Algorithms and Flowcharts

Problem Analysis

Problem analysis can be defined as studying a problem to arrive at a satisfactory solution. To solve a problem successfully, the first step is to understand the problem. Also the problem must be stated clearly and accurately without any confuse. A well-defined problem and clear description of input and output are important for an effective and efficient solution. Study the outputs to be generated so that input can be specified. Design the steps, which will produce the desired result after supplying the input.

If the problem is very complex, split the problem into multiple sub-problems. Solve each sub-problem and combine the solutions of all the sub-problems to at arrive at overall solution to a large problem. This is called divide and conquer technique.

Problem solving steps

- Understand the problem and plan its logic
- Construction of the List of Variables
- Develop an algorithm
- Refine the algorithm. Refining means making changes
- Program Development
- Testing the Program (solution)
- Validating the Program (solution)

Algorithm

An algorithm is defined as **sequence of steps to solve a problem (task)**. The steps must be finite, well defined and unambiguous. Writing algorithm requires some thinking. Algorithm can also be defined as a plan to solve a problem and represents its logic. Note that an algorithm is of no use if it does not help us arrive at the desired solution

Algorithm characteristics

1. **It should have finite number of steps**. No one can be expected to execute infinite number of steps.
2. The steps must be in order and simple
3. **Each step should be defined clearly stated i.e. without un-ambiguity**
4. Must include all required information
5. Should exhibit at least one output

For accomplishing a particular task, different algorithms can be written. Different algorithms can differ in their requirements of time and space. Programmer selects the best suited algorithm for the given task to be solved.

Algorithm for preparing two cups of tea

1. Add 1.5 cups of water to the vessel
2. Boil water
3. Add 2 tea spoons of tea leaves
4. Add half cup of milk
5. Add some sugar

Statement 5 is an example of an **ambiguous** (unclear) statement. This statement doesn't clearly state the amount of sugar to be added.

Algorithm characteristics

1. **It should have finite number of steps**. No one can be expected to execute infinite number of steps.
2. The steps must be in order and simple
3. **Each step should be defined clearly stated i.e. without un-ambiguity (without doubtfulness)**
4. Must include all required information
5. Should exhibit at least one output

Algorithm	Flowchart	Program
An algorithm is defined as **sequence of steps to solve a problem (task)**.	**A flowchart is pictorial (graphical) representation of an algorithm**.	Set of instructions. Instruction is a command to the computer to do some task.
Algorithm can also be defined as a plan to solve a problem and represents its logic.	A picture is worth of 1000 words. We can understand more from picture than words.	Implementation of Algorithm or flowchart

Different algorithms have different performance characteristics to solve the same problem. Some algorithms are fast. Some are slow. Some occupy more memory space. Some occupy less memory space. Some are complex and some algorithms are simple.

Logically algorithm, flowchart and program are the same.

Algorithm design

- Design an algorithm that is easy to understand code and debug. Debugging is the process finding and fixing errors in a program
- Design an algorithm that makes use of resource such as space (memory) and time efficiently

Flowchart

A flowchart is a pictorial (graphical) representation of an algorithm. A flowchart is drawn using different kinds of symbols. A symbol is used for a specific purpose. Each symbol has name.

Flowcharts use different shapes of boxes to denote different type of instructions. ANSI recommended a number of different rules and guidelines to help standardize the flowcharting process.

- Algorithms are represented using flowcharts
- Flowchart symbols are standardized by ANSI
- Flowchart helps to divide a large complex problem into small manageable ones
- Generally, algorithm is first represented as a flowchart and then expressed in a programming language
- While preparing a flowchart, the sequence, selection and iterative structures may be used wherever required

Note

Experienced programmers, sometimes write programs without drawing a flowchart. Beginners should first draw a flowchart to reduce number of errors in the program.

Rules for Drawing a Flowchart

- It should contain only one start and one end symbol
- The relevant symbols must be used while drawing a flowchart
- The direction of arrows should be top to bottom and left to right
- It should be simple and drawn clearly and neatly
- Be consistent in using names, variables in the flow chart
- Use properly labeled connectors to link the portions of the flowchart on different pages
- The branches of decision box must be label

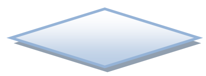
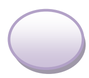
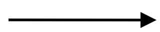
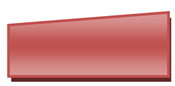
Advantages of Flowcharts

- Conveys better meaning
- Analyses the problem effectively
- Good tool for documentation
- Provide guide for coding
- Systematic debugging
- Systematic testing

Disadvantages of Flowcharts

- Takes more time to draw. Imagine developing a detailed flowchart for a program containing 50000 lines or statements of instructions
- Difficult to make changes
- Non-standardization - No standards to determine amount of details should be included in a flowchart

<u>Flowchart Symbols</u>

Symbol	Meaning
	**Start/Stop**
	**Process**
	**Input/Output**
	**Decision/Branching**
	**Connector**
	**Flow**
	**Manual Input**
	**Predefined Process**

PROGRAMMING CONSTRUCTS

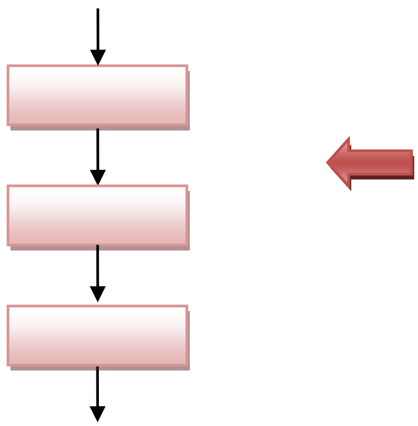
There are THREE basic programming constructs. They are:

- SEQUENCE
- SELECTION
- ITERATION

SEQUENCE

Sequence logic is used for performing instructions one after another in sequence.

