```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define qsize 3
int f=0, r=-1, ch;
int item, q[10];

int isfull()
  {
    return (r==qsize-1)? 1:0;
  }

int isempty(int f, int r)
  {
    return (f>r)? 1:0;
  }
void insert_rear()
  {
    if(isfull(r))
      {
        printf("Queue overflow \n");
        return;
      }
```

```
    r=r+1;
    q[r]=item;
}

void delete_front()
{
    if(isempty(f,r))
    {
        printf("queue empty \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");
}
```

```
void delete_queue()
{
  if( isempty(f, r))
    {
      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( isempty(f, r))
    {
      printf("queue empty");
      return;
    }
  for(i=f; i<=r; i++)
    printf("%d \n", q[i]);
}
```

```
void main()
  {
    for(;;)
    {
      printf("Enter a choice");
      scanf("%d" &ch);
      switch(ch)
        {
          case 1: printf("Enter item to be inserted");
                  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;
          default: exit(0);
        }
    }
}
```