Symbol scanned		STACK
1	5	5
2	6	5,6
3	2	5,6,2
4	+	5,8
5	*	40
6	12	40,12
7	4	40,12,4
8	/	40, 3
9	—	37

Algorithms to Evaluate a Post fix Notation :-

Step-1 Initialize a stack.

Step 2. Repeat step 3 & 4 until the end of expression is en Countered.

Step 3 If an operand is encountered. then push it onto stack.

Step-4 If an operator is encountered then:

a) POP the two values from the stack.

b) apply the operator to these values.

c) Push the result of step 'b' back onto the stack.

{End of if structure}

{End of step 2 loop}

5. At the end of expression, result is contained at the top of the stack.

6. End.

Recursion:—

Recursion is a program to call itself again.

Ex - Factorial.

- Fibonacci series

5! = 1*2*3*4*5 =120

0, 1, 1, 2, 3, 5, 8, 13 -----

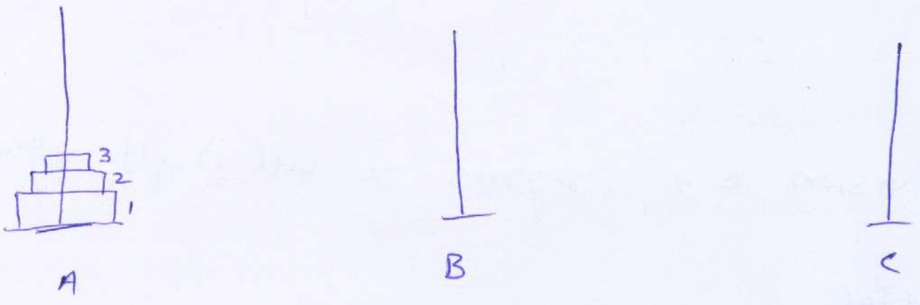
Algo. Factorial (FACT, N)

```
Step-1  If N=0 then:
            Set FACT = 1 and return
Step-2  set fact = 1  {initializes fact for loop}
3       Repeat for K = 1 to N
            Set fact = K * Fact
            [End of loop]
4.      Return.
```

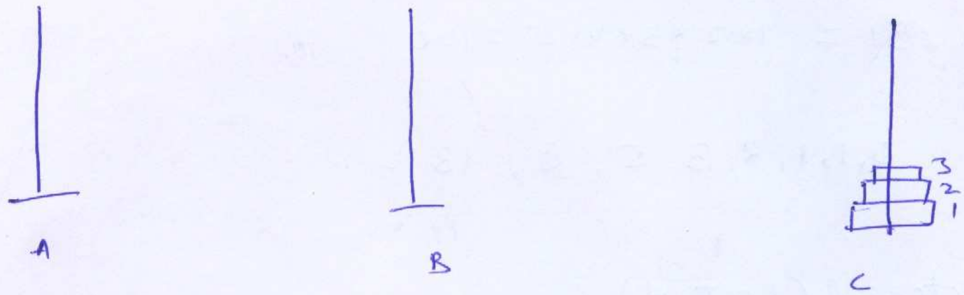
Tower of Hanoi:—

~~It~~ The Tower of Hanoi is a mathmetical game or puzzle. It consist of three rods and a no. of disks of different sizes which can slide onto any rod.

starting goal



Final Goal



mov I more disk 3 to pol C

2 Move disk 2 to Pol B

3 more disk 3 to Pol B

4 more disk 1 to Post C

5 more disk 3 to Post A

6 more disk 2 to post C

7 more disk 3 to post C

Parenthesis checking :—

In a C or higher level language program, stack is used to check the proper opening and closing of Parenthesis. A program Can have ~~follow~~ following braces-

< > , {}, ( ), [ ]

The stack is used to check these all type of Parenthesis.

A Program will be called properly parenthesized if in the Program-

a) There is a closing Parenthesis for every opening Parenthesis.

b) There are equal no. of closing and opening parenthesis.

