```c
#include<stdio.h>
#define MAX_SIZE 10
int Q[MAX_SIZE], FRONT = -1, REAR = -1;

void enqueue(int value)
{
          (REAR >= MAX_SIZE - 1)
        {
                printf("Queue is Full\n");
        }

        {
          ( FRONT == -1 && REAR == -1)
        {
                FRONT = 0;
        }
        REAR += 1;
        Q[REAR] = value;
        }
}

void dequeue()
{
           (FRONT < 0)
        {
                printf("Queue is Empty\n");
        }

        {
                printf("Deleted Element is: %d\n", Q[FRONT]);
                  (FRONT == REAR)
                {
                        FRONT = -1;
                        REAR = -1;
                }

                {
                        FRONT = FRONT + 1;
                }
        }
}

void display()
{
           (REAR >= 0)
        {
                printf("Stack Elements are : ");
                    (int i = REAR; i >= FRONT; i--)
                {
                        printf("%d\t", Q[i]);
                }
                printf("\n");
        }

        {
                printf("QUEUE is empty");
        }
}


int main()
{
```

```
        int count = 0, var, value;
              (count != 1)
        {
                printf(" 1.Enqueue\t 2.Dequeue\t 3.Display\t 4.Exit\n ");
                printf(" Enter Your Choice: ");
                scanf("%d", &var);
                      (var)
                {
                             1: printf("Enter a value: ");
                                scanf("%d",&value);
                                enqueue(value);
                                     ;
                             2: dequeue();
                                     ;
                             3: display();
                                     ;
                             4: count++;
                                printf("Exited...\n");
                                      (0);
                                     ;
                               : printf("invalid Number!!!\n");
                }
        }
              (0);
}
```