- Sequence is the most basic of the constructs
- It is simply performing one step after another
- Each step is followed in a specific sequence, hence the name
- Sequence can be thought of as "do this, then do this, then do this"

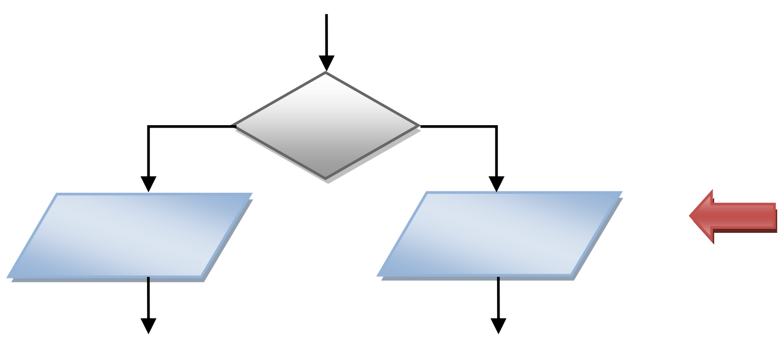


Sequence Flowchart

SELECTION

Selection logic, also known as decision logic, is used for making decisions. Selection logic is depicted as either an IF...THEN...ELSE or IF.....THEN structure.

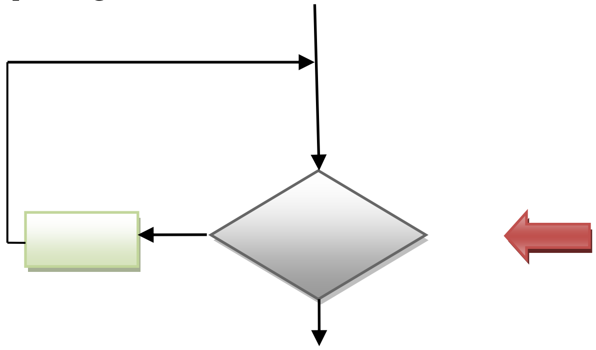
- Selection is the decision-making construct.
- It is used to make yes/no or true/false decisions logically.
- Selection can be thought of as "if something is true, take this action, otherwise take that action".



Selection Flowchart

ITERATION

Iteration logic is also known as **Loop**. Iteration logic is used when one or more instructions may be executed several times depending on some condition.



Iteration Flowchart

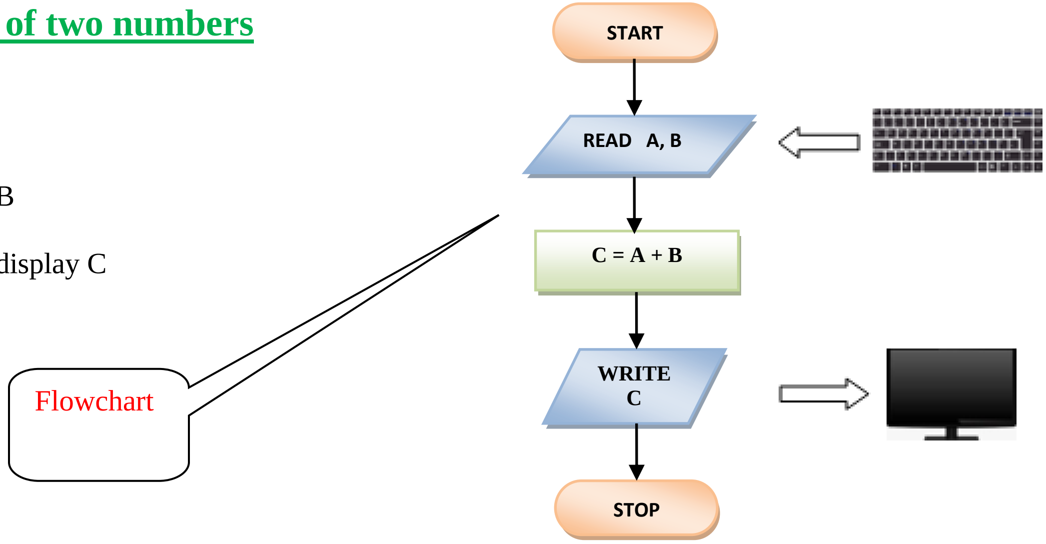
ITERATION

- Iteration comes from the word "reiterate", which means to repeat
- Iteration is a looping construct
- Iteration is a combination of decision and sequence and can repeat steps
- Iteration can be thought of as "while something is true, do this, otherwise stop"

