Menu driven

#include<stdio.h>

#include<stdlib.h>

#define SIZE 5

#define MAX 100

int x, front=-1,rear=-1;

int q[SIZE];

void enqueue1();

int dequeue();

void display();

///////////////////////////////////////

int data[MAX];

int rear1=0,front1=0;

void enqueue(int);

int dequeueFront();

int dequeueRear();

void enqueueRear(int);

void enqueueFront(int);

void display1();

int main()

{

int z;

int c;

int a,b,val;

while(1)

{

printf("\n1.Circular Queue\n");

printf("2.Double Ended Queue\n");

printf("Enter a choice:");

scanf("%d",&z);

switch(z)

{

case 1:while(1)

{

printf("\n1.Insert");

printf("\n2.Delete");

printf("\n3.Display");

printf("\n4.Exit\n");

printf("Enter a choice:");

scanf("%d",&c);

switch(c)

{

case 1:enqueue1();

break;

case 2:dequeue();

break;

case 3:display();

break;

case 4:exit(1);

default:printf("Invalid choice");

break;

}

}

case 2:while(1)

{

printf("\n1.Input-restricted dequeue");

printf("\n2.Output-restricted dequeue\n");

printf("Enter a choice:");

scanf("%d",&a);

switch(a)

{

case 1:

while(1)

{

printf("\n1.Insert");

printf("\n2.Delete from Rear");

printf("\n3.Delete from Front");

printf("\n4.Display\n");

printf("Enter a choice:");

scanf("%d",&b);

switch(b)

{

case 1:enqueueRear(val);

display1();

break;

case 2:val=dequeueRear();

display1();

break;

case 3:val=dequeueFront();

display1();

break;

case 4:display1();

break;

default:printf("Invalid choice");

break;

}

}

case 2:

while(1)

{

printf("\n1.Insert at Rear");

printf("\n2.Insert at Front");

printf("\n3.Delete ");

printf("\n4.Display\n");

printf("Enter a choice:");

scanf("%d",&b);

switch(b)

{

case 1:enqueueRear(val);

display1();

break;

case 2:enqueueFront(val);

display1();

break;

case 3:val=dequeueFront();

display1();

break;

case 4:display1();

break;

default:printf("Invalid choice");

break;

}

}

}

}

}

}

}

void enqueue1()

{

if(((rear==SIZE-1)&&(front=0))||(front>0&&(rear==front-1)))

{

printf("\nQueue full\n");

}

else

{

printf("Enter element to insert:");

scanf("%d",&x);

if((front>0)&&(rear==SIZE-1))

{

rear=0;

q[rear]=x;

}

else

{

if((front==0&&rear==-1)||(rear!=front-1))

{

q[++rear]=x;

}

}

}

}

int dequeue()

{

if(front==-1)

{

printf("\nQueue empty\n");

}

else if(front==rear)

{

x=q[front];

rear=-1;

front=0;

}

else if(front==SIZE-1)

{

x=q[front];

front=0;

}

else

{

x=q[front++];

}

printf("\nDeleted item is %d\n",x);

}

void display()

{

int i,j;

if(front==0&&rear==-1)

{

printf("\nQueue empty\n");

}

else if(front>rear)

{

for(i=0;i<=rear;i++)

{

printf("\t%d",q[i]);

}

for(j=front;j<=SIZE-1;j++)

{

printf("\t%d",q[j]);

}

printf("\nRear is at %d",q[rear]);

printf("\nFront is at %d",q[front]);

}

else

{

for(i=front;i<=rear;i++)

{

printf("\t%d",q[i]);

}

printf("\nRear is at %d",q[i]);

printf("\nFront is at %d",q[i]);

}

printf("\n");

}

///////////////////////////////////////////////////////////////////////////

void enqueueRear(int val)

{

if(front1==MAX/2)

{

printf("\nQueue Full\n");

}

else

{

printf("Enter number to insert:");

scanf("%d",&val);

data[front1]=val;

front1++;

}

}

void enqueueFront(int val)

{

if(front1==MAX/2)

{

printf("\nQueue Full\n");

}

else

{

printf("Enter number to insert:");

scanf("%d",&val);

rear1--;

data[rear1]=val;

}

}

int dequeueRear()

{

int x;

if(front1==rear1)

{

printf("\nQueue Empty\n");

}

front1--;

x=data[front1+1];

return x;

}

int dequeueFront()

{

int x;

if(front1==rear1)

{

printf("\nQueue Full\n");

}

rear1++;

x=data[rear1-1];

return x;

}

void display1()

{

int i;

if(front1==rear1)

{

printf("\nQueue Empty\n");

}

else

{

printf("Elements in double ended queue are\n");

for(i=rear1;i<front1;i++)

{

printf("%d\t",data[i]);

}

}

}

**Output**



















