



SIMPLE STRATEGIES FOR DEVELOPING ALGORITHMS:

1. iterations
2. Recursions

1. Iterations:

A sequence of statements is executed until a specified condition is true is called iterations.

1. for loop
2. While loop

Syntax for For:

```
FOR( start-value to end-value) DO  
Statement  
...  
ENDFOR
```

Example: Print n natural numbers

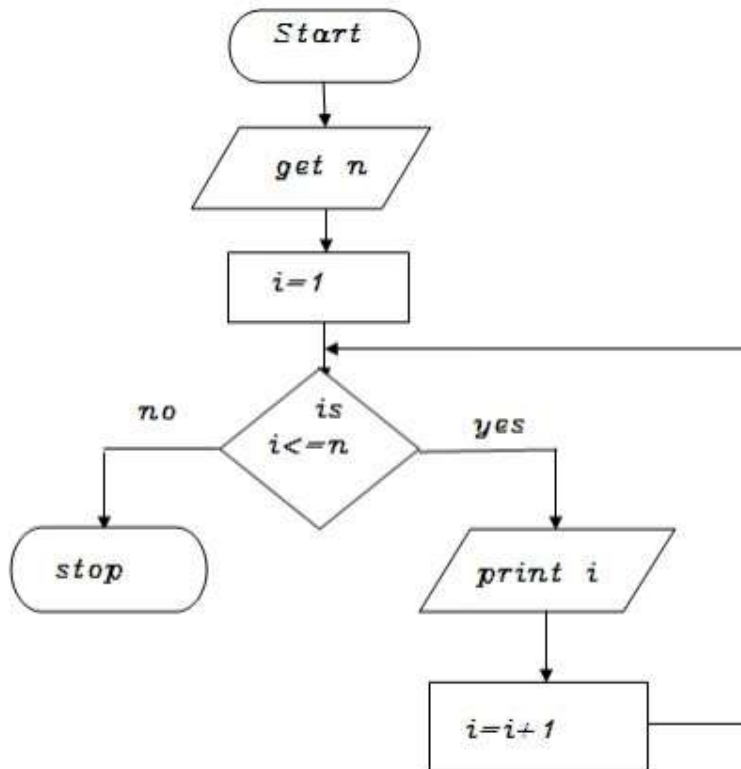
```
BEGIN  
GET n  
INITIALIZE i=1  
FOR (i<=n) DO  
PRINT i  
i=i+1  
ENDFOR  
END
```

Syntax for While:

```
WHILE (condition) DO  
Statement  
...  
ENDWHILE
```

Example: Print n natural numbers

```
BEGIN  
GET n  
INITIALIZE i=1  
WHILE(i<=n) DO  
PRINT i  
i=i+1  
ENDWHILE  
END
```



Recursions:

-
- v A function that calls itself is known as recursion.
- v Recursion is a process by which a function calls itself repeatedly until some specified condition has been satisfied.

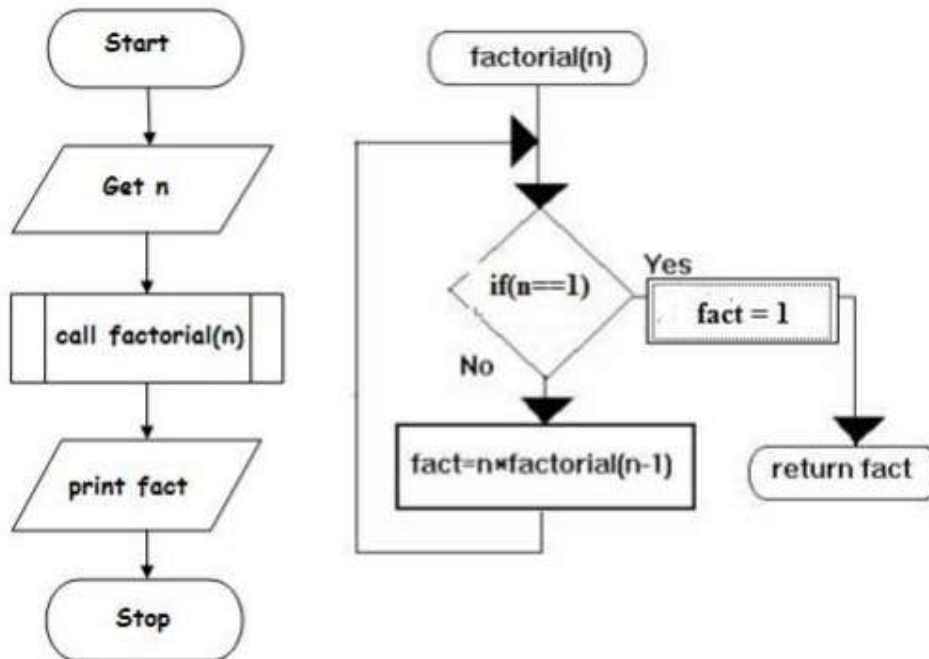
Algorithm for factorial of n numbers using recursion:

