



ALGORITHM





Algorithm

- ▶ An algorithm is a complete step-by-step procedure for solving a problem or accomplishing a task.



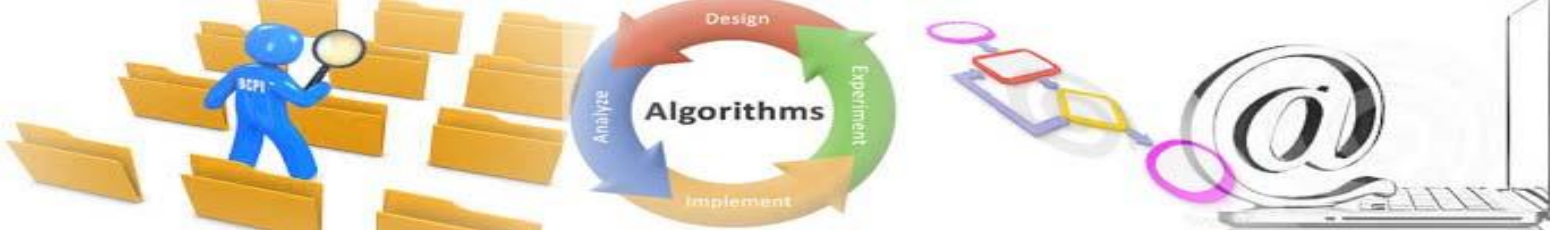
An Algorithm must be...

- Be Precise
- Be Logical
- Be Effective
- Terminate Eventually




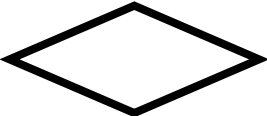


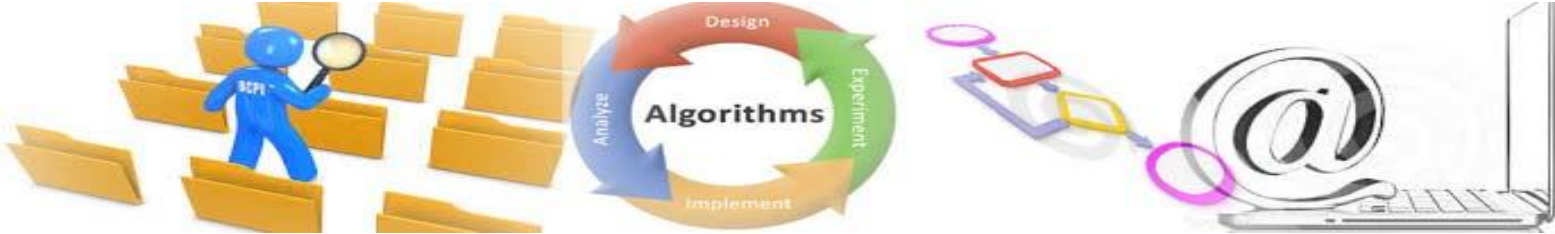
Flow Charts

- It is a step by step Diagrammatic representation of the program
- Each type of task is represented by a symbol






Standard Flow Charts Symbols

Diagram	Shape	Representation
	Oval	Start / End of a Program
	Parallelogram	Input / Output of Data
	Rectangle	Processing Operation
	Rhombus	Decision Box



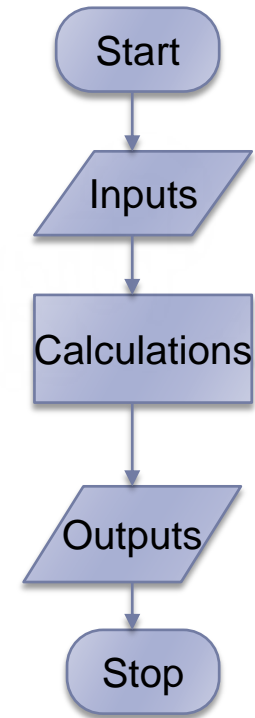
ගැලීම් සටහන් සඳහා යොදාගන්නා සංකේත Standard Flow Charts Symbols

Diagram සංකේතය	Shape හැඩය	Representation අදහස
	වෘත්තය / Circle	ගැලීම් සටහන දීර්ඝ වන විට එහි කොටස් 2ක් සම්බන්ධ වන පෙන්වීම. Connection
	ඊතලය / Arrow	දත්ත ගලා යන දිශාව පෙන්වීම Direction of Flow
	-	උප කාර්යය / Sub Process



A Typical Algorithm

Most simple algorithms that you will develop will involve inputting some data, performing some calculation and finally displaying the output.





Simple Example

- ▶ Consider the process of getting up in the morning and going to school.
 - Wake Up
 - Get Out of Bed
 - Have A Shower
 - Dry Yourself
 - Get Dressed
 - Have Breakfast
 - Pack Bag/Lunch
 - Leave



The logic constructs

- ▶ It has been proven that three basic constructs for flow of control are sufficient to implement any "proper" algorithm.
 - **Sequence**
 - **Selection**
 - **Iteration / Loop**



SEQUENCE

- ▶ SEQUENCE is a linear progression where one task is performed sequentially after another...
- ▶ solution steps must follow each other in a logical sequence
- ▶ Statements are executed in the same order as they are written.



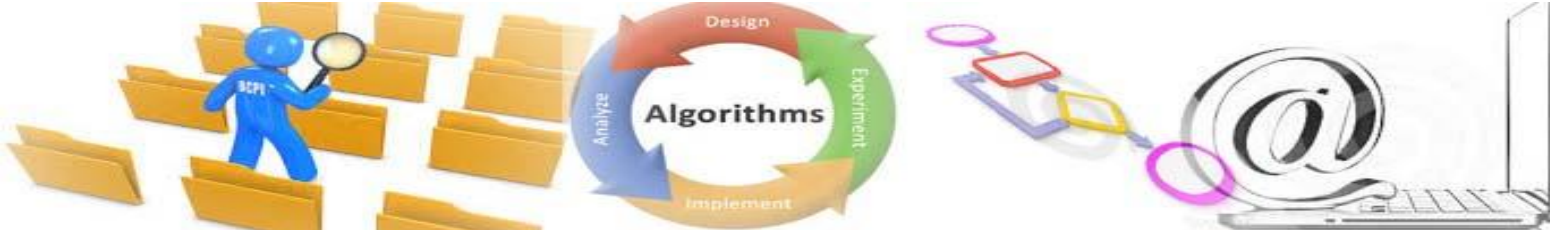
Simple Example

- ▶ Consider the process of getting up in the morning and going to school.
 - Wake Up
 - Get Out of Bed
 - Have A Shower
 - Dry Yourself
 - Get Dressed
 - Have Breakfast
 - Pack Bag/Lunch
 - Leave

