must be counted, specially on repeating some actions multiple times. Another use is on solving cumulative arithmetic operations like calculating the factorial of an integer number.

Example 9:

Write a pseudocode to print the word " Hello" 100 times

Solution:

1. Let X=0
2. Repeat
   Print "Hello"
   Let X=X+1
   Until X==100
3. End

Note4: in example 8 and 9 we can notice that any text should surrounded with (") double quotations, like the word "Hello".

Flowchart

A flowchart represents an algorithm using a diagram and are commonly used in programming to find the steps to write a program. Flowchart diagram uses different shapes to represent different kinds of operations as follows:

Start, End

Assignment Operations (Let)

Input/Output (Get, Print)

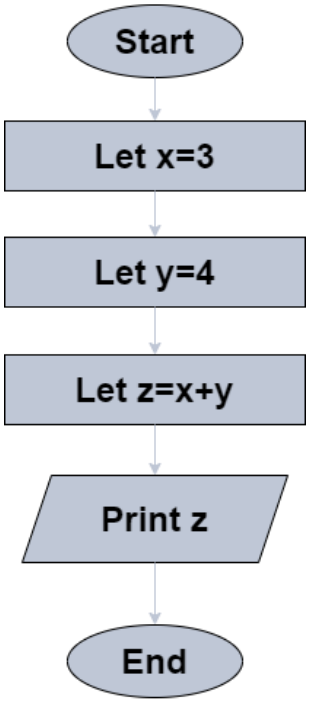
Conditions (If, Until)

Next Step

Example 1:

Draw a flowchart to find and print the summation of 3 and 4.

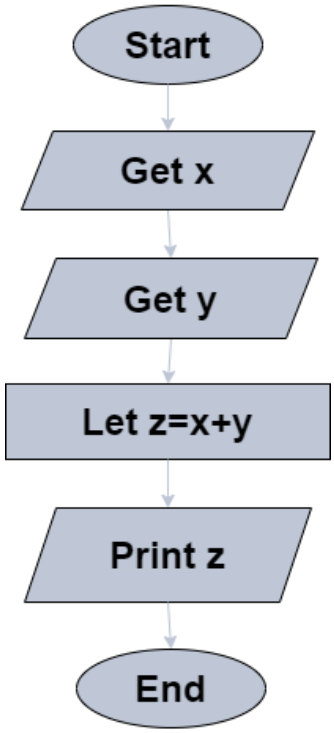
Solution:



Example 2:

Draw a flowchart to find and print the summation of any two numbers.

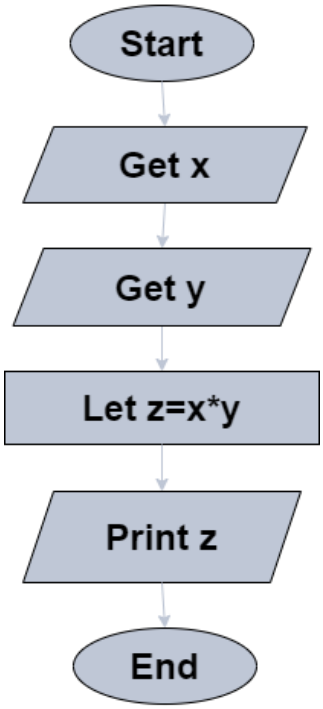
Solution:



Example 3:

Draw a flowchart to find and print the multiplication of two numbers.

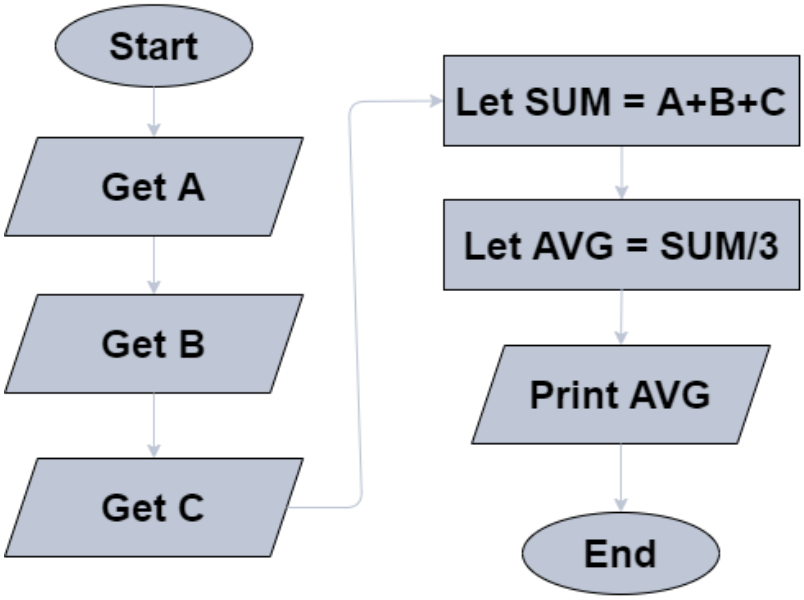
Solution:



Example 4:

Draw a flowchart to find and print the average of 3 numbers.