Main function:

- Step1: Start
- Step2: Get n
- Step3: call factorial(n)
- Step4: print fact
- Step5: Stop

Sub function factorial(n):

- Step1: if(n==1) then fact=1 return fact
- Step2: else fact=n*factorial(n-1) and return fact



Pseudo code for factorial using recursion:

Main function:

```
BEGIN
GET n
CALL factorial(n)
PRINT fact
BIN
```

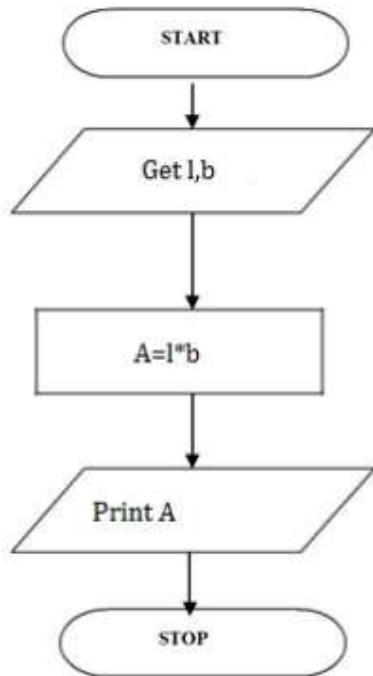
Sub function factorial(n):

```
IF(n==1) THEN
    fact=1
    RETURN fact
ELSE
    RETURN fact=n*factorial(n-1)
```

More examples:

Write an algorithm to find area of a rectangle

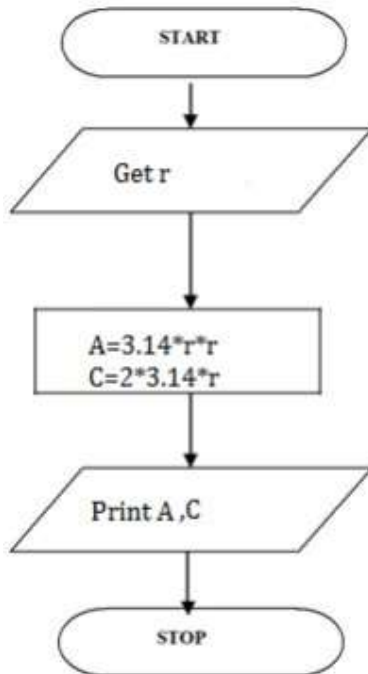
- Step 1: Start
- Step 2: get l,b values
- Step 3: Calculate $A=l*b$
- Step 4: Display A
- Step 5: Stop



```
BEGIN  
READ l,b  
CALCULATE A=l*b  
DISPLAY A  
END
```

Write an algorithm for Calculating area and circumference of circle

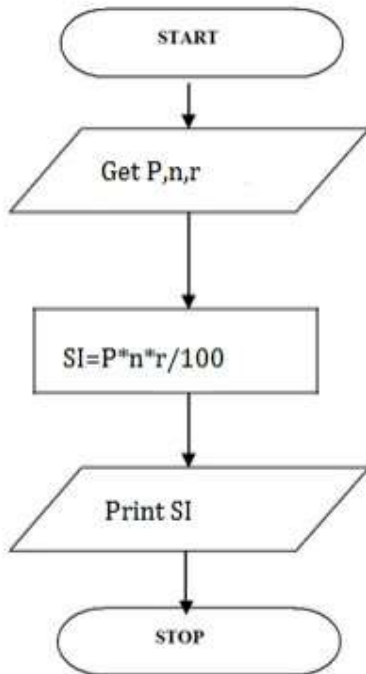
Step 1: Start
Step 2: get r value
Step 3: Calculate $A=3.14*r*r$
Step 4: Calculate $C=2.3.14*r$
Step 5: Display A,C
Step 6: Stop



BEGIN
READ r
CALCULATE A and C
 $A=3.14*r*r$
 $C=2*3.14*r$
DISPLAY A
END

Write an algorithm for Calculating simple interest

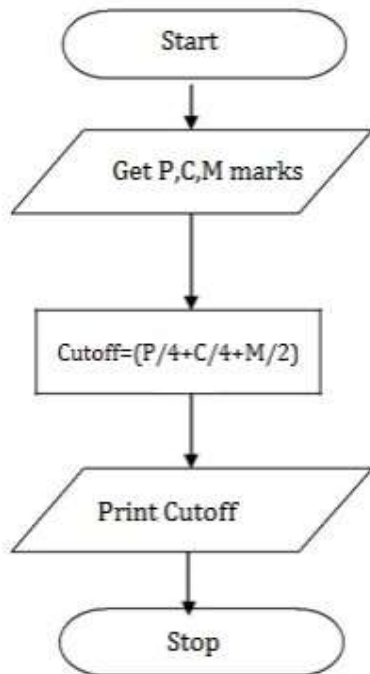
Step 1: Start
Step 2: get P, n, r value
Step3:Calculate
 $SI=(p*n*r)/100$
Step 4: Display S
Step 5: Stop



```
BEGIN  
READ P, n, r  
CALCULATE S  
SI=(p*n*r)/100  
DISPLAY SI  
END
```

