/\*

\* LinearQueue.cpp

\*

\* Created on: Sep 9, 2020

\* Author: hp

\*/

#include<iostream>

using namespace std;

//structure declaration

struct Patient

{

int patien\_id;

string name;

string city;

string disease;

};

//class

class Queue

{

private:

Patient p[20];

int front,rear;

public:

Queue(){

front=-1;

rear=-1;

}

//methods declaration

void enqueue(int);

void dequeue(int);

void display(int);

};

//definition of enqueue

void Queue::enqueue(int n)

{

if(rear==n-1)

cout<<"Overflow Condition!!!!"<<endl;

else

{

if(front==-1)

front=0;

rear++;

cout<<"Enter details"<<endl;

cout<<"Id:";

cin>>p[rear].patien\_id;

cin.ignore();

cout<<"Name:";

getline(cin,p[rear].name);

cout<<"City:";

getline(cin,p[rear].city);

cout<<"Disease:";

getline(cin,p[rear].disease);

cout<<"Record added"<<endl;

}

}

//definition of dequeue

void Queue::dequeue(int n)

{

if(front==-1||front==rear+1)

cout<<"Underflow condition!!!!"<<endl;

else

{

front++;

cout<<"Record deleted"<<endl;

}

}

//definition of display

void Queue::display(int n)

{

if(front==-1||front==rear+1)

cout<<"Empty queue"<<endl;

else

{

cout<<"Id\tName\tCity\tDisease"<<endl;

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

cout<<p[i].patien\_id<<"\t"<<p[i].name<<"\t"<<p[i].city<<"\t"<<p[i].disease<<endl;

}

}

//driver function

int main()

{

Queue queue;

int n;

cout<<"Enter number of patients:";

cin>>n;

int choice;

do

{

cout<<"--------------------"<<endl;

cout<<"1:Insert new record"<<endl<<"2:Delete a record"<<endl<<"3:Display records"<<endl;

cout<<"Enter choice:";

cin>>choice;

cout<<"--------------------"<<endl;

switch(choice)

{

case 1:

queue.enqueue(n);

break;

case 2:

queue.dequeue(n);

break;

case 3:

queue.display(n);

break;

case 4:

cout<<"Thank you...";

break;

}

}while(choice!=4);

return 0;

}