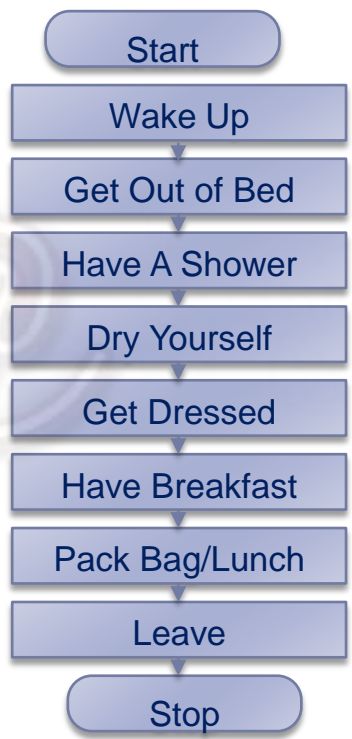
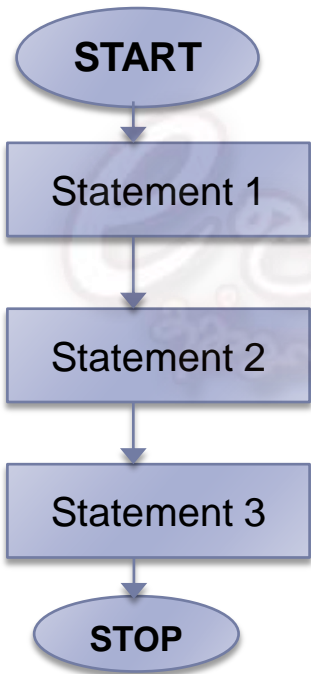


Note

- Certain events must occur in a particular order . Example, we should get out of bed before a shower.
- Some events must occur prior to another event(s). Example, there is no drying yourself prior to a shower
- Some other events may occur in any order and do not affect the overall solution. Example, we can pack bag before or after having breakfast



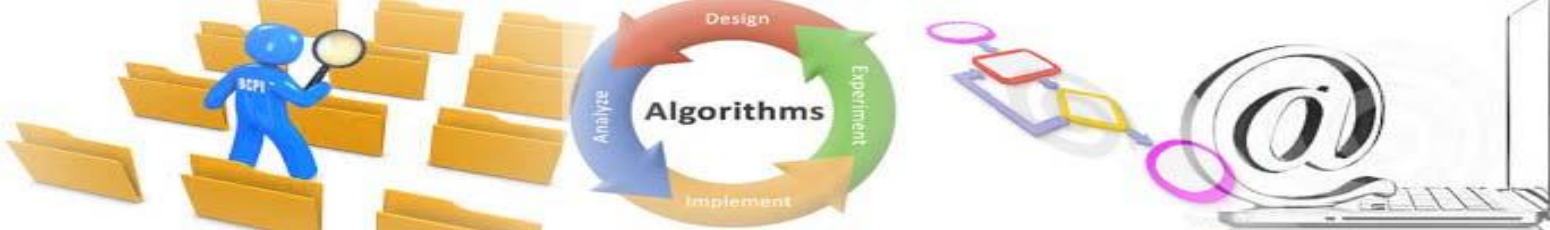
SEQUENCE



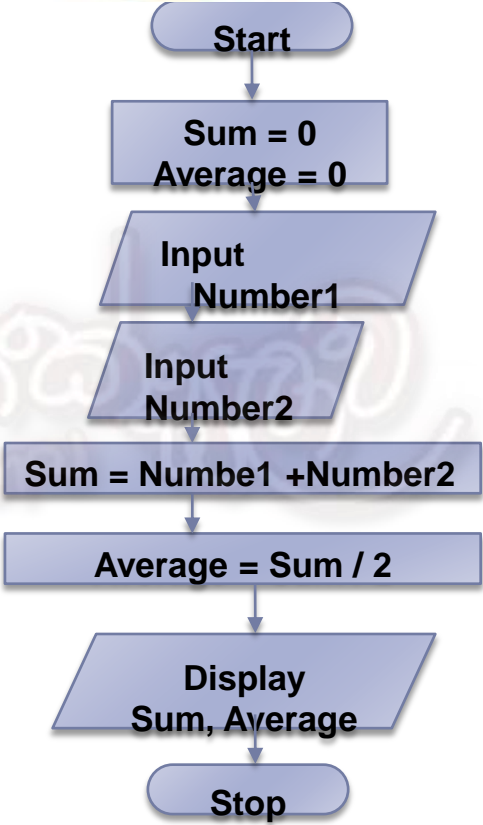


Example -1

- ▶ Draw a flow chart to find the sum and average of two numbers.



Solution





Exercises

- ▶ Input length and width of a rectangle and calculate the area and the perimeter.
- ▶ Input area of the square and calculate the length.
- ▶ Input the perimeter of a circle and calculate the area.





Note

- Recalling our example algorithm of going to School, everyone had to follow the same steps.
- But what if you weren't dirty or have had a shower the night before?
- ...you still had to have a shower, dry yourself, etc.



Note

- The same could be said about breakfast - what if you weren't hungry?
- Obviously in the real world we make choices.



Note

- In solving a problem we can make different choices depending on certain conditions - i.e. we make decisions
- The same can be done in programming as a part of decision making.