Write an algorithm for Calculating engineering cutoff

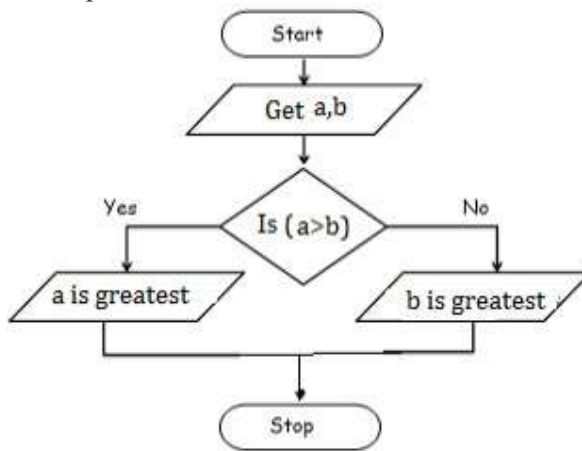
```
Step 1: Start  
Step2: get P,C,M value  
Step3:calculate  
Cutoff= (P/4+C/4+M/2)  
Step 4: Display Cutoff  
Step 5: Stop
```



BEGIN
READ P,C,M
CALCULATE
Cutoff= (P/4+C/4+M/2)
DISPLAY Cutoff
END

To check greatest of two numbers

Step 1: Start
Step 2: get a,b value
Step 3: check if(a>b) print a is greater
Step 4: else b is greater
Step 5: Stop



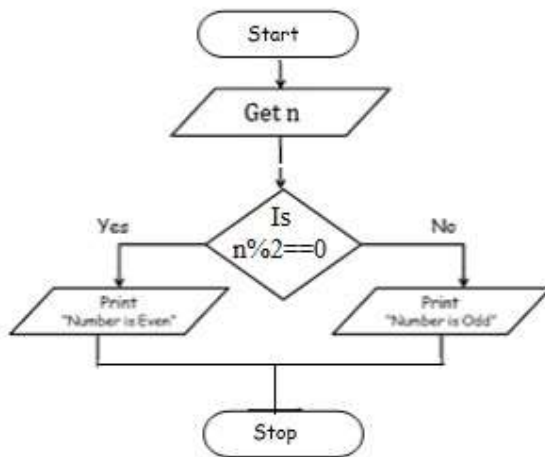
BEGIN



```
READ a,b
IF (a>b) THEN
DISPLAY a is greater
ELSE
DISPLAY b is greater
END IF
END
```

To check leap year or not

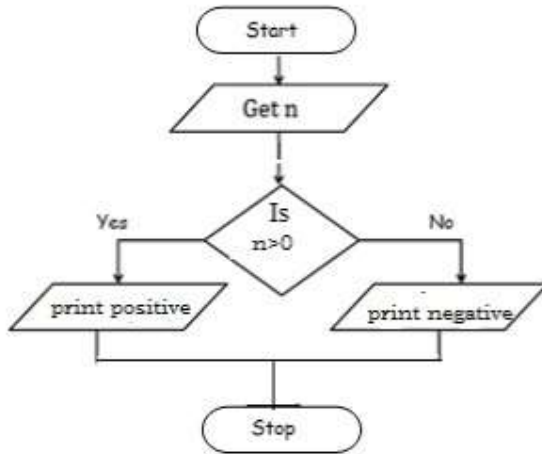
Step 1: Start
Step 2: get y
Step 3: if($y\%4==0$) print leap year
Step 4: else print not leap year
Step 5: Stop



```
BEGIN
READ y
IF (y%4==0) THEN
DISPLAY leap year
ELSE
DISPLAY not leap year
END IF
END
```

