//Dsa 3 queue

#include<iostream>

#define Max 5

using namespace std;

class Queue

{

int Q[Max],Front,Rear;

public:

Queue(){ Front=0;Rear=-1;

for(int i=0;i<Max-1;i++) Q[i]=-1;

}

int is\_Queue\_Full();

int is\_Queue\_Empty();

int eQueue(int);

int dQueue(int&);

void Display();

};

int Queue::is\_Queue\_Full()

{

if(Rear!=-1 && Front==(Rear+1)%Max)

return 1;

return 0;

}

int Queue::is\_Queue\_Empty()

{

if(Rear==-1)

return 1;

return 0;

}

int Queue::eQueue(int Data)

{

if(is\_Queue\_Full())

return false;

Rear=(Rear+1)%Max;

Q[Rear]=Data;

return true;

}

int Queue::dQueue(int& Data)

{

if(!is\_Queue\_Empty())

{

Data=Q[Front];

Q[Front]=-1;

Front=(Front+1)%Max;

if(Front==(Rear+1)%Max)

{ Front=0; Rear=-1; }

return true;

}

return false;

}

void Queue:: Display()

{

for(int i=0;i<Max;i++)

cout<<" "<<Q[i];

}

int main()

{

Queue QU;

int Element,Choice;

char Answer;

do

{

cout<<"\n1:Entry\t2:Exit\n Enter your Choice: ";

cin>>Choice;

switch(Choice)

{

case 1:

cout<<"Enter Data";

cin>>Element;

if(QU.eQueue(Element))

cout<<"\n Entered Successfully";

else

cout<<"\nQueue Full Can not Enter";

break;

case 2:

if(QU.dQueue(Element))

cout<<"\nExit Element "<<Element;

else

cout<<"Queue is Empty can not Remove";

break;

}//switch end

cout<<"\nContinue(y/n)...";

cin>>Answer;

}while(Answer=='y'||Answer=='Y');//while end

return 1;

}//main end