STACKS AND QUEUES

import java.util.Arrays;  
import java.util.Stack;  
  
class circularqueue{  
 public int[] data;  
 public static int *default\_size*=10;  
 public circularqueue(){  
 this(*default\_size*);  
 }  
 public circularqueue(int size){  
 this.data=new int[size];  
 }  
 int end=0;  
 int front=0;  
 int size=0;  
 public boolean isfull(){  
 return size==data.length;  
 }  
 public boolean isempty(){  
 return size==0;  
 }  
 public boolean insert(int item){  
 if(isfull()){  
 return false;  
 }  
 data[end++]=item;  
  
 end=end%data.length;  
 size++;  
  
 return true;  
 }  
 public int remove(){  
 if(isempty()){  
 System.*out*.println("enpty");  
 return 0;  
 }  
 int num=data[front++];  
  
 front=front%data.length;  
 size--;  
 return num;  
 }  
 public void display(){  
 if(isempty()){  
 System.*out*.println("empty");  
 return;  
 }  
 int i=front;  
 do{  
 System.*out*.print(data[i]+"-->");  
  
 i++;  
 i=i%data.length;  
 }while(i!=end);  
 }  
}  
  
class Customqueue{  
 public int[] data;  
 public static int *default\_size*=10;  
 public Customqueue(){  
 this(*default\_size*);  
 }  
 public Customqueue(int size){  
 this.data=new int[size];  
 }  
 int end=0;  
 public boolean isfull(){  
 return end==data.length;  
 }  
 public boolean isempty(){  
 return end==0;  
 }  
 public boolean insert(int item){  
 if(isfull()){  
 return false;  
 }  
 data[end++]=item;  
 return true;  
 }  
 public int remove(){  
 if(isempty()){  
 System.*out*.println("enpty");  
 return 0;  
 }  
 int num=data[0];  
 for(int i=1;i<end;i++){  
 data[i-1]=data[i];  
 }  
 end--;  
 return num;  
 }  
 public void display(){  
 for(int i=0;i<end;i++){  
 System.*out*.print(data[i]+"-->");  
 }  
 System.*out*.print("END");  
 }  
  
}  
class Customstack{  
 public int[] data;  
 public static int *default\_size*=10;  
 public Customstack(){  
 this(*default\_size*);  
 }  
 public Customstack(int size){  
 this.data=new int[size];  
 }  
 int ptr=-1;  
  
 public boolean push(int num){  
 if(isfull()){  
 int[] temp=new int[data.length\*2];  
 for(int i=0;i<data.length;i++){  
 temp[i]=data[i];  
 }  
 data=temp;  
 }  
 ptr++;  
 data[ptr]=num;  
 return true;  
 }  
 public boolean isfull(){  
 return ptr==data.length-1;  
 }  
 public boolean isempty(){  
 return ptr==-1;  
 }  
 public int pop(){  
 if(isempty()){  
 System.*out*.println("stack is empty");  
 return -1;  
 }  
 int num=data[ptr];  
 ptr--;  
 return num;  
 }  
 public void display(){  
 for(int i=0;i<data.length;i++){  
 System.*out*.print(data[i]+"-->");  
 }  
 System.*out*.print("END");  
 }  
  
}  
public class Main {  
  
  
 public static void main(String[] args) {  
// Customstack s=new Customstack(5);  
// s.push(11);  
// s.push(15);  
// s.push(14);  
// s.push(13);  
// s.push(12);  
// s.push(12);  
// s.display();  
  
// System.out.println(s.pop());  
// System.out.println(s.pop());  
// System.out.println(s.pop());  
// System.out.println(s.pop());  
// System.out.println(s.pop());  
// System.out.println(s.pop());  
 circularqueue q=new circularqueue(5);  
 q.insert(33);  
 q.insert(34);  
 q.insert(35);  
 q.insert(36);  
 q.insert(37);  
 q.display();  
 System.*out*.println(q.remove());  
// System.out.println(q.remove());  
 q.insert(39);  
  
 q.display();  
 }  
}