To check positive or negative number

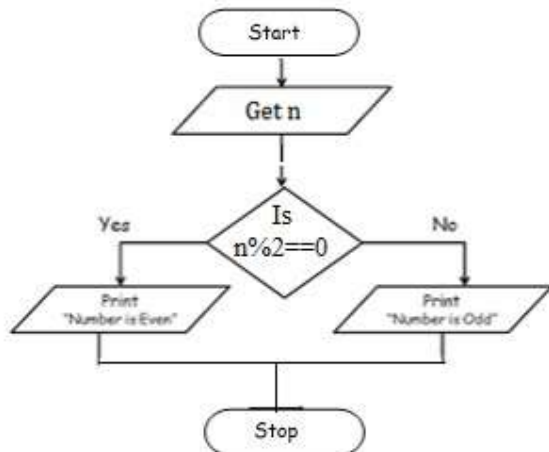
Step 1: Start
Step 2: get num
Step 3: check if($\text{num}>0$) print a is positive
Step 4: else num is negative
Step 5: Stop



```
BEGIN  
READ num  
IF (num>0) THEN  
  DISPLAY num is positive  
ELSE  
  DISPLAY num is negative  
END IF  
END
```

To check odd or even number

Step 1: Start
Step 2: get num
Step 3: check if(num%2==0) print num is even
Step 4: else num is odd
Step 5: Stop



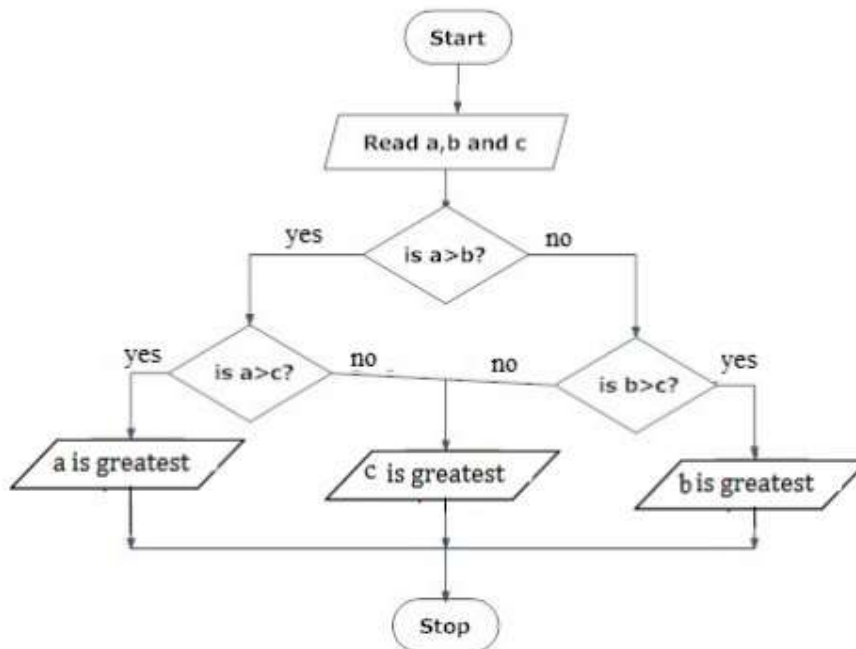
```
BEGIN  
READ num  
IF (num%2==0) THEN  
  DISPLAY num is even  
ELSE  
  DISPLAY num is odd  
END IF
```



END IF
END

To check greatest of three numbers

Step1: Start
Step2: Get A, B, C
Step3: if(A>B) goto Step4 else goto step5
Step4: If(A>C) print A else print C
Step5: If(B>C) print B else print C
Step6: Stop