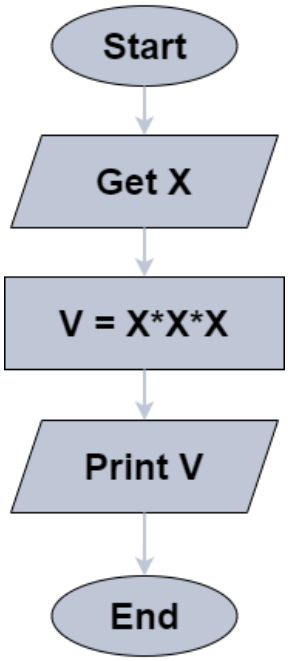
Solution:



Example 5:

Draw a flowchart to find and print the volume of any cube.

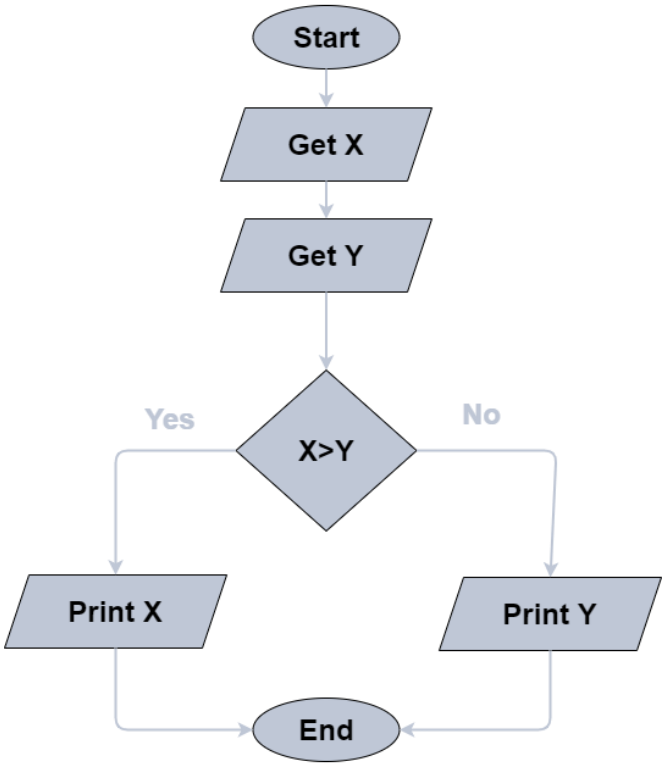
Solution:



Example 6:

Draw a flowchart to find and print the greater number between two inputs.

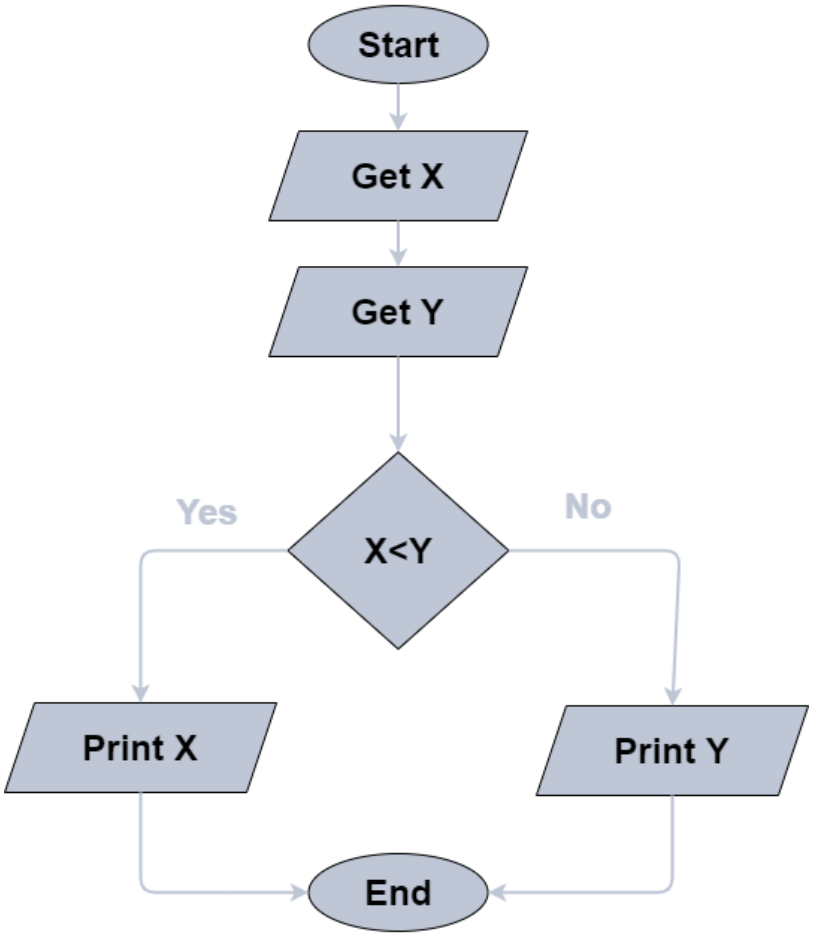
Solution:



Example 7:

Draw a flowchart to find and print the smaller number between two inputs.

Solution:



Example 8:

Write a pseudocode to print the word " Hello" 10 times

Solution: