```
#include<stdio.h>
int rear = -1 ,front = -1;
int q[30];
int stack[100],top;
void push(int data);
int pop();
void enqueue(int q[],int n,int data);
int dequeue(int q[],int n);
void display(int q[],int n);
void reverse(int q[],int n);
int main(){
     int choice,n,data;
     char ch;
     printf("enter the size of the queue");
     scanf("%d",&n);
     while(1){
          printf("\n1. Enqueue \n2. Dequeue \n3.
Reverse\n4.Display ");
          scanf("%d",&choice);
          switch(choice){
               case 1 : printf("Enter the Number");
                     scanf("%d",&data);
                     enqueue(q,n,data);
                        break;
               case 2 : dequeue(q,n);
                        break;
            case 3 : reverse(q,n);
                     break;
               case 4 : display(q,n);
                        break;
                  default: printf("error");
                      break;
               }
            printf("\nDo you want to continue y/n ");
            scanf(" %ch",&ch);
          if (ch=='n'){
              break;
          }

            }
     return 0;
}
void enqueue(int q[],int n,int data){
     if(rear>n){
```

```
          printf("\nOverflow");
     }
     else {
          if(front==-1&&rear ==-1){
               front = 0;
          }
          rear +=1;
        q[rear]=data;
     }
}
void display(int q[],int n){
     if(front ==-1 ){
          printf("\nempty");
     }
     else {
          for(int i=front;i<=rear;i++){
               printf("%d\t",q[i]);
          }
     }
}
int dequeue(int q[],int n){
     if(front== -1){
          printf("\nunderflow");
     }
     else{
        int item = q[front];
          if(front==rear){
               front=rear =-1;
           }
           else{
               front++;
           }
         return item;
     }
}
void push(int data)
{
if(top==99)
     {
      printf("\nStack Overflow !!!!\n");

     }
else
     {
```

```
     top=top+1;
     stack[top]=data;
}
}
int pop(){
if(top==-1)
{
     printf("\nUnderflow !!!!\n");
}
else
{
    int item = stack[top];
    top = top -1 ;
     return item;
}
}
void reverse(int q[],int n){
     if(front ==-1 ){
          printf("\nempty");
     }
     else {
          for(int i=front;i<=rear;i++){
               push(dequeue(q,n));
          }
        front=rear=-1;
          for(int i=top-1;i>=0;i--){
               enqueue(q,n,pop());
          }
     }
}
```