To find sum of two numbers

Algorithm

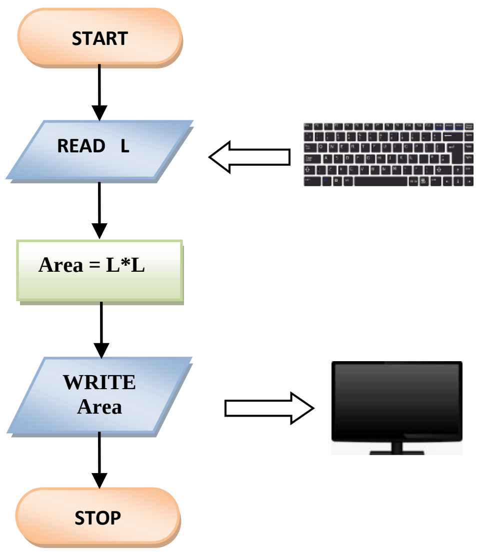
1. Start
2. Read A,B
3. C=A+B
4. Print or display C
5. Stop



Finding Area of the square

Algorithm

1. Start
2. Read length, L
3. Area = L*L
4. Print or display Area
5. Stop

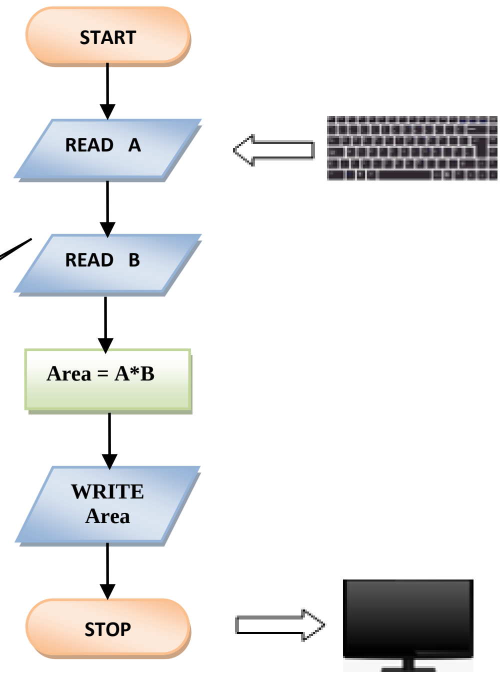


Finding Area of the rectangle

Algorithm

1. Start
2. Read side length, A
3. Read side Length B
4. Area = A*B
5. Print or display Area
6. Stop

Flowchart



Interchange the value of two numbers

Algorithm

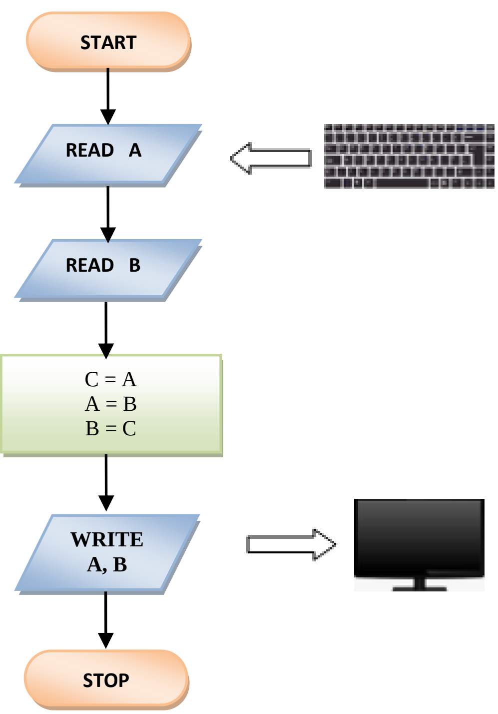
1. Start
2. Read two values into two variables a, b
3. Declare third variable, c

   c = a

   a = b

   b = c
4. Print or display a, b
5. Stop

Flowchart

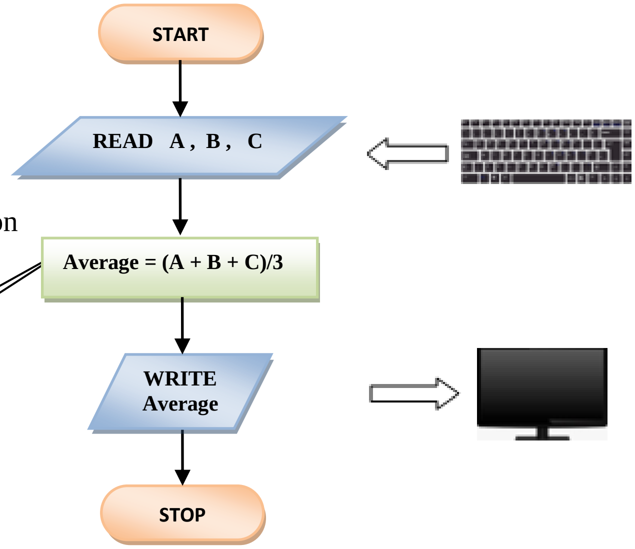


Calculating the average for 3 numbers

Algorithm

1. Start
2. Read 3 numbers A, B, C
3. Calculate the average by the equation
   Average = (A + B + C)/3
4. Print average
5. Stop