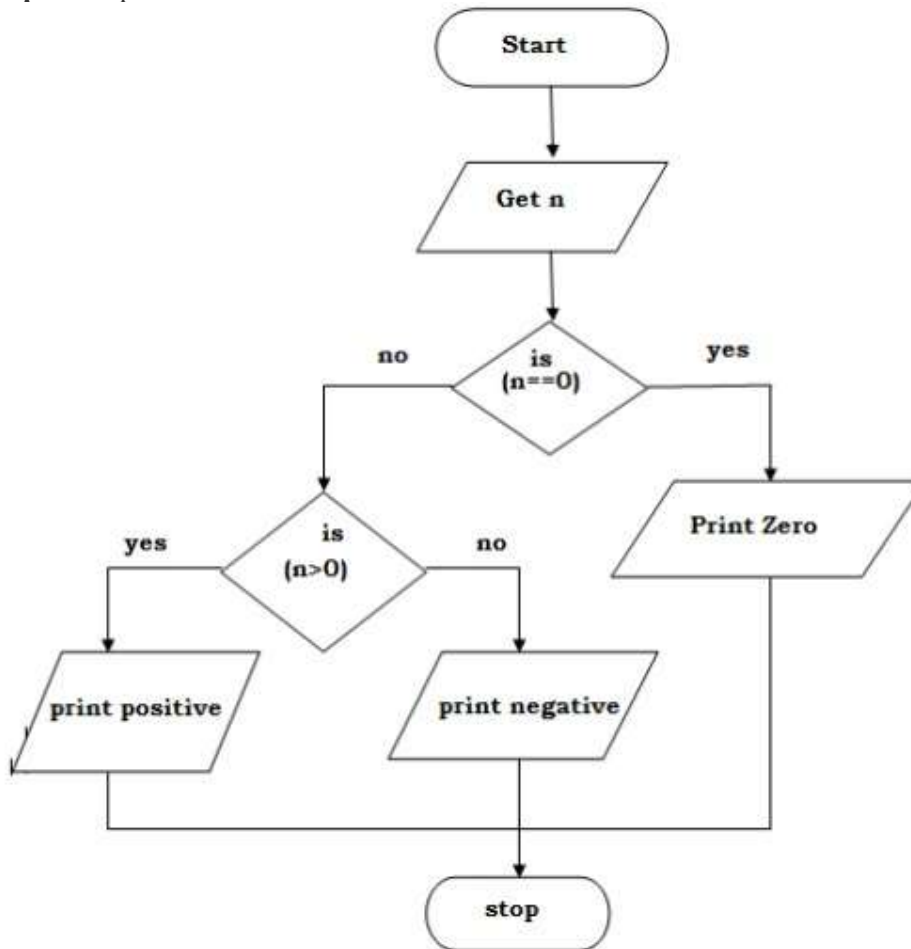


```
BEGIN  
READ a, b, c  
IF (a>b) THEN  
IF(a>c) THEN  
DISPLAY a is greater  
ELSE  
DISPLAY c is greater  
END IF  
ELSE  
IF(b>c) THEN  
DISPLAY b is greater  
ELSE  
DISPLAY c is greater  
END IF  
END IF  
END
```

Write an algorithm to check whether given number is +ve, -ve or zero.



- Step 1:** Start
- Step 2:** Get n value.
- Step 3:** if (n ==0) print “Given number is Zero” Else goto step4
- Step 4:** if (n > 0) then Print “Given number is +ve”
- Step 5:** else Print “Given number is -ve”
- Step 6:** Stop



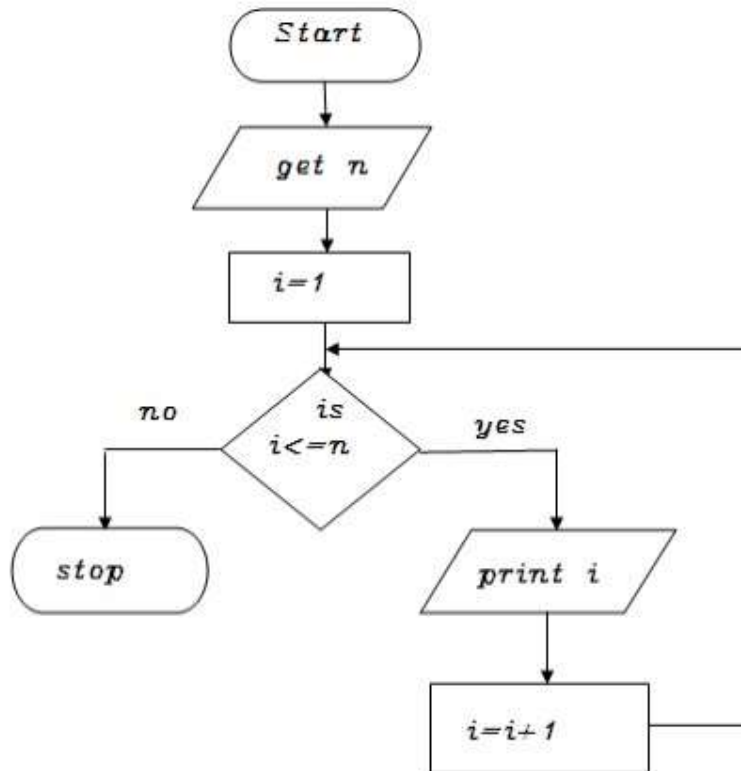
```
BEGIN  
GET n  
IF(n==0) THEN  
    DISPLAY “ n is zero”  
ELSE  
    IF(n>0) THEN  
        DISPLAY “n is positive”  
    ELSE  
        DISPLAY “n is positive”  
    END IF  
END IF  
END
```