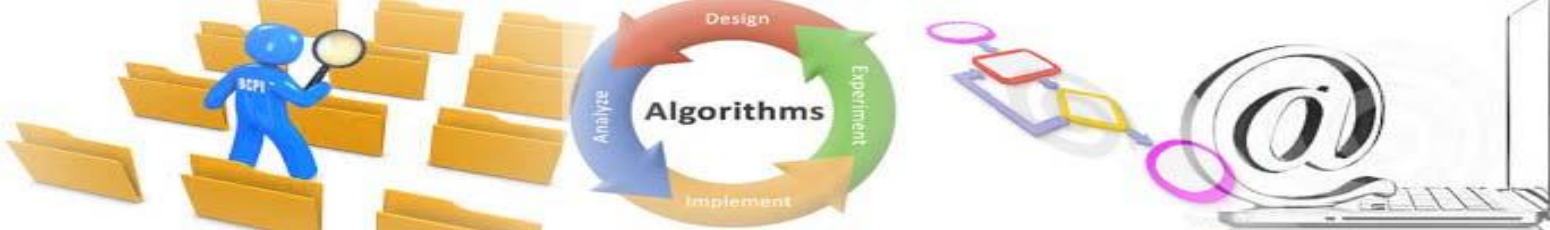
Note

- ▶ Note that even though selection is a separate construct to sequence, the two are combined in the overall solution, and remember that one doesn't replace the other.

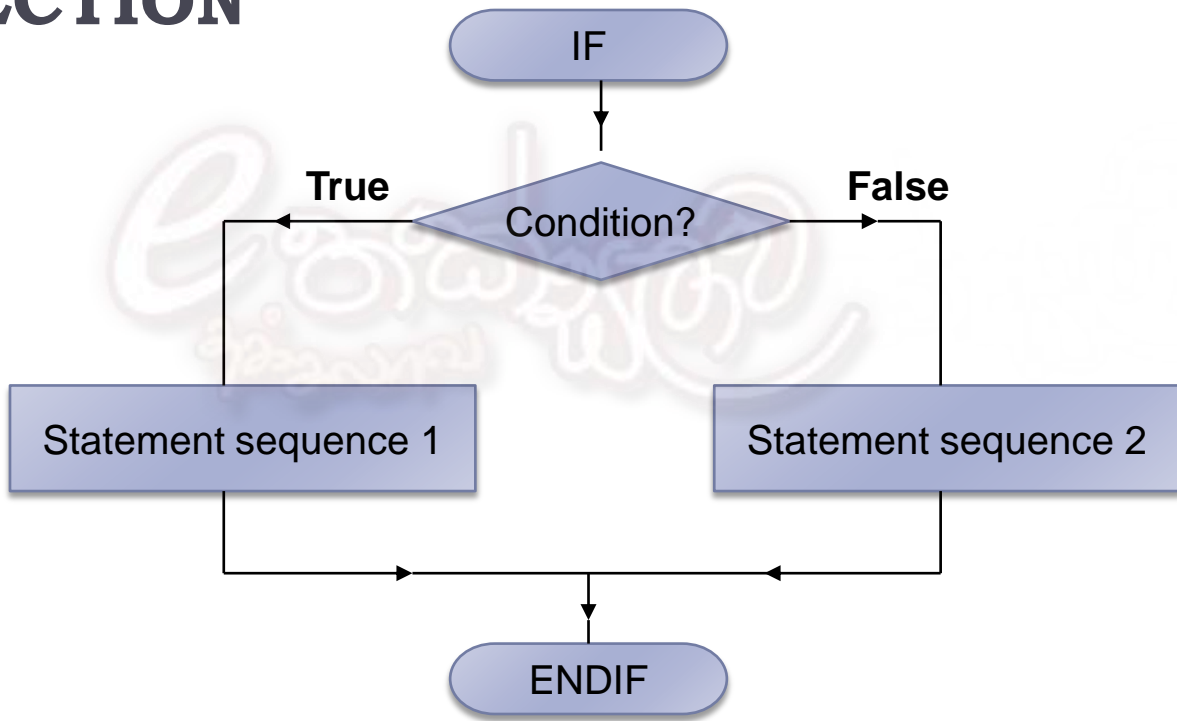


SELECTION

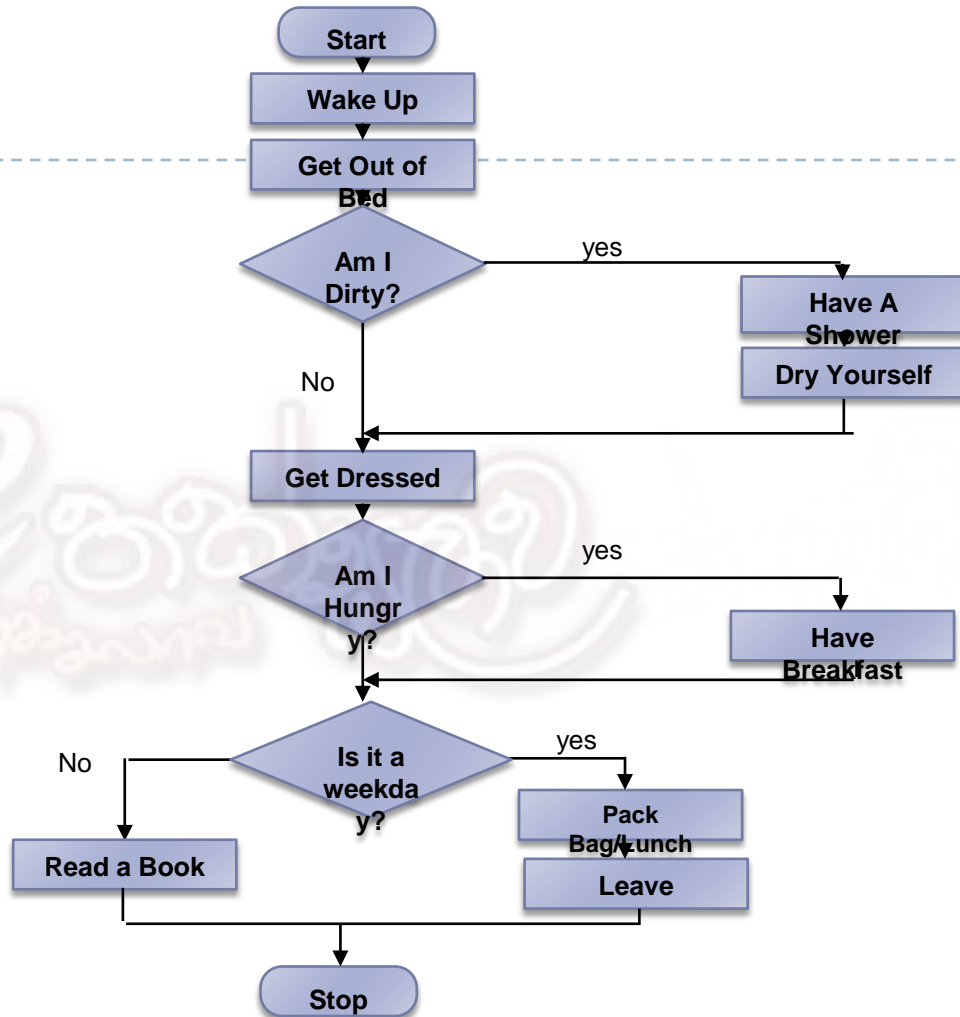
- ▶ There may be alternative steps that could be taken subject to a particular condition
- ▶ **IF-THEN-ELSE** is a decision (selection) in which a choice is made between two alternative courses of action