Flowchart

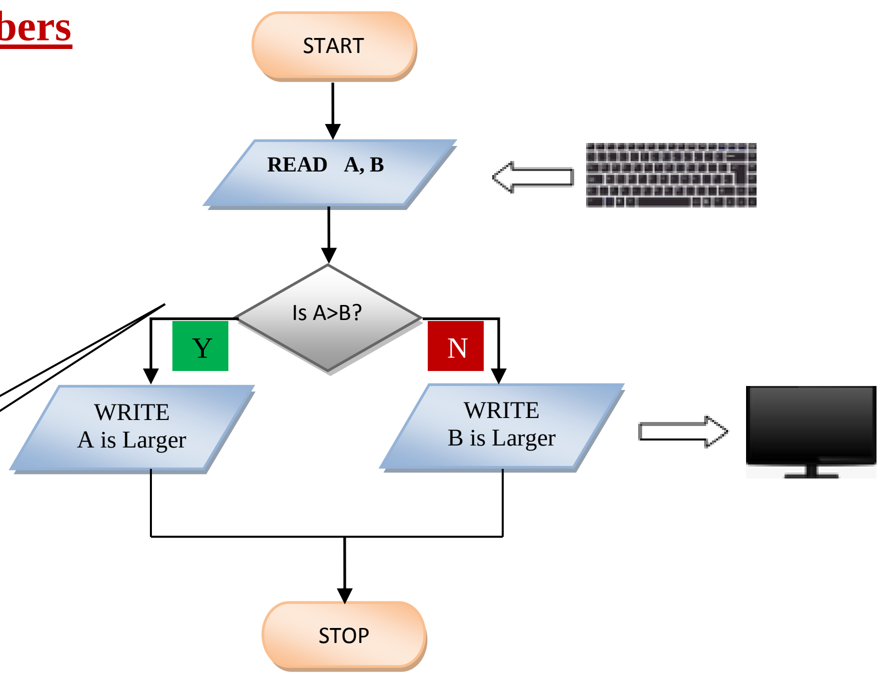


Greatest of two numbers

Algorithm

1. Start
2. Read A,B
3. If A>B then
   Print A is large
   else
   Print B is large
4. Stop

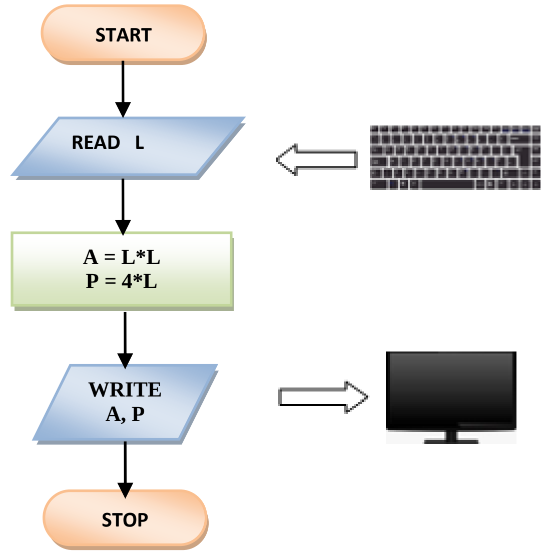
Flowchart



Find the area & perimeter of a square

Algorithm

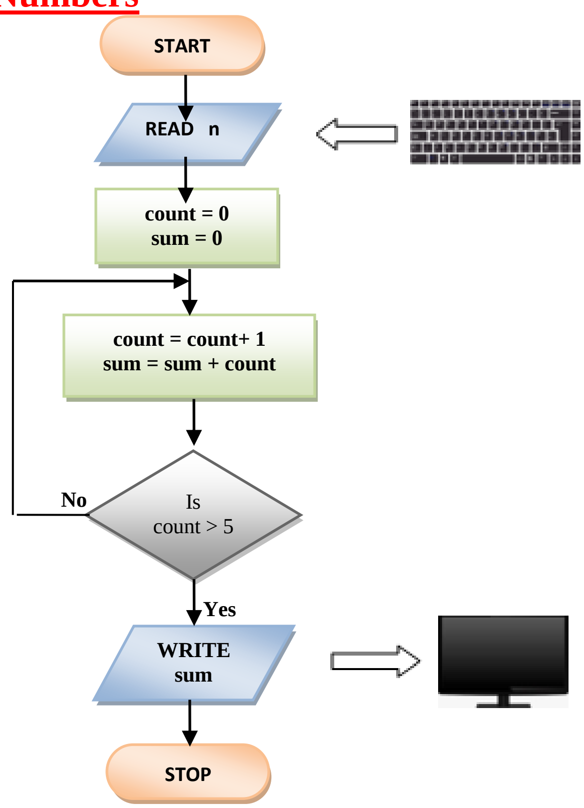
1. Start
2. Read length L
3. Area A = L*L
4. Perimeter P = 4*L
5. Print or display A, P
6. Stop

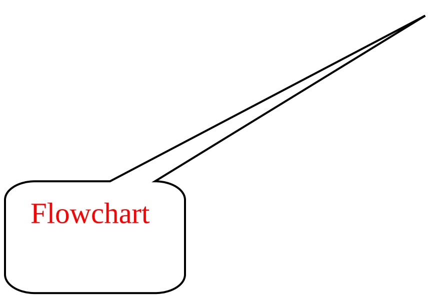


Find the Sum of First Five Natural Numbers

Algorithm

1. Start
2. Read n, initialize count = 0, sum = 0
3. count = count + 1
4. sum = sum + count
5. Repeat steps 3,4 until count>5
6. Print sum
7. Stop



