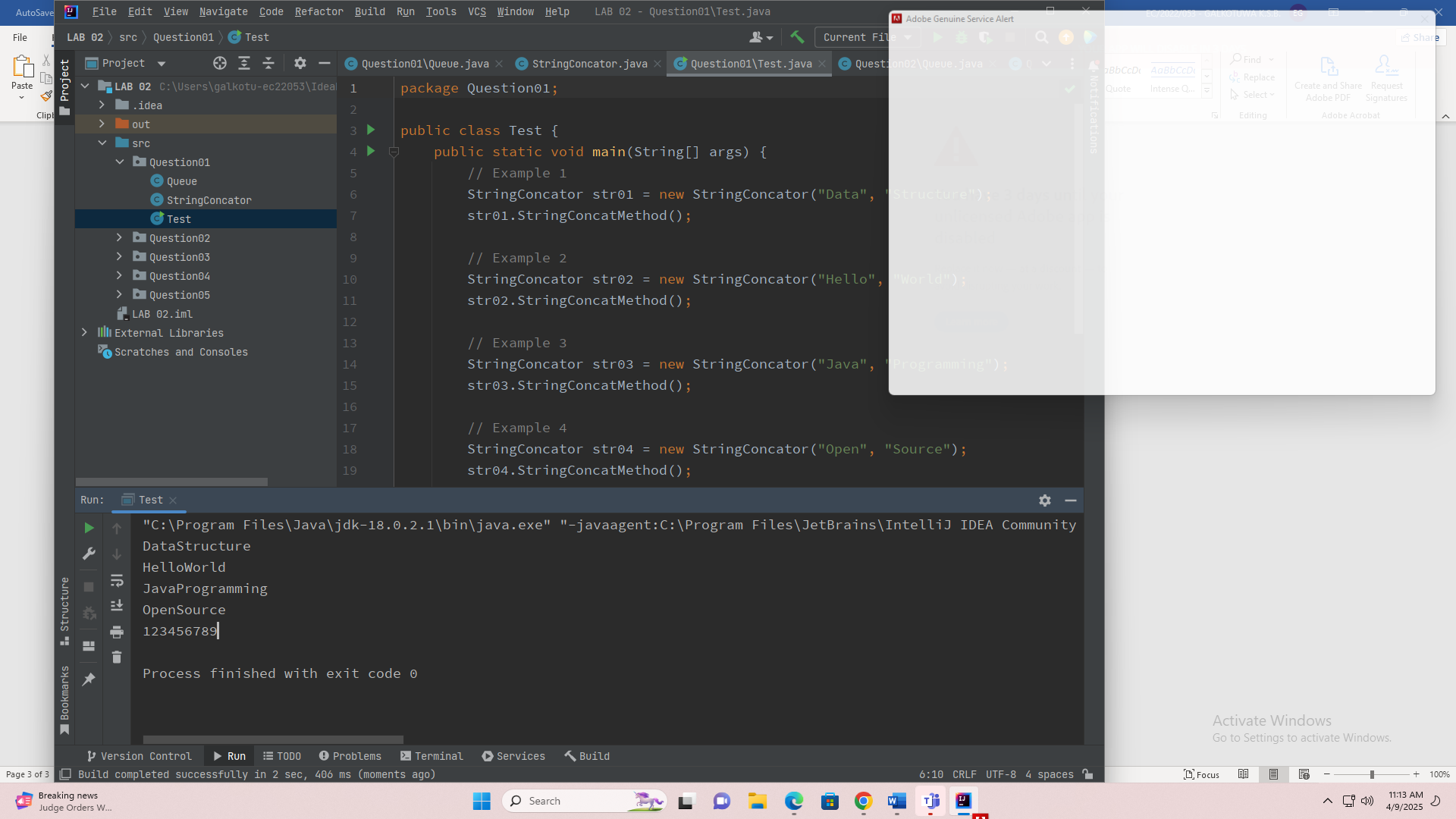
Question 01.

package Question01;  
public class Queue {  
 int front;  
 int count;  
 char[] queue;  
 int maxSize;  
 int rear;  
 public Queue(int maxSize) {  
 this.front = 0;  
 this.count = 0;  
 this.queue = new char[maxSize];  
 this.maxSize = maxSize;  
 this.rear = -1;  
 }  
 boolean IsQueueEmpty(){  
 if (rear<front)  
 return true;  
 else  
 return false;  
 }  
 boolean IsQueueFull() {  
 if (rear == maxSize - 1) {  
 return true;  
 }  
 return false;  
 }  
 void Append(char item) {  
 if (IsQueueFull()) {  
 System.*out*.printf("\nQueue is full\n");  
 } else {  
 queue[++rear] = (char) item;  
 count++;  
 }  
 }  
 char Serve() {  
 if (IsQueueEmpty()) {  
 System.*out*.printf("\nQueue is empty\n");  
 return 0;  
 }  
 else {  
 char x = queue[front++];  
 count--;  
 return x;  
 }  
 }  
}

package Question01;  
import java.util.Arrays;  
public class StringConcator {  
  
 String str01;  
 String str02;  
  
 public StringConcator(String str01, String str02) {  
 this.str01 = str01;  
 this.str02 = str02;  
 }  
  
 public String StringConcatMethod() {  
 Queue q4str01 = new Queue(str01.length());  
 Queue q4str02 = new Queue(str02.length());  
 Queue q4all = new Queue(q4str01.maxSize + q4str02.maxSize);  
  
 while (q4all.IsQueueEmpty()){  
 char[] q1charArray =str01.toCharArray();  
 char[] q2charArray =str02.toCharArray();  
 for (char i : q1charArray) {  
 q4all.Append(i);  
 }  
 for (char i2:q2charArray) {  
 q4all.Append(i2);  
  
 }  
 }  
 System.*out*.println(q4all.queue);  
 return Arrays.*toString*(q4all.queue);  
  
 }  
}

package Question01;  
  
public class Test {  
 public static void main(String[] args) {  
 // Example 1  
 StringConcator str01 = new StringConcator("Data", "Structure");  
 str01.StringConcatMethod();  
  
 // Example 2  
 StringConcator str02 = new StringConcator("Hello", "World");  
 str02.StringConcatMethod();  
  
 // Example 3  
 StringConcator str03 = new StringConcator("Java", "Programming");  
 str03.StringConcatMethod();  
  
 // Example 4  
 StringConcator str04