SELECTION



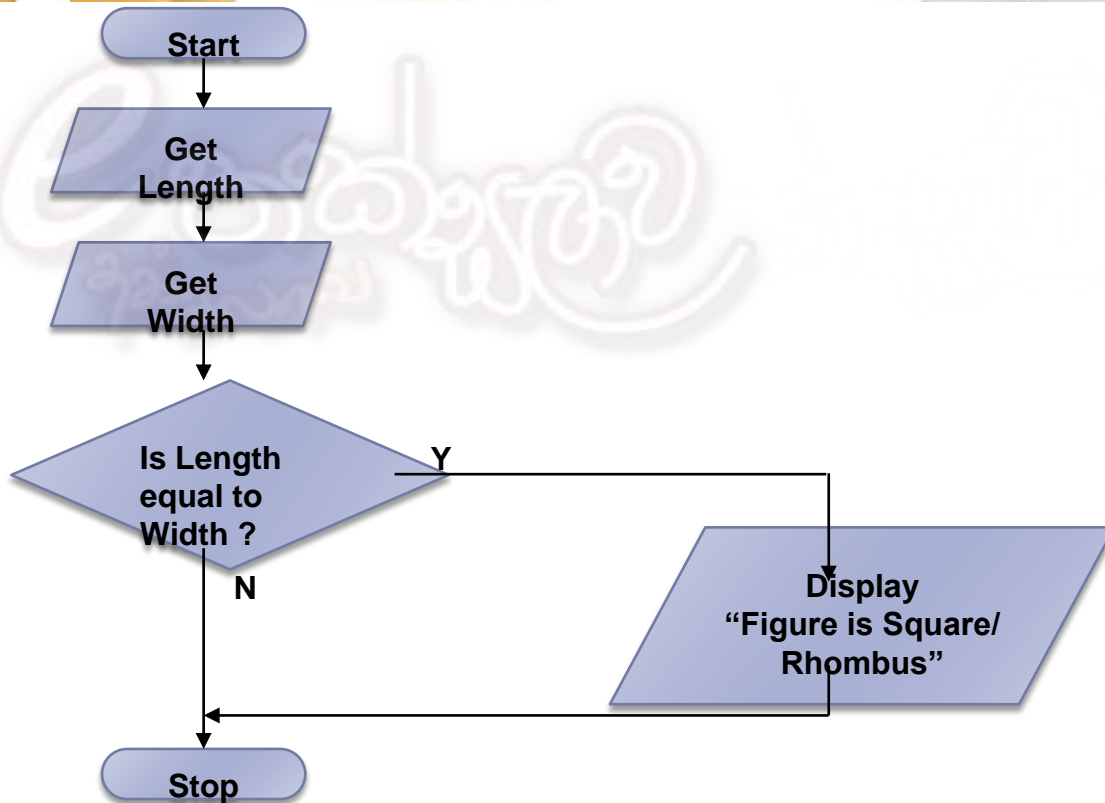
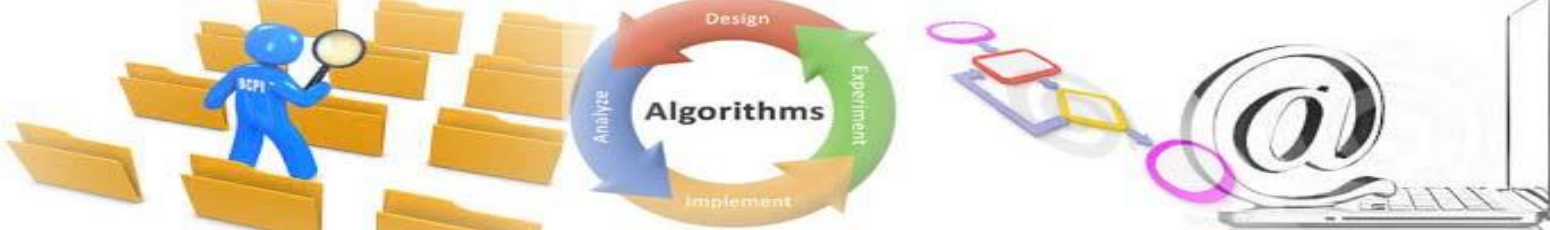
Example





