```
//circular queue
#include<stdio.h>
#include<conio.h>
# define sz 5

struct queue
{
  int que[sz];
  int f,r;
};

int empty(struct queue *p)
{
  if(p->f==p->r)
    return 1;
  else
    return 0;
}

void remoove(struct queue *p)
{
 if(empty(p))
   printf("Queue Oveflow...");
 else
 {
   if(p->f==sz-1)
    p->f=0;
   else
    (p->f)++;

  printf("Removed element : %d",p->que[p->f]);
 }
}

void insert(struct queue *p)
{
 int x;
 printf("Enter element:");
 scanf("%d",&x);
 if(p->r==sz-1)
   p->r=0;
 else
```

```
  (p->r)++;
 if(p->r==p->f)
 {
  printf("Overflow..");
  (p->r)--;
 }
 else
 p->que[p->r]=x;
}

/*void display(struct queue q)
{

   if(empty(&q))
     printf("QUEUE is empty");
   else
   {
     printf("Queue:");
     while(q.r!=q.f)
     {
      if(q.f==sz-1)
        q.f=0;
      else
        (q.f)++;
     printf("%d",q.que[q.f]);
     }
}

} */
void main()
{
  struct queue q;
    int ch;
    q.f=q.r=sz-1;
  clrscr();

while(1)
{
  printf("\n\t Queue:\n\t1.Insert\n\t2.Remove\n\t4.Exit\n\tEnter your chioce:");
  scanf("%d",&ch);
  switch(ch)
  {
   case 1:insert(&q);
```

```
        break;
   case 2:remoove(&q);
        break;
  /* case 3:display(q);
        break; */
   case 4:exit(0);
  }
}
 getch();
}
```