CODE:(Queue)

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define SIZE 3
int item,front=0,rear=-1,q[10];
void insertrear()
{
	if(rear==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 display()
{
	int i;
	if(front>rear)
	{
		printf("Queue is EMPTY!!\n");
		return;
	}
	printf("Contents of Queue:\n----------\n");
	for(i=front;i<=rear;i++)
	{
		printf("%d\n",q[i]);
	}

}
void main()
{
	int choice;
	while(1)
	{
		printf("\n1 : INSERT \n2 : DELETE\n3 : DISPLAY\n4 : EXIT\n");
		printf("Enter choice:\n");
		scanf("%d",&choice);
		switch(choice)
		{
			case 1: printf("Enter 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);

    }

  }
}
```

OUTPUT:

```
1 : INSERT
2 : DELETE
3 : DISPLAY
4 : EXIT
Enter choice:
3
Queue is EMPTY!!

1 : INSERT
2 : DELETE
3 : DISPLAY
4 : EXIT
Enter choice:
1
Enter item to be inserted:
11
```

```
1 : INSERT
2 : DELETE
3 : DISPLAY
4 : EXIT
Enter choice:
1
Enter item to be inserted:
22

1 : INSERT
2 : DELETE
3 : DISPLAY
4 : EXIT
Enter choice:
3
Contents of Queue:
----------
11
22
```

CODE(TOWER OF HANOI):

```
#include <stdio.h>
#include<conio.h>
void towers(int n,char src,char temp,char dest)
{
	if(n==1)
	{
		printf("Move disk 1 from %c to %c \n",src,dest);
		return;
	}
	towers(n-1,src,dest,temp);
	printf("Move disk %d from %c to %c \n",n,src,dest);
	towers(n-1,temp,src,dest);
}
void main()
{
	int n;
	printf("Enter number of disks:\n");
	scanf("%d",&n);
	towers(n,'S','T','D');
}
```

OUTPUT:

```
Enter number of disks:
2
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
```

```
Enter number of disks:
4
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
Move disk 3 from S to T
Move disk 1 from D to S
Move disk 2 from D to T
Move disk 1 from S to T
Move disk 4 from S to D
Move disk 1 from T to D
Move disk 2 from T to S
Move disk 1 from D to S
Move disk 3 from T to D
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
```