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.

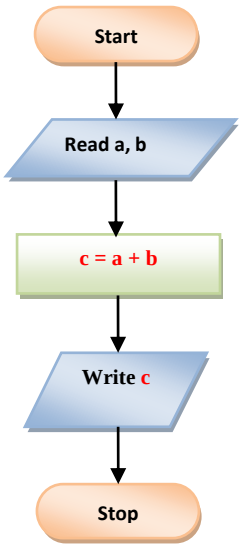
Examples of Algorithms and Flowcharts (with MATLAB code)

1. To find sum of two numbers

Algorithm

1. Start
2. Read a, b
3. c = a + b
4. Print or display c
5. Stop

Flowchart



Program

1. Open **MATLAB** software
2. **File > New > Blank m-file**
3. Type below program

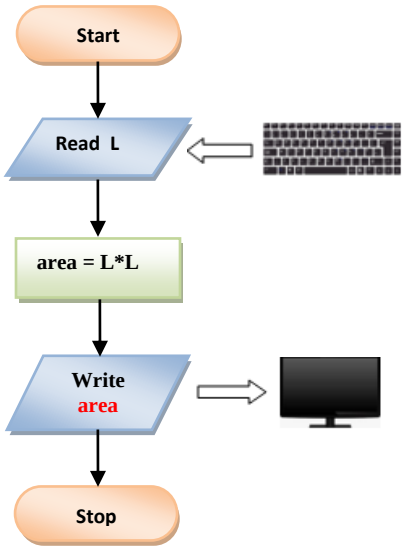
```
a = input(' Enter value of a: ');
b = input(' Enter value of b: ');

c = a+b;
disp(c);
```

2. Finding Area of the square

Algorithm

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

Flowchart



Program

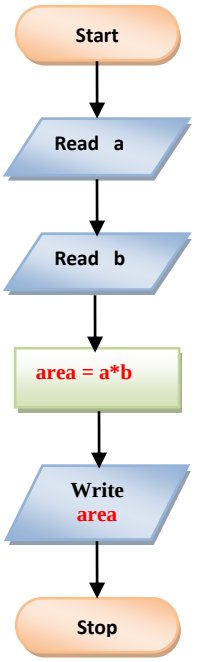
```
%Program to find area of a square

L = input(' Enter length of square L: ');

area = L * L;
disp(' Area of square is: ');
disp(area)
```

3. 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



Program

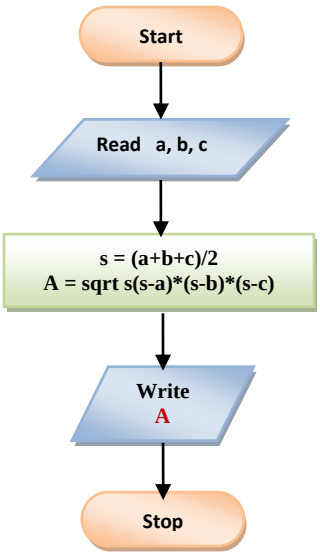
```
%Program to find area of a rectangle

a = input(' Enter side length a: ');
b = input(' Enter side length b: ');

area = a * b;
disp('Area of rectangle is: ');
disp(area);
```

4. 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



Program

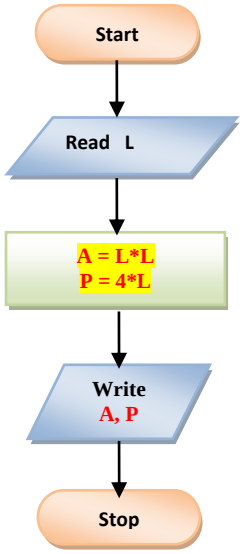
```
%Area of a triangle with 3 sides

A = input(' Enter value of a: ');
A = input(' Enter value of b: ');
A = input(' Enter value of c: ');

S = (a+b+c)/2;

A = sqrt(s*(s-a)*(s-b)*(s-c));

disp(' Area of triangle is: ');
disp(A);
```