Write an algorithm to print all natural numbers up to n

Step 1: Start



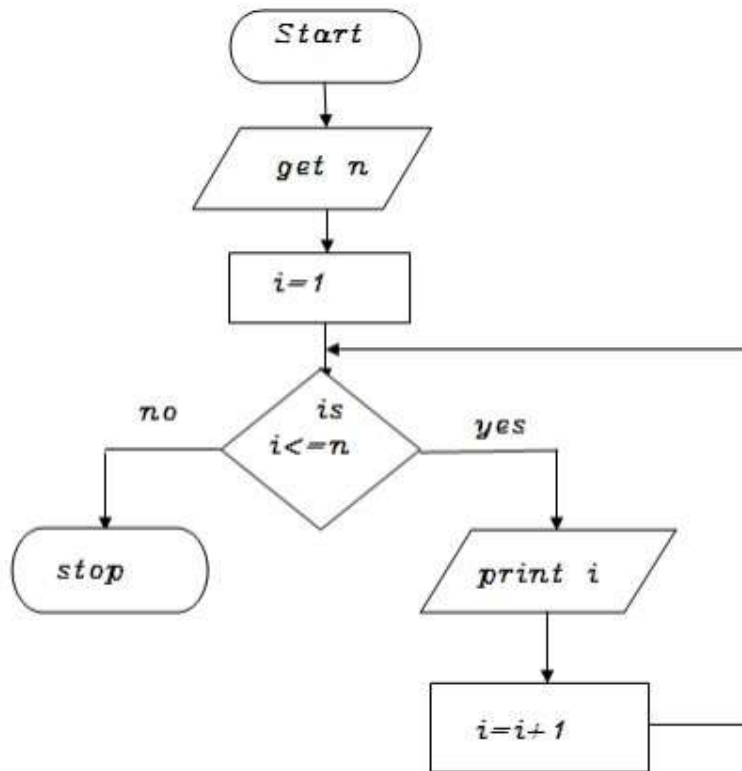
- Step 2: get n value.
- Step 3: initialize $i=1$
- Step 4: if ($i \leq n$) go to step 5 else go to step 8
- Step 5: Print i value
- step 6 : increment i value by 1
- Step 7: go to step 4
- Step 8: Stop



```
BEGIN  
GET n  
INITIALIZE i=1  
WHILE(i<=n) DO  
PRINT i  
i=i+1  
ENDWHILE  
END
```

Write an algorithm to print n odd numbers

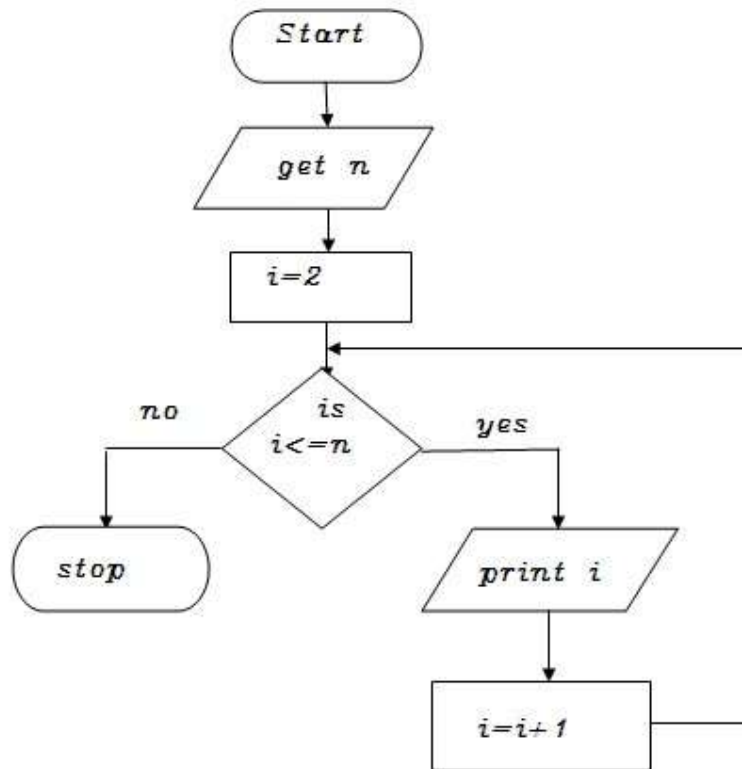
- Step 1: start
- step 2: get n value
- step 3: set initial value $i=1$
- step 4: check if($i \leq n$) goto step 5 else goto step 8
- step 5: print i value
- step 6: increment i value by 2
- step 7: goto step 4
- step 8: stop



```
BEGIN  
GET n  
INITIALIZE i=1  
WHILE(i<=n) DO  
    PRINT i  
    i=i+2  
ENDWHILE  
END
```

