Implementation of Queue and Circular Queue using arrays.

```
#include <stdio.h>
#define max 10
int lqueue[max],cqueue[max],lfront=-1,lrear=-1,cfront=-1,crear=-1;
int linsert(){
    if (lrear==max-1){
        printf("queue overflow\n");
        return 0;
    }
    int num;
    printf("enter the number: ");
    scanf("%d",&num);
    lqueue[++lrear]=num;
    printf("number inserted\n");
    if (lfront==-1){
        lfront=0;
    }
    return 1;
}
int ldelete(){
    if (lfront==-1){
        printf("queue underflow\n");
        return 0;
    }
    int num=lqueue[lfront];
    printf("deleted item: %d",num);
    lfront++;
    if(lfront>lrear){
        lfront=lrear=-1;

    }
    return 1;
}
int ldisplay(){
    if(lfront==-1){
        printf("queue empty\n");
        return 0;
    }
    printf("queue elements: ");
    for(int i=lfront;i<=lrear;i++){
        printf("%d ",lqueue[i]);
    }
    return 1;
}
int cinsert(){
    int num;
    if (cfront==(crear+1)%max){
        printf("queue overflow\n");
        return 0;
    }
    printf("enter the number: ");
    scanf("%d",&num);
    printf("item added\n");
    if (cfront==-1){
        cfront=crear=0;
    }
    else{
        crear=(crear+1)%max;
    }
```

```
    cqueue[crear]=num;
    return 1;

}
int cdelete(){
    if (cfront==-1){
        printf("queue underflow \n");
        return 0;
    }
    int num=cqueue[cfront];
    printf("deleted item: %d",num);
    if (cfront==crear){
        cfront=crear=-1;
    }
    else{
        cfront=(cfront+1)%max;
    }
    return 1;
}
int cdisplay(){
    if (cfront==-1){
        printf("queue empty\n");
        return 0;
    }
    printf("queue elements: ");

    if (cfront<=crear){
        for (int i=cfront;i<=crear;i++){
            printf("%d ",cqueue[i]);
        }

    }
    else{
        for(int i=cfront;i<max;i++){
            printf("%d ",cqueue[i]);
        }
        for(int i=0;i<=crear;i++){
            printf("%d ",cqueue[i]);
        }
    }
}

int main() {
    int mch,sch;
    do{
        printf("MENU\n 1.linear queue\n 2.circular queue\n 3.exit\n");
        printf("enter your main choice: ");
        scanf("%d",&mch);
        if(mch==1 || mch==2){
            do{
            printf("\nOPERATIONS\n 1.insert\n 2.delete\n 3.display\n 4.exit sub menu\n");
            printf("enter your operation: ");
            scanf("%d",&sch);
            switch(mch){
                case 1:
                  switch(sch){
                     case 1:
                        linsert();
                        break;
                     case 2:
```

```
                    ldelete();
                    break;
                  case 3:
                    ldisplay();
                    break;
                  case 4:
                    break;
                  default:
                  printf("invalid choice:\n");

               }
               break;
             case 2:
              switch(sch){
                  case 1:
                    cinsert();
                    break;
                  case 2:
                    cdelete();
                    break;
                  case 3:
                    cdisplay();
                    break;
                  case 4:
                    break;
                  default:
                  printf("invalid choice:\n");

               }
               break;



          }
          }while(sch!=4);
       }
       else if(mch!=3){
          printf("invalid main choice\n");
       }


   }while(mch!=3);


   return 0;
}
```

Output

```
MENU
 1.linear queue
 2.circular queue
 3.exit
enter your main choice: 1

OPERATIONS
 1.insert
 2.delete
```

```
 3.display
 4.exit sub menu
enter your operation: 1
enter the number: 10
number inserted

OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 1
enter the number: 20
number inserted

OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 1
enter the number: 30
number inserted

OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 3
queue elements: 10 20 30
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 2
deleted item: 10
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 3
queue elements: 20 30
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 4
MENU
 1.linear queue
 2.circular queue
 3.exit
enter your main choice: 2

OPERATIONS
 1.insert
 2.delete
```

```
 3.display
 4.exit sub menu
enter your operation: 1
enter the number: 10
item added

OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 1
enter the number: 20
item added

OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 3
queue elements: 10 20
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 2
deleted item: 10
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 3
queue elements: 20
OPERATIONS
 1.insert
 2.delete
 3.display
 4.exit sub menu
enter your operation: 4
MENU
 1.linear queue
 2.circular queue
 3.exit
enter your main choice: 3
```