5. 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

Flowchart



Program

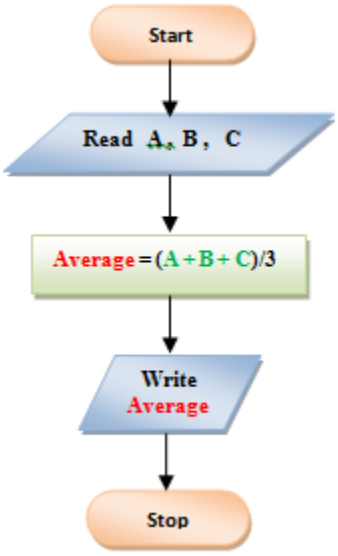
```
% Program to find area & perimeter of square

L = input('Enter length of a square: ');

A = L*L;
P = 4*L;

disp(' Area of square is: ');
disp(A);

disp(' Perimeter of square is: ');
disp(P);
```

6. 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



Program

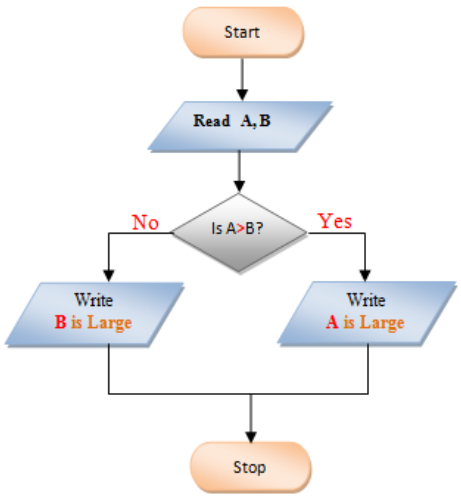
```
%Average of given 3 numbers

A = input(' Enter value of A: ');
B = input(' Enter value of B: ');
C = input(' Enter value of C: ');

Average = (A+B+C)/3;

disp(' Average of A, B, C is: ');
disp(Average);
```

7. 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



Program

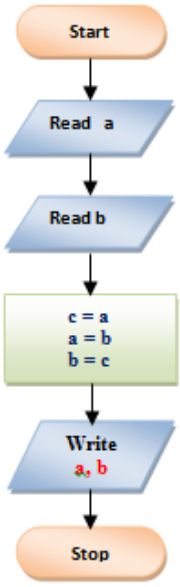
```
%Program to find greatest of two numbers

A = input(' Enter value of A: ');
B = input(' Enter value of B: ');

if (A>B)
        disp(' A is Larger: ');
else
        disp(' B is Larger');
end
```

8. 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



Program

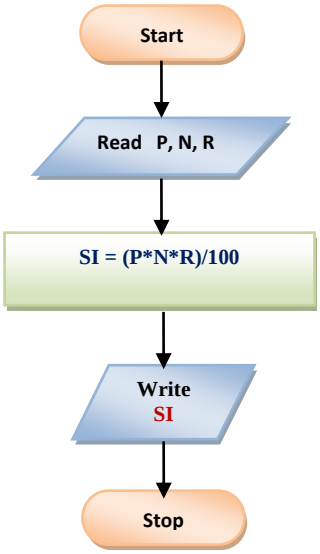
```
%Interchange values of two variables

a = input(' Enter value of a: ');
b = input(' Enter value of b: ');

c = a;
a = b;
b = c;

disp(' Values of a and b after swapping: ');
disp(' a = ');
disp(a);

disp(' b = ');
disp(b);
```

9. 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



Program

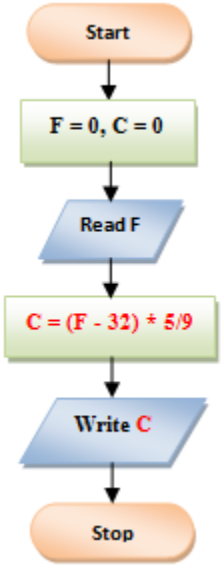
```
% Calculate simple interest

P = input(' Enter Principal: ');
N = input(' Enter Time: ');
R = input(' Enter Rate of interest: ');

SI = (P*N*R)/100;

disp(' Simple interest is: ');
disp(SI);
```