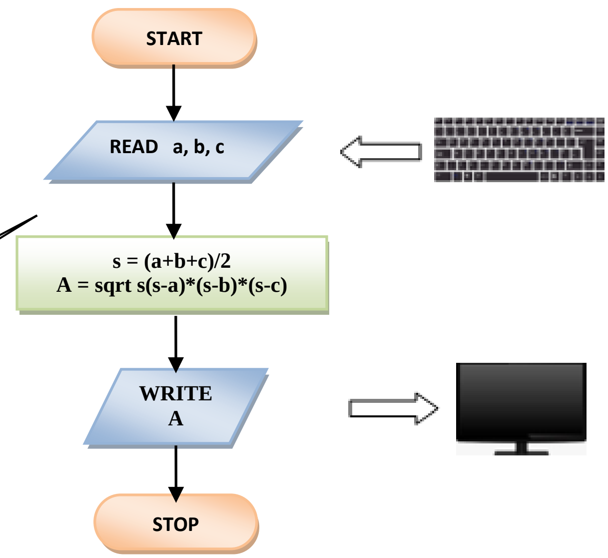


Area of a triangle where three sides are given

Algorithm

1. Start
2. Read a, b, c
3. s=(a+b+c)/2
4. A=sqrt (s *(s-a)*(s-b)*(s-c))
5. Print or display A
6. Stop

Flowchart

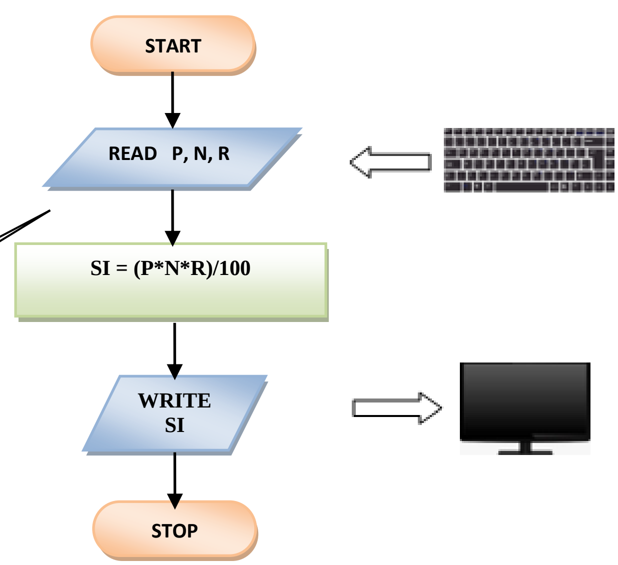


Calculate simple interest using the expression (SI=PNR/100)