Write an algorithm to print n even numbers

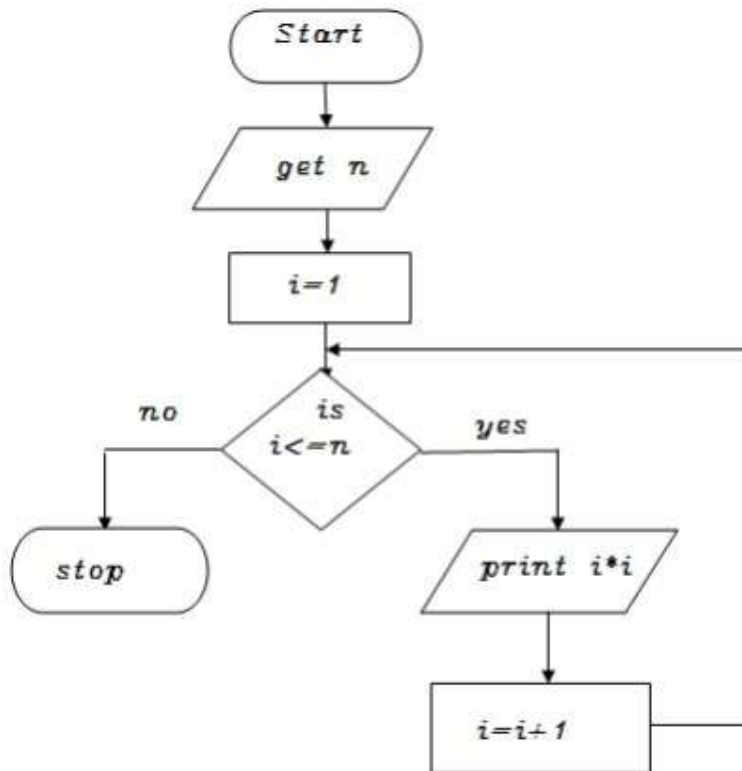
- Step 1: start
- step 2: get n value
- step 3: set initial value i=2
- step 4: check if(i<=n) goto step 5 else goto step8
- step 5: print i value
- step 6: increment i value by 2
- step 7: goto step 4
- step 8: stop



```
BEGIN  
GET n  
INITIALIZE i=2  
WHILE(i<=n) DO  
    PRINT i  
    i=i+2  
ENDWHILE  
END
```

Write an algorithm to print squares of a number

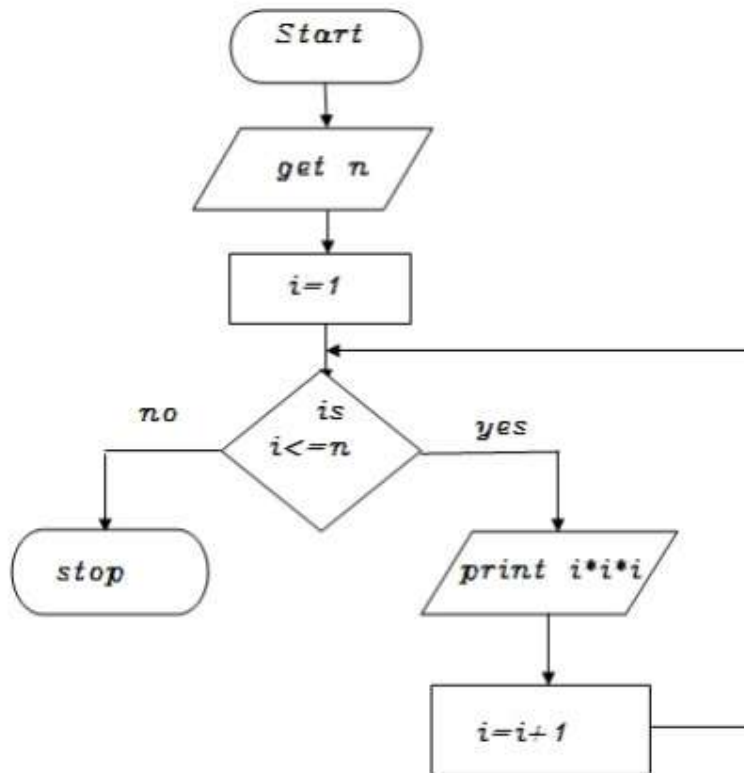
- Step 1: start
- step 2: get n value
- step 3: set initial value i=1
- step 4: check i value if(i<=n) goto step 5 else goto step8
- step 5: print i*i value
- step 6: increment i value by 1
- step 7: goto step 4
- step 8: stop



```
BEGIN  
GET n  
INITIALIZE i=1  
WHILE(i<=n) DO  
    PRINT i*i  
    i=i+1  
ENDWHILE  
END
```