10. 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



Program

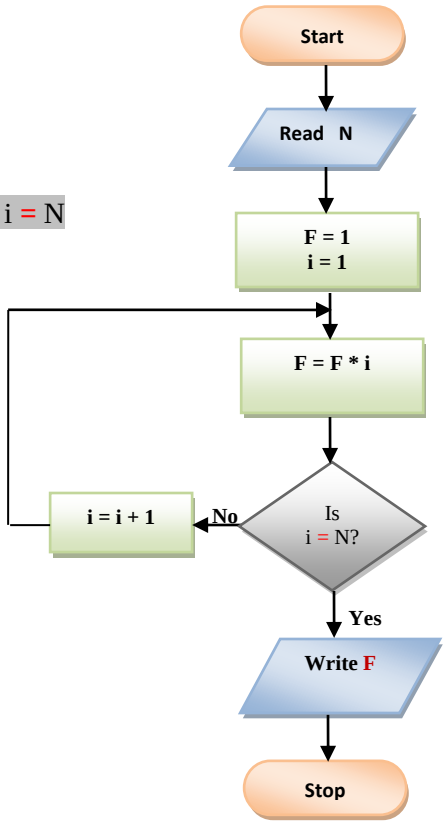
```
%Convert Fahrenheit to Celsius

F = input(' Enter Temp. in Fahrenheit: ');

C = (F-32) * 5/9;

disp(' Temp. in Celsius is: ');
disp(C);
```

11. 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



Program

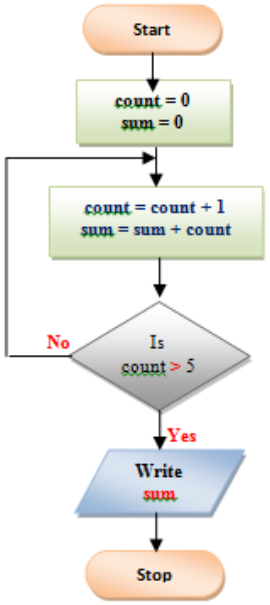
```
%Compute factorial of given number N

N = input(' Enter value of N: ');
F = 1;

for i = 1 : N
F = F * i;
end

disp(' Factorial is: ');
disp(F);
```

12. Find the Sum of First Five Natural Numbers

Algorithm

1. Start
2. 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

Flowchart



Program

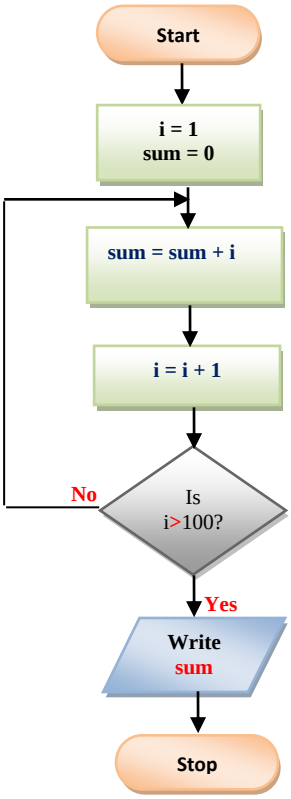
```
% Sum of 1st 5 natural numbers

sum = 0;
for count = 1 : 5
sum = sum + count;
end

disp(' Sum of 1st 5 numbers is: ');
disp(sum);
```

13. Calculating sum of integers from 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



Program

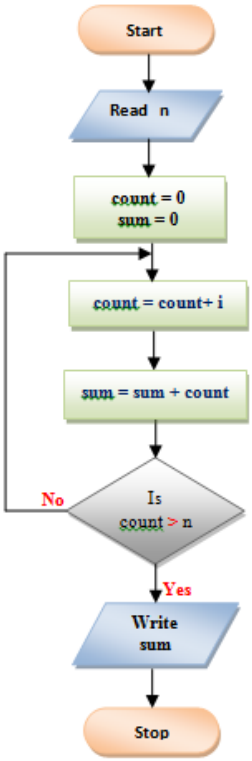
```
% Sum of integers from 1 to 100

sum = 0;
for count = 1 : 100
sum = sum + count;
end

disp(' Sum of integers from 1 to 100 is: ');
disp(sum);
```

14. To find the sum of n natural Numbers

Algorithm

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

Flowchart



Program

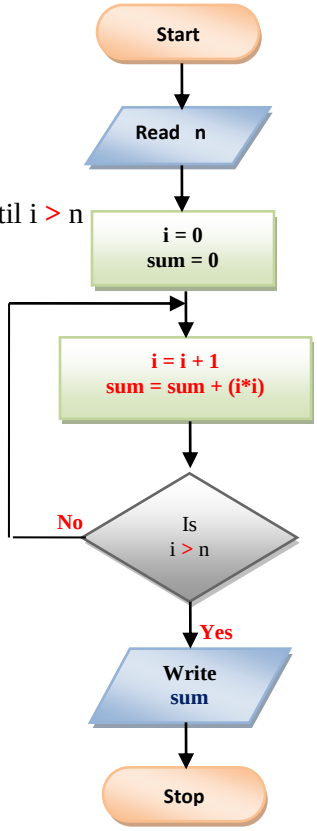
```
%Sum of n natural numbers

N = input(' Enter value of n: ');

sum = 0;
for i = 1 : n
sum = sum + i;
end

disp(' Sum of n natural numbers is: ');
disp(sum);
```

