Practice program

```
# include <stdio.h>
# define qsize 5
  int f=0, r=-1, ch;
  int item, q[10];

  int isfull()
   {
     return (r==qsize-1);
   }
 int isempty()
   {
     return (f>r);
   }
 void insert_rear()
    {
       if (isfull())
          {
            printf("queue overflow \n");
            return;
          }
       r=r+1;
       q[r] = item;
   }
  void delete front(){

     if (isempty())
         {
           Printf("queue underflow \n");
           return;
         }
```

```
printf ("item deleted is %d\n", q[(f)++]);
    if (f>r)
      {
        f=0;
        r=-1;
      }
  }
void insert_front()
{
   if (f != 0)
   {
     f = f-1;
    q[f] = item;
     return;
   }
   else if (f==0 && r==-1)
   {
     q[++(r)] = item;
     return;
   }
   else
     printf ("insertion not possible \n");
}
void delete_rear () {
    if (is empty())
      {
        printf ("queue is empty \n");
        return;
      }
  printf ("item deleted is %d \n", q[(r)--]);
  if (f>r) {
     f=0;
     r=-1;
  }
}
```

```
void display() {
    int i;
    if (is empty())
      {
        printf ("queue is empty \n");
        return;
      }
    printf ("item deleted is %d\n", q[(r)--]);
    if (f>r)
      {
        f=0;
        r=-1;
      }
}
void display() {
    int i;
    if (is empty()) {
        printf("queue empty \n");
        return;
    }
    for (i = f; i <= r; i++)
        printf ("%d\n", q[i]);
}
void main(){
    int n= 1;
    while(n != 0) {
        f = 0;
        r = -1
    }
}

void main() {
    int n=1;
    while (n!=0) {
        printf ("1. insert-rear\n 2. insert-front\n 3. delete-rear\n
        4. delete-front \n 5. display\n 6. exit\n");
```

```
printf ("enter choice \n");
scanf ("%d", &choice);
switch (choice)
 {
   case 1: printf ("enter the item \n");
           scanf ("%d", &item);
           insert_rear ();
           break;
   case 2: printf ("enter the item \n");
           scanf ("%d", &item);
           insert_front ();
           break;
   case 3: delete_rear ();
           break;
   case 4: delete_front ();
           break;
   case 5: display ();
           break;
   case 6: n=0;
           break;
   default: exit (0);
  }
 }
}
```