Example - 2

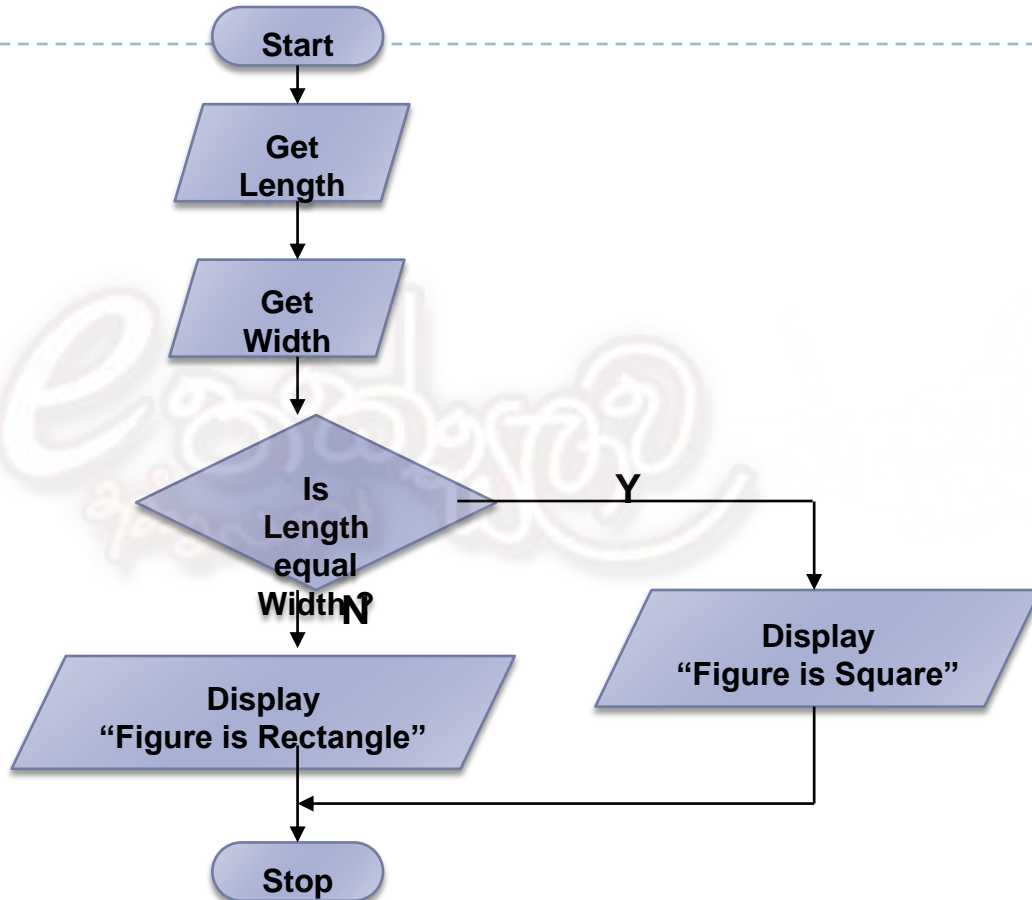
- ▶ Draw a flowchart to input the length and width of a quadrilateral and state whether it is a square/ rhombus





Example - 3

- ▶ Draw a flowchart to input the length and width of a quadrilateral and state whether it is a square or a rectangle.





Exercises

- ▶ Input the marks of a student and to print his/her grade.

Marks ≥ 75 - Distinction

$75 > \text{Marks} \geq 60$ - Merit

$60 > \text{Marks} \geq 40$ - Pass

$40 > \text{Marks}$ - Fail



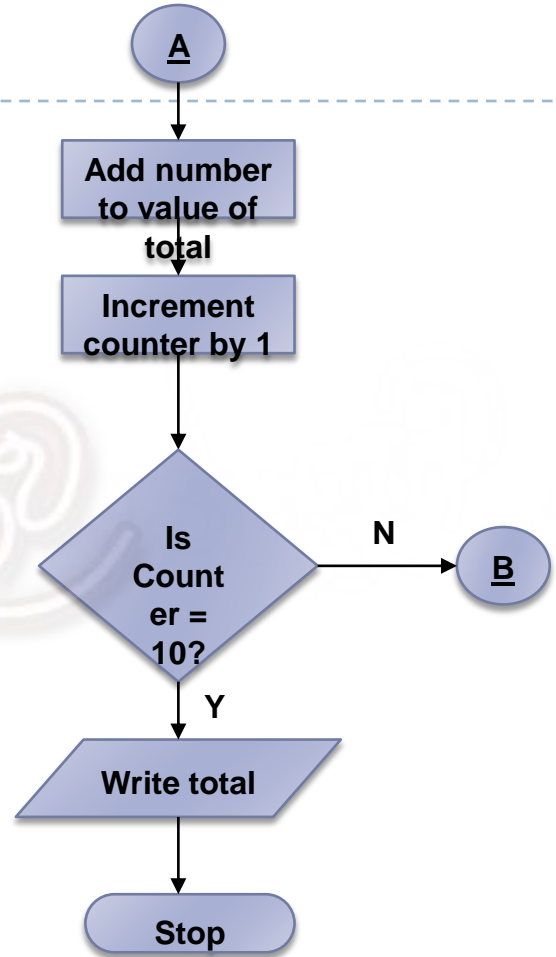
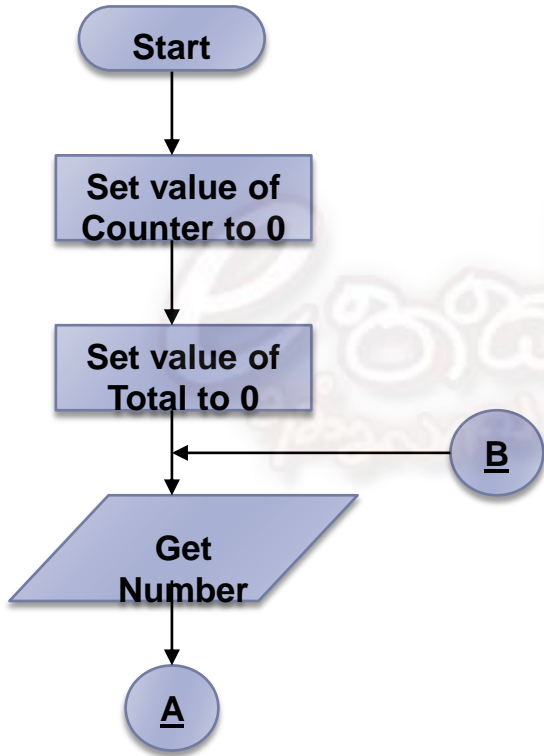
Exercises

- ▶ Inputs two integer values. If the first is less than the second, print the message “up”. If the second is less than the first, print the message “down”. If the numbers are equal, print the message “equal”.



Note

- When a flowchart is too long to fit on a page and is to be continued on another page a connector symbol is used.
- A small circle is used to represent a connector in a flowchart.
- An alphabet or a number is written inside the circle to identify the pairs of connectors to be joined.