Algorithm

1. Start
2. Read P, N, R
3. SI=(PNR)/100
4. Print SI
5. Stop

Flowchart

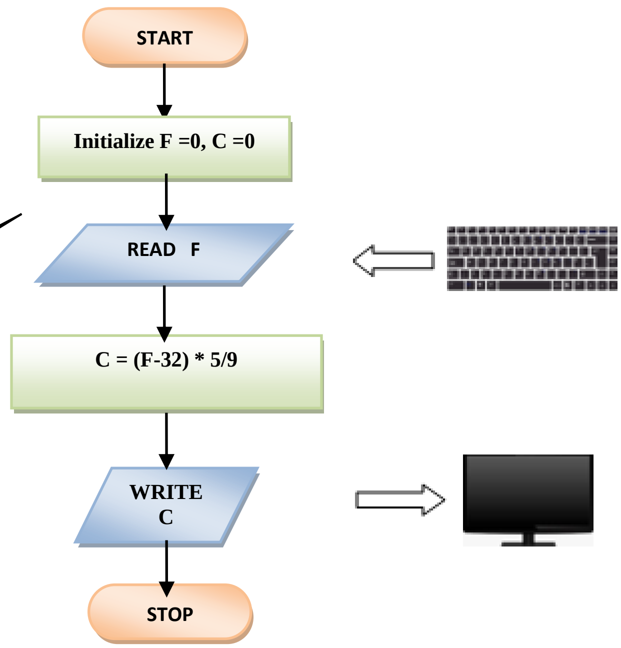


Convert temperature from Fahrenheit to Celsius

Algorithm

1. Start
2. Initialize F = 0, C = 0
3. Read F
4. C = (F-32) * 5/9
5. Write C
6. Stop

Flowchart

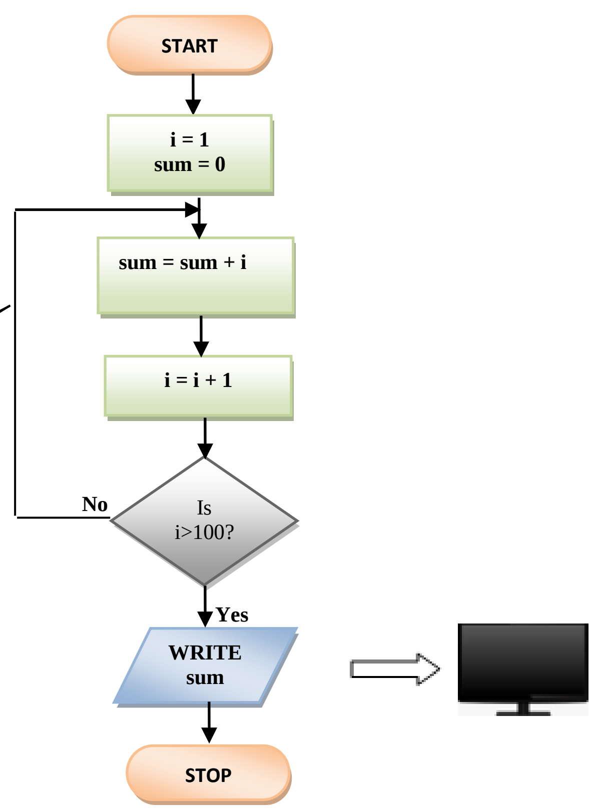


Calculating sum of integers 1 to 100

Algorithm

1. Start
2. Initialize count i = 1, sum = 0
3. sum = sum + i
4. Increment i by 1
5. Repeat steps 3 & 4 until i > 100
6. Print sum
7. Stop

Flowchart

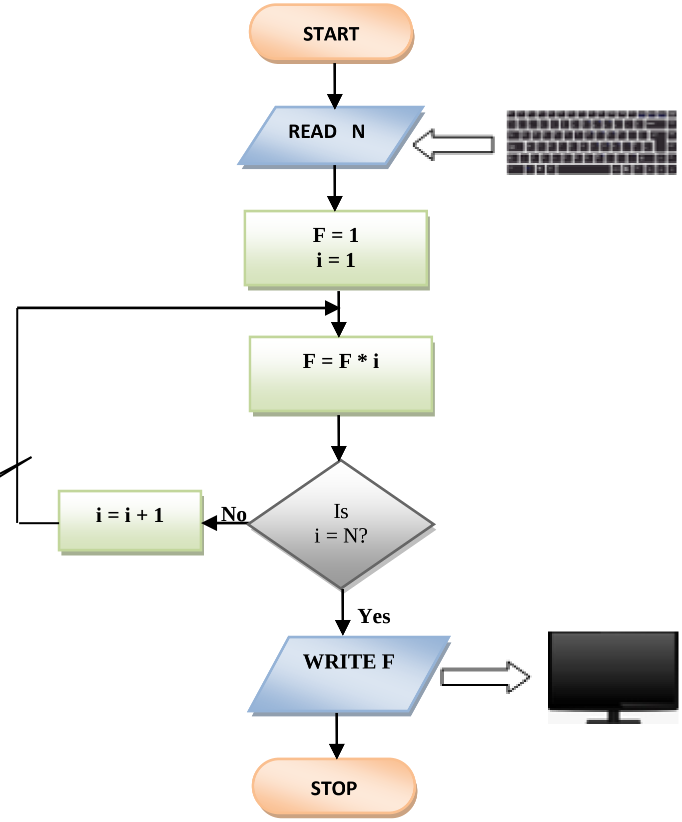


Draw a flowchart for computing factorial N
Where N! = 1 * 2 * 3 * …… N

Algorithm

1. Start
2. Read N
3. Initialize F =1, i = 1
4. F = F * i;
5. Increment i by 1
6. Repeat step 4 & 5 until i = N
7. Print F
8. Stop

Flowchart

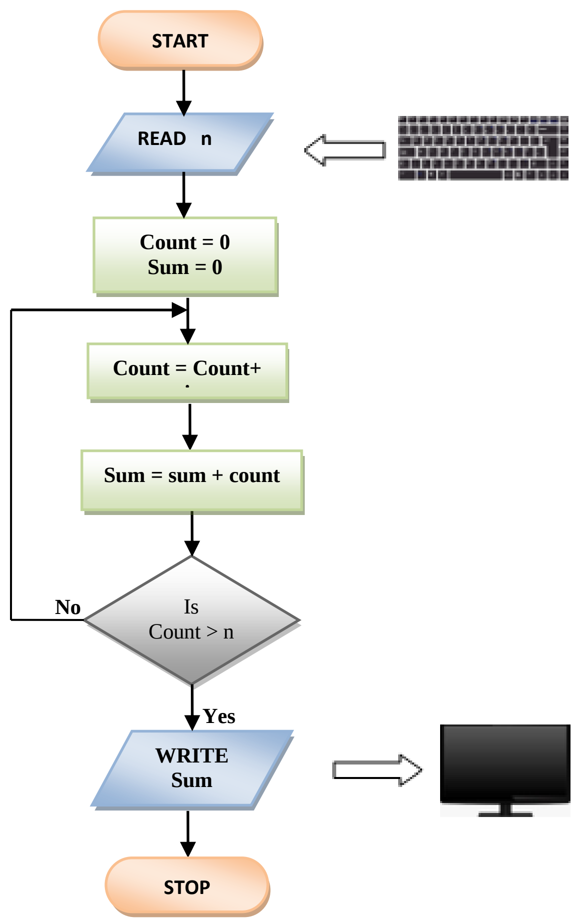


To find the sum of n natural Numbers

Algorithm

8. Start
9. Read n
10. Count=0
11. Sum=0
12. Count=Count+1
13. Sum=Sum + Count
14. Repeat steps 5 & 6 until
15. Count>n
16. Print sum
17. Stop

