#include<iostream>

using namespace std;

class stack

{

private:

int arr[20];

int e,s,top;

public:

stack()

{

top=10;

e=-1;

s=-1;

}

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

bool is\_full()

{

if(e==20)

return true;

else

return false;

}

bool is\_empty()

{

if(e==-1)

return true;

else

return false;

}

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

void enqueue(int y)

{

if(is\_full())

{

cout<<"the queue is full"<<endl;

}

else

{

arr[++e]=y;

cout<<"the value "<<y<<" is entered in the queue at index"<<"["<<e<<"]"<<endl;

}

}

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

int dequeue()

{

if(is\_empty())

{

cout<<"the queue is empty"<<endl;

}

else

{

int z;

z=arr[e--];

return z;

}

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

{

arr[i]=arr[i+1];

}

}

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

void push(int var)

{

if(top==20)

cout<<"overflow"<<endl;

else

{

top++;

arr[top]=var;

cout<<arr[top]<<"at index "<<"["<<top<<"]"<<endl;

}

}

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

void pop()

{

if(top==10)

cout<<"previous full"<<endl;

else

{

arr[top]='\0';

cout<<top--<<endl;

}

}

};

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

void main()

{

stack c,s;

c.enqueue(1);

c.enqueue(2);

c.enqueue(3);

c.enqueue(4);

c.enqueue(5);

c.enqueue(6);

c.enqueue(7);

c.enqueue(8);

c.enqueue(9);

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

cout<<"\n\n";

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

s.push(11);

s.push(12);

s.push(13);

s.push(14);

s.push(15);

s.push(16);

s.push(17);

s.push(18);

s.push(19);

s.push(20);

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

cout<<"\n\n";

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

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

cout<<c.dequeue()<<endl;

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

cout<<"\n\n";

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

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

s.pop();

system("pause");

}