ITERATION

- **ITERATION** - certain steps may need to be repeated while, or until, a certain condition is true
- We call it as a **Loop**



ITERATION

- There are three different types of **loops** in structured programming, all of which are available in all programming language.
- The first, and most common of these is the *while loop*.



ITERATION

- *Consider the example of getting out of bed*
- For the step --having a shower-- the following loop could be substituted;
while I am dirty
wash myself
test cleanliness
- The loop continues until the condition of cleanliness changes

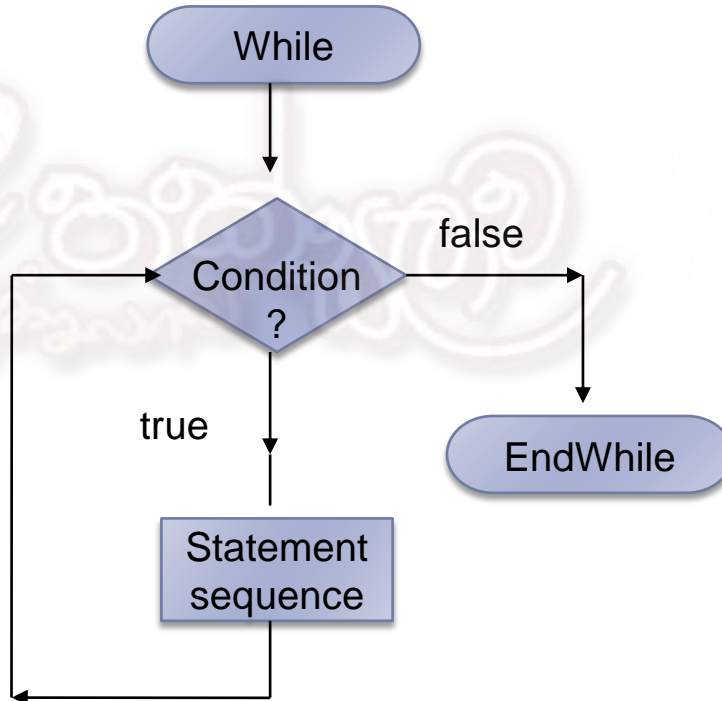


ITERATION- While Loop

- Section(s) of an algorithm are repeated over and over (obviously these loops must eventually terminate)
- This is achieved by a test of whether a condition is *true* or *false*
- In a while loop we continue to repeat something **while a condition is true** – we terminate the loop when it is **false**



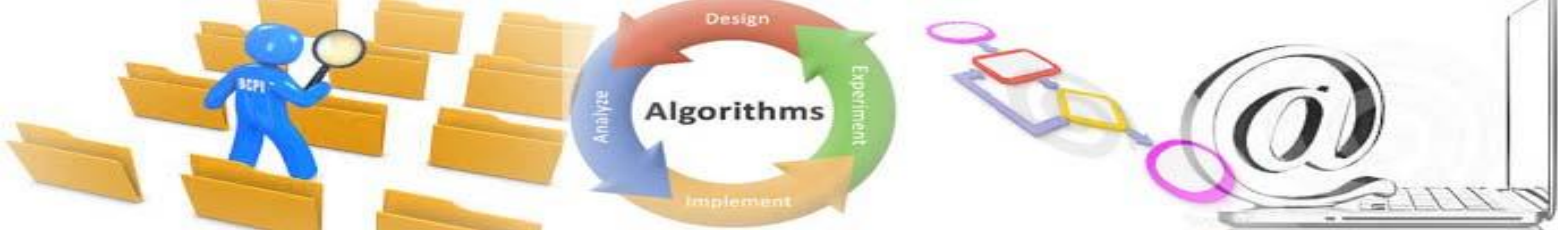
ITERATION- While Loop



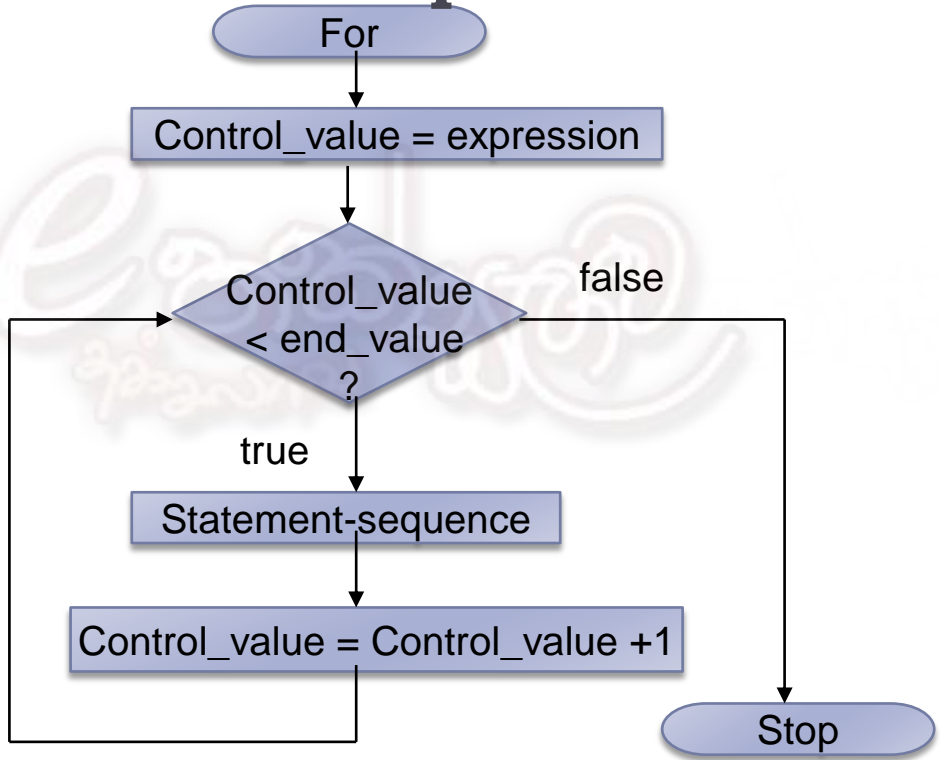


ITERATION- for Loop

- ▶ A loop that is repeated a specified number of times.