Write an algorithm to print to print cubes of a number

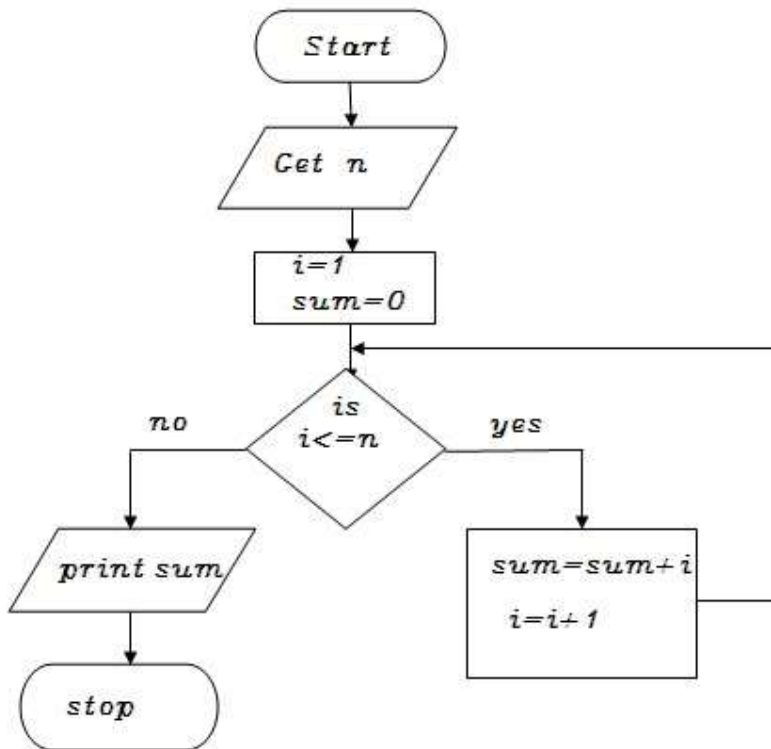
- Step 1: start
- step 2: get n value
- step 3: set initial value i=1
- step 4: check i value if(i<=n) goto step 5 else goto step8
- step 5: print i*i *i value
- step 6: increment i value by 1
- step 7: goto step 4
- step 8: stop



```
BEGIN  
GET n  
INITIALIZE i=1  
WHILE(i<=n) DO  
    PRINT i*i*i  
    i=i+1  
ENDWHILE  
END
```

Write an algorithm to find sum of a given number

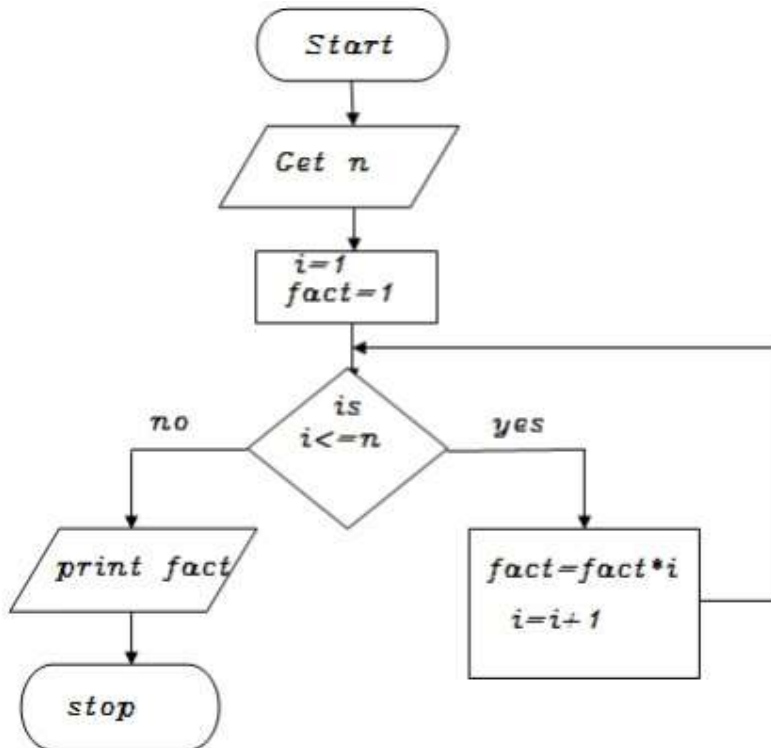
- Step 1: start
- step 2: get n value
- step 3: set initial value i=1, sum=0
- Step 4: check i value if(i<=n) goto step 5 else goto step8
- step 5: calculate sum=sum+i
- step 6: increment i value by 1
- step 7: goto step 4
- step 8: print sum value
- step 9: stop



```
BEGIN  
GET n  
INITIALIZE i=1,sum=0  
WHILE(i<=n) DO  
    sum=sum+i  
    i=i+1  
ENDWHILE  
PRINT sum  
END
```

Write an algorithm to find factorial of a given number

Step 1: start
step 2: get n value
step 3: set initial value i=1, fact=1
Step 4: check i value if(i<=n) goto step 5 else goto step8
step 5: calculate fact=fact*i
step 6: increment i value by 1
step 7: goto step 4
step 8: print fact value
step 9: stop



```
BEGIN  
GET n  
INITIALIZE i=1,fact=1  
WHILE(i<=n) DO  
    fact=fact*i  
    i=i+1  
ENDWHILE  
PRINT fact  
END
```