Flowchart

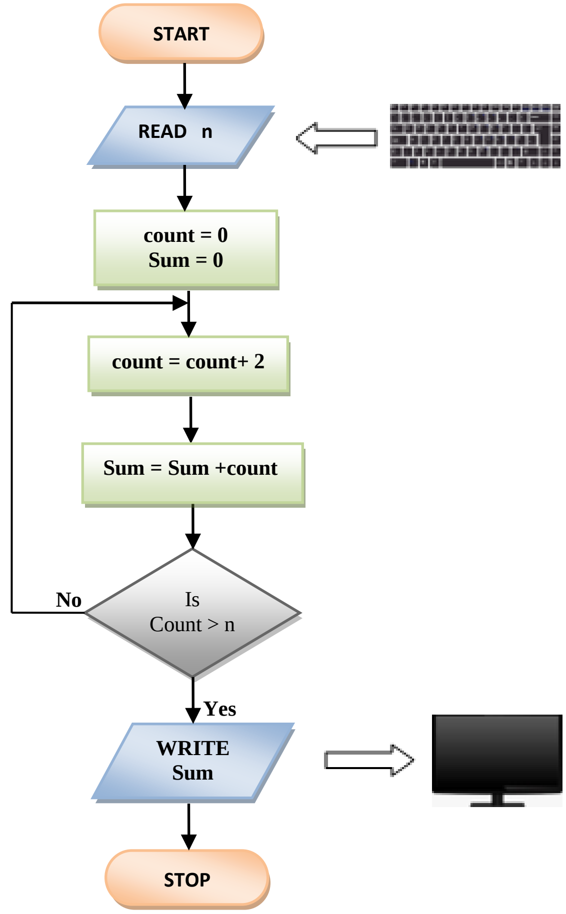


To find the sum of all even numbers up to 'n'

Algorithm

1. Start
2. Read n
3. Count=0
4. Sum=0
5. Count=count+2
6. Sum=sum + count
7. 7Repeat steps 5 &6 until count>=n
8. Print sum
9. Stop

Flowchart

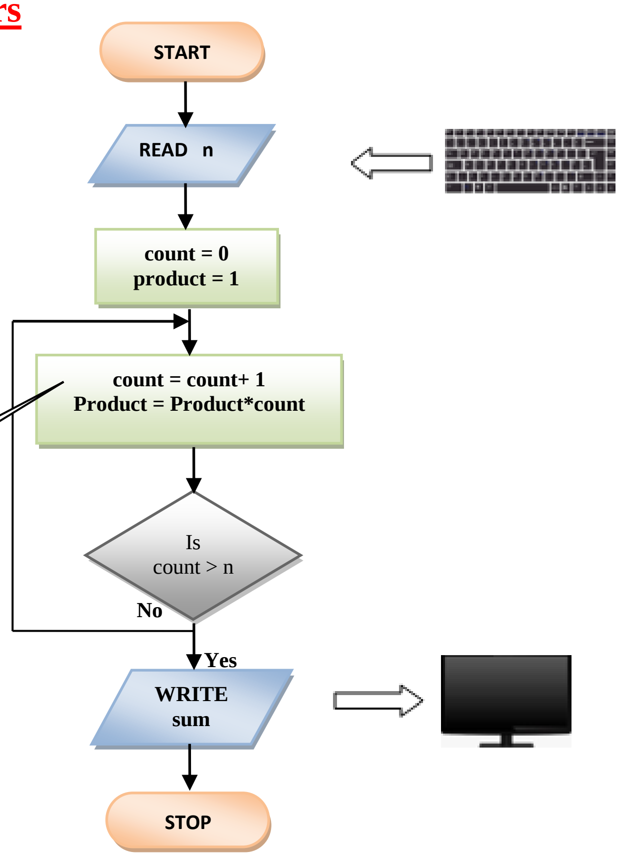


To find Product of N numbers

Algorithm

1. Start
2. Read n
3. Count=0
4. Product=1
5. Count=count+1
6. Product=count*product
7. Repeat steps 5,6 until count>N
8. Print Product
9. Stop