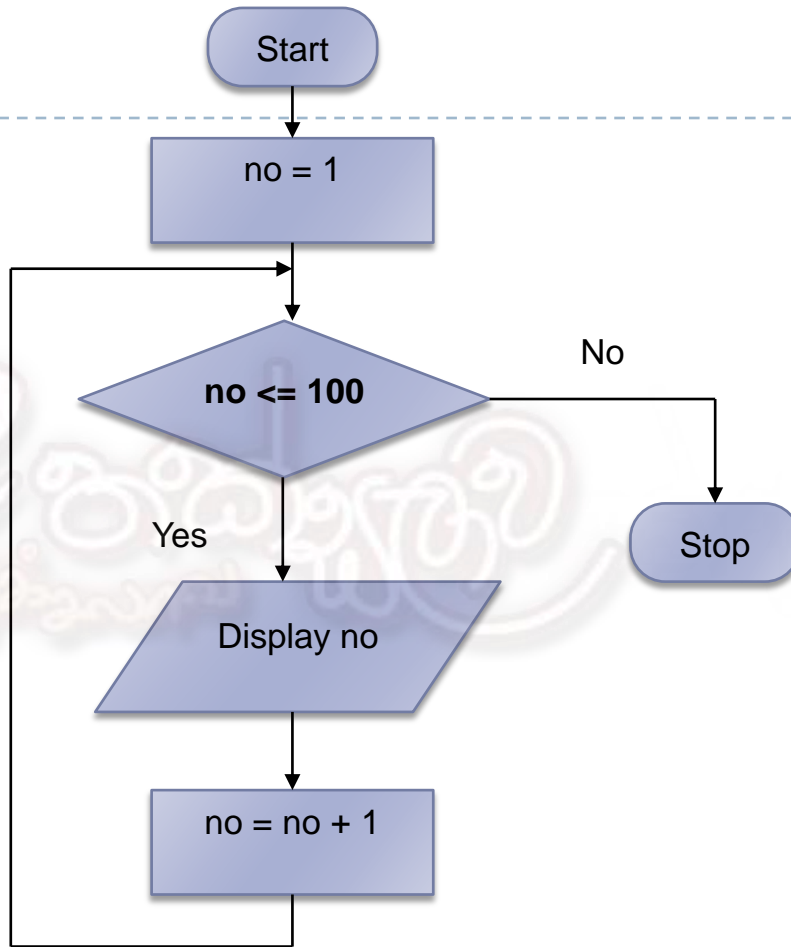
ITERATION- for Loop

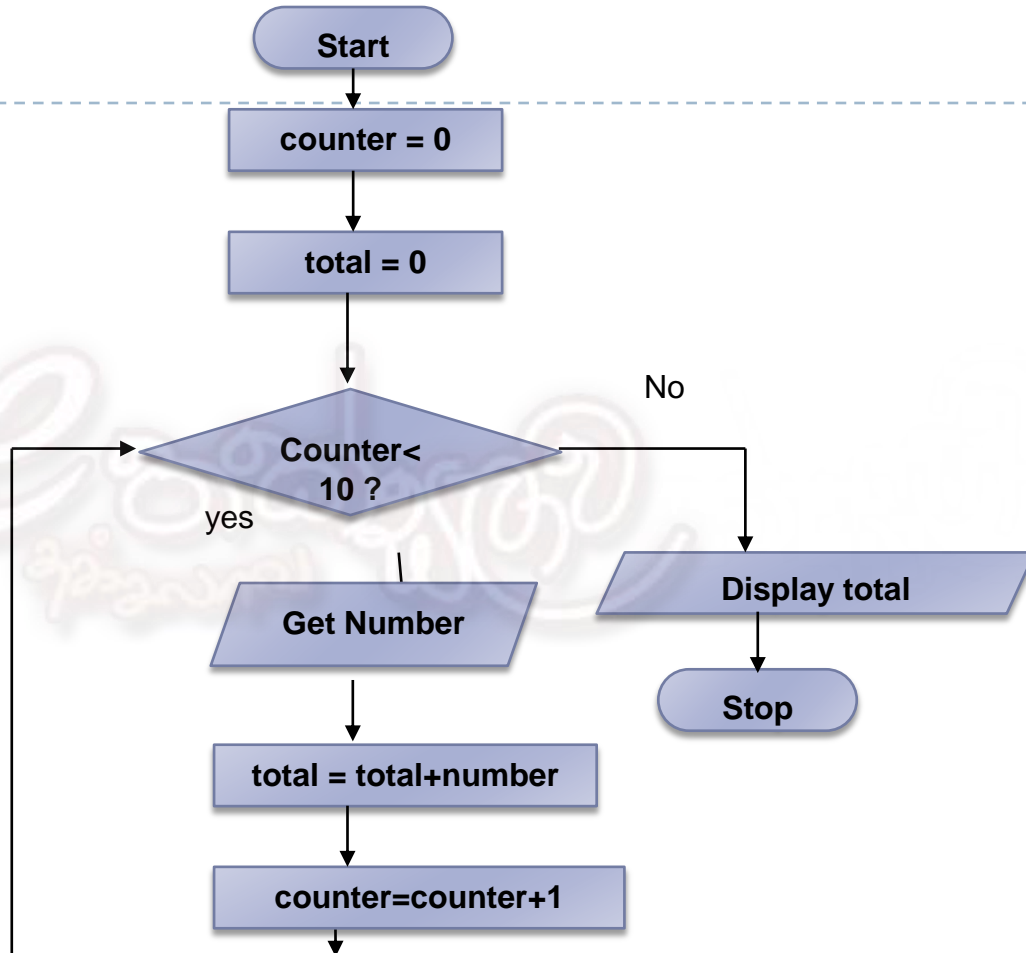




Exercise

- ▶ Draw a flowchart to display the numbers 1, 2, 3, 4, 5,, 100
- ▶ Write an algorithm to find the sum of 10 numbers.

