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)
  { print ("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 (i)
  {  printf("\n 1: INSERT \n 2: DELETE \n 3 : DISPLAY
              \n 4: 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);

     }
  }
}
```

# Tower of HANOI

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