Lab 3:
→ Working of Queue including insert rear, delete front and display operations.

```
#include <stdio.h>
#include <process.h>
#define QUE_SIZE 3
int item, front=0, rear=-1, q[10];
void Insertrear()
{
  if (rear==QUE_SIZE-1)
  {
    printf("Queue overflow\n");
    return;
  }
 rear=rear+1;
 q[rear]=item;
}
int deletefront()
{
  if (front>rear)
  {
    front=0;
    rear=-1;
    return -1;
  }
 return q[front++];
}
void displayQ()
{ int i;
  if (front>rear)
  {
    printf("Queue is empty\n");
    return;
  }
  printf("Contents of queue\n");
  for(i=front; i<=rear; i++)
  {
    printf("%d\n", q[i]);
  }
```

```
}
void main()
{
  int choice;
  for(;;)
  {
    printf("\n1: Insert rear \n 2: delete front \n 3: display \n
     4: exit \n");
    printf("Enter the choice \n");
    scanf("%d", &choice);
    switch(choice)
    {
    case 1: printf("Enter the item to be inserted \n");
            scanf("%d", &item);
            insertrear();
            break;
    case 2: item = deletefront();
            if(item == -1)
            printf("Queue is empty \n");
            else
            printf("Item deleted = %d \n", item); break;
    case 3: display();
            break;
    default: exit(0);
    }
  }
```

→ Tower of Hanoi

```
#include <stdio.h>
void towers(int n, char src, char temp, char dest)
{
  if(n==1)
  {
    printf("Move disk 1 from %c to %c", src, dest);
    return;
  }
  towers(n-1, src, dest, temp);
  printf("\n Move disk %d from %c to %c \n", n, src, dest);
```

```
towers (n-1, temp, src, dest);
}
void main ()
{
  int n;
  printf ("Enter the number of disks \n");
  scanf ("%d", &n);
  towers (n, 'S', 'T', 'D');
}
```