Queue Data Structure Implementation using Arrays | Lab 3 | Logic and Program | Tamil | MPR

```c
#include <stdio.h>

#define MAX 100

int queue[MAX], i;
static int front = -1,back=-1;

void add(int);
void rem();
void display();

int main() {
    int choice, element;
    while(1) {
    printf("Enter the choice : \n");
    printf(" 1. Add an element to the queue \n 2. Remove an element to the queue \n 3. Display all elements of the queue \n 4. Exit \n");
    scanf("%d",&choice);
    switch(choice) {
    case 1:
      printf("Enter the element to be added: \n");
      scanf("%d",&element);
      add(element);
      break;
    case 2:
      rem();
      break;
    case 3:
      display();
      break;
    case 4:
      printf("Exiting Program....\n");
```

```c
            printf("Done! Good Bye!!!");
            return 0;
        default:
            printf("Invalid Input....Try Again!!!\n");}
    }
        }


void add(int n) {
   if(front==-1){
       front = 0;
   }
  if(back==MAX-1) {
   printf("The queue is full!");
   return;
  }
  else{
  back++;
  queue[back] = n;
  printf("%d has been added to the Queue!\n",n);
}}


void rem() {
   if(front==-1) {
       printf("Queue has no elements to remove!\n");
   }
   else{
    int rmEle = queue[front];
    front++;
    printf("%d is removed from the Queue!\n",rmEle);
}}


void display() {
    if(front==-1) {
       printf("Queue has no elements to display!\n");
   }
    else{
    printf("The Queue Elements are: ");
    for (i=front;i<=back;i++) {
      printf(" %d ",queue[i]);
```

```
        }
        printf("\n");
    }
}
```