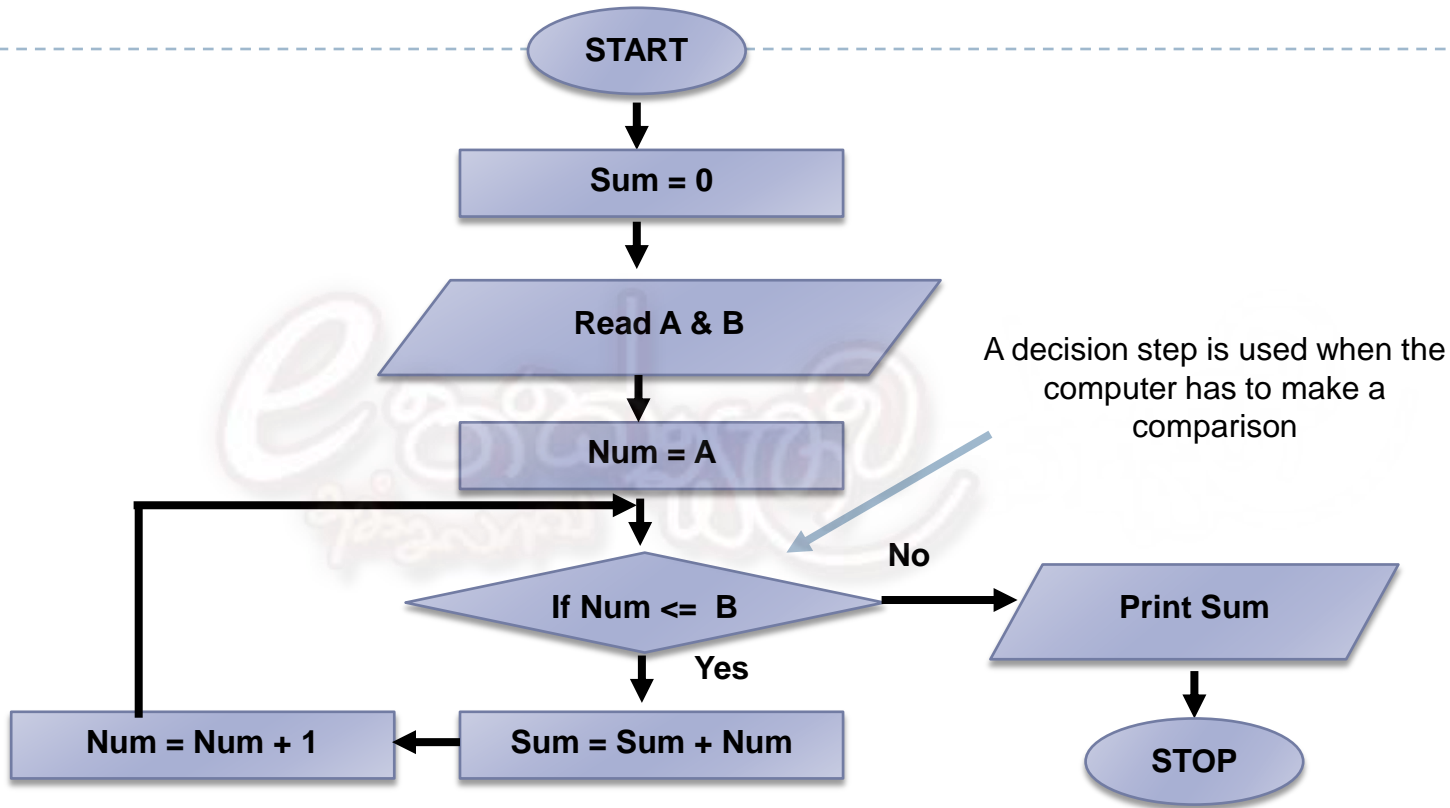






Exercise

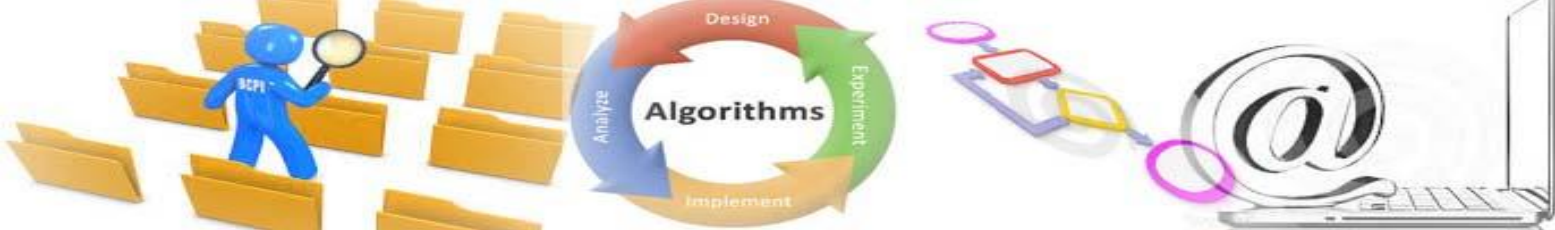
- ▶ Given 2 integer numbers (say a and b) where $a < b$, find the sum of numbers between a and b , including a and b





Exercises

1. Read in 10 numbers and compute the average, maximum and minimum values.
2. Input a 10 numbers from the keyboard and print the number if it is less than 5.



Exercises

3. සිසුන් 5 ක ගේ උස ප්‍රමාණ ඇතුළත් කර සාමාන්‍ය ලබාගැනීමට ගැලීම් සටහනක් අඳින්න. උසෙහි සාමාන්‍යය අඩි 6 ට වැඩි නම් “The students are very tall” යනුවෙන් පණිවුඩයක් දිස්වීමට ද සලස්වන්න.

Input the height of 5 students. Calculate and print the average height. If the average height is over 6 feet then print the message “The students are very tall”.

4. සංඛ්‍යා 10 ක් ඇතුළත් කර එම සංඛ්‍යා වලින් කොපමණ සංඛ්‍යාවක් ධන සංඛ්‍යා ද? කොපමණ සංඛ්‍යාවක් සෘණ සංඛ්‍යා ද? යන්න සෙවීමට ගැලීම් සටහනක් අඳින්න.

Input 10 numbers and count how many negative numbers and how many positive numbers are there.

Delete

End

End



THE END