15. 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

Flowchart



Program

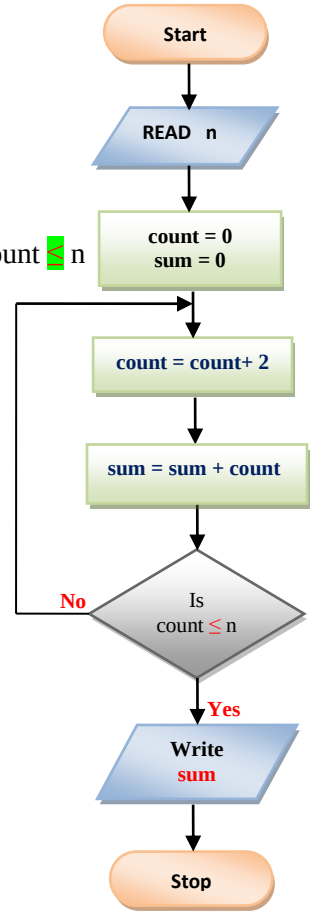
```
% Sum of squares of n natural numbers

n = input(' Enter value of n: ');

sum = 0;
for i = 1 : n
sum = sum + i*i;
end

disp(' Sum of squares of integers up to n: ');
disp(sum);
```

16. 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



Program

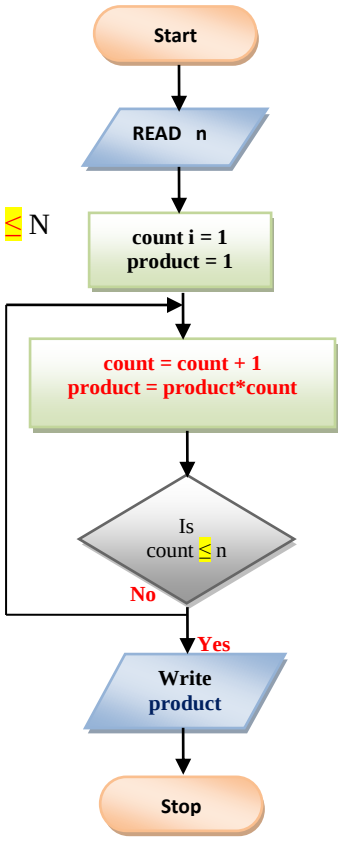
```
%Sum of even numbers up to n

n = input(' Enter value of n: ');

sum = 0;
for i = 0 : 2: n
sum = sum + i;
end;

disp(' Sum of even numbers up to n: ');
disp(sum);
```

17. To find Product of numbers up to N

Algorithm

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

Flowchart



Program

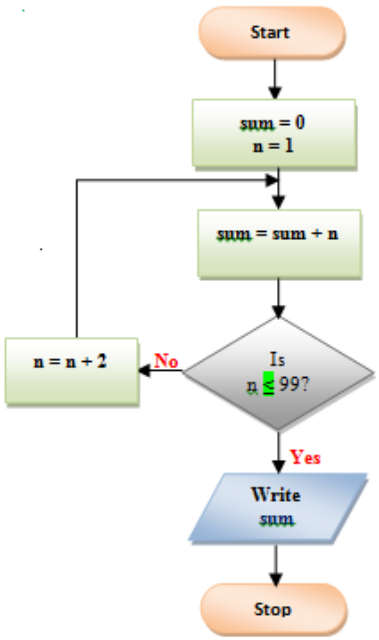
```
%Product of numbers up to n

n = input(' Enter value of n: ')

prod = 1;
for i = 1 : n
prod = prod * i;
end

disp(' Product of numbers up to n: ');
disp(prod);
```

18. 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

Flowchart



Program

```
%Sum of 1st 50 odd numbers

sum = 0;
n = 1;

while(n<=99)
sum = sum + n;
n = n + 2;
end

disp(' Sum of 1st 50 odd numbers is: ');
disp(sum);
```