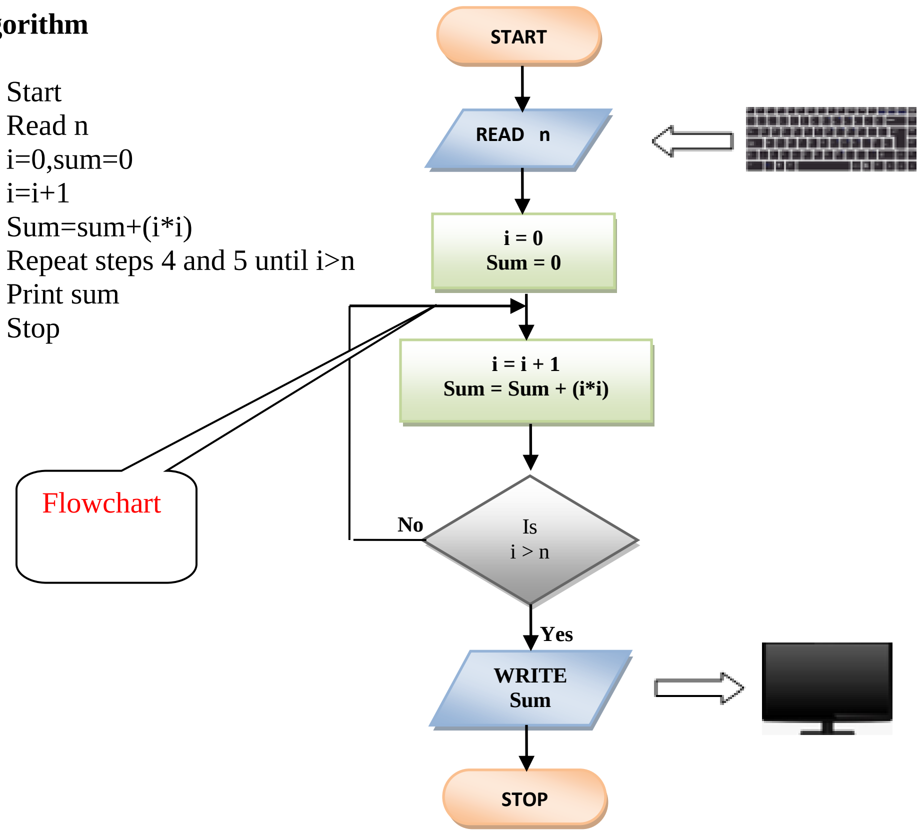
Flowchart



Sum of squares of n natural numbers

Algorithm

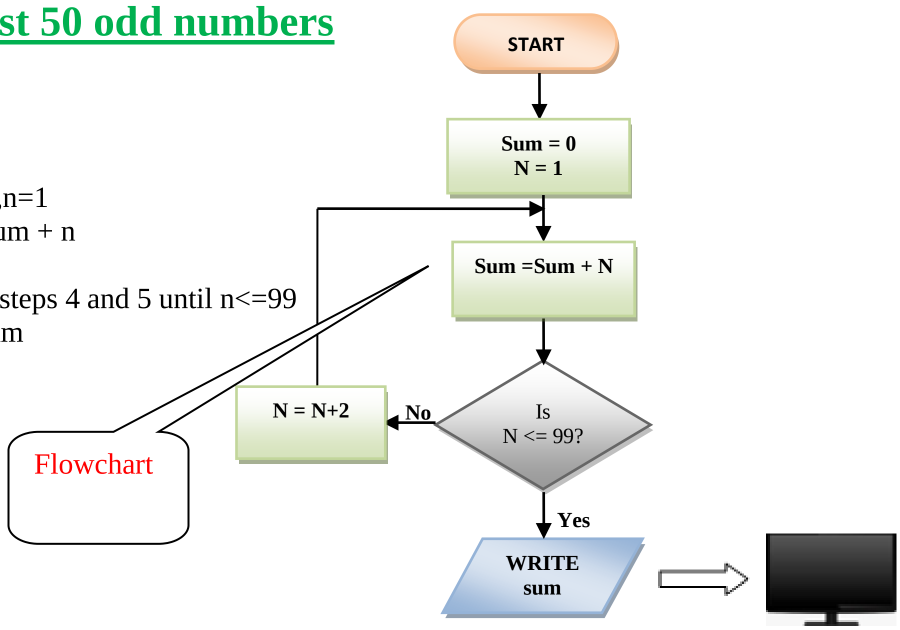
1. Start
2. Read n
3. i=0,sum=0
4. i=i+1
5. Sum=sum+(i*i)
6. Repeat steps 4 and 5 until i>n
7. Print sum
8. Stop



Sum of first 50 odd numbers

Algorithm

1. Start
2. Sum=0,n=1
3. Sum=sum + n
4. n=n+2
5. Repeat steps 4 and 5 until n<=99
6. Print sum
7. Stop

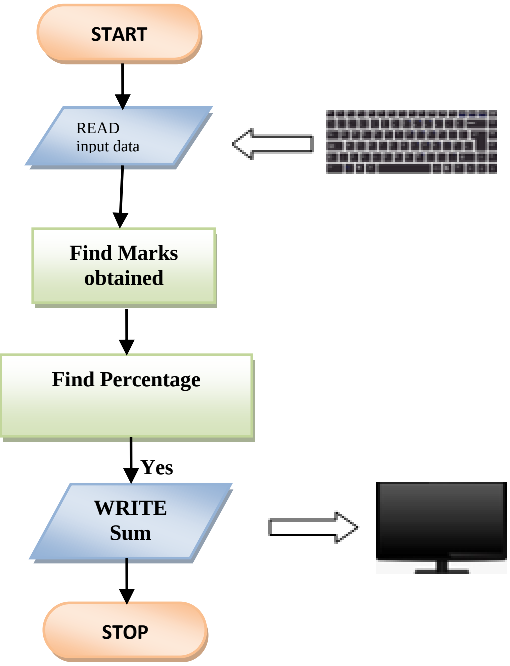


Calculating percentage of marks

Algorithm

1. Start
2. Read Input data
3. Add marks of all subjects giving total
4. Percentage = $\frac{Marks\ obtained}{Total\ marks} * 100$
5. Write Percentage
6. Stop

Flowchart



To find the sum of all even numbers up to 'n'

Algorithm

1. Start
2. Read n
3. Count=0
4. Sum=0
5. Count=count+2
6. Sum=sum+count
7. Repeat steps 5 &6 until count>=n
8. Print sum
9. Stop

Flowchart