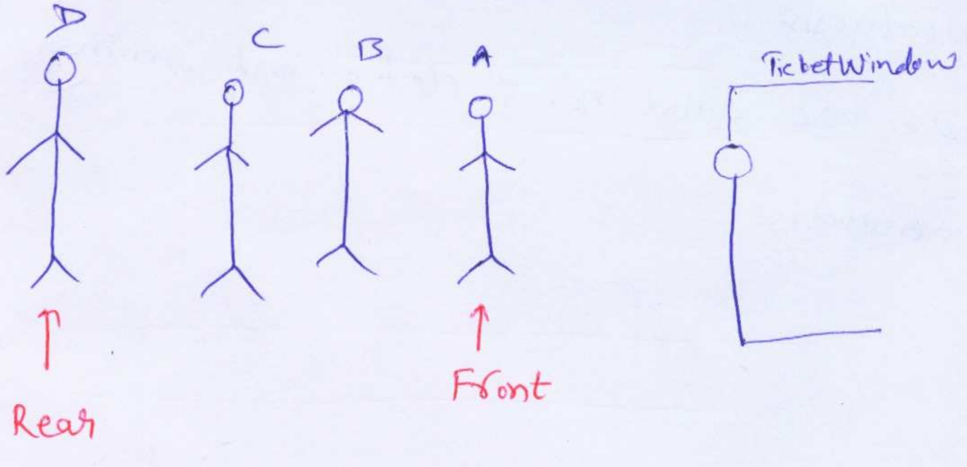
Queues;—

A Q is a linear data structure in which insertion of new element can take place only at ~~an~~ end called Rear and the deletion of existing element can take place only at other end called FRONT.

The queue is work as first In first out (FIFO) order). In FIFO order, the first element added to the queue will be removed first.

Example —

① queue ~~for~~ of people waiting at a ticket window or bus stop.



② time sharing system in Computer science, in which program with the priority from a queue while waiting to be executed.

Type of queue;—

i) linear queue

ii) circular queue

iii) double queue

iv) Priority queue.

queue representation:—

i) By array   ii) By link list

Front=0
Rear =3

0	1	2	3	4	5
A	B	C	D		

Delete.

Front=1
rear=3

0	1	2	3	4	5
	B	C	D		

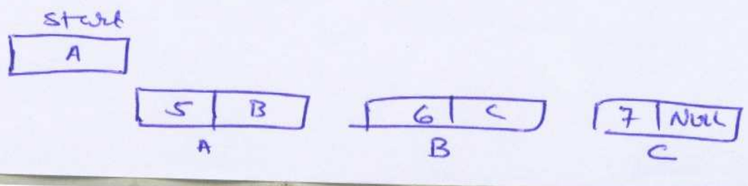
Insert

Front = 1
rear = 4

0	1	2	3	4	5	6
	B	C	D	E	'	

when Q is empty then F=R=-1
Q is overflow when F=R+1
Q is underflow when F=R-1

Link list →



Operation on queue;—

(I) Front   ii) Rear

Algorithm for insertion (rear)

Q IN (Q, maxsize, F, R, item)

(Let Q is array and maxsize represent limit, F is Front and R is rear where item is no. to insert.)

```
Step-1   if F=0 and R=maxsize -1 then
             write overflow and exit          {check for empty Q}
         {End of if}
~~step II.~~   ~~set Q[R] = item~~
stepIII  if F = -1 then                       {check for ~~single~~ empty element in Q}
             Set F=R=0
         Else if R= maxsize -1 then write overflow and return
             set R= R+1
             {End of if stmt}
3            Set Q[R] = item                  {insert no}
4        Exit
```

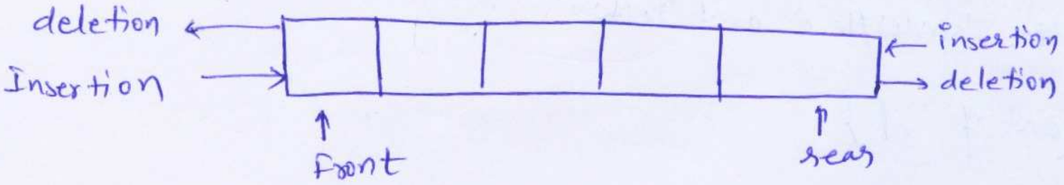
Deletion of elements in Q (Front)

Algo for front

```
Q de( Q, maxsize, F, R)

Step1   if F=-1   then                {check for Q empty}
        write 'underflow' and return
        {end of if}

Step 2  set item = Q[F]

3       if F=R then                   {check for single element in Q}
            set F=R=-1
        Else
            set F=F+1
        {end of If}

4       Exit.
```

Double Ended Queue (Deque)

It is a linear list in which elements can be added or removed from both the ends but not from the middle. it is known as Deque.



There are two type of deque -

i) Input restricted deque :— in this, insertion in such queue can take place at only one end; however, an element can be delected from both ends.

ii) Output restricted deque ;— deletion in such queue can take place at only one end, however, an element can be inserted at both the ends.

Both insertion and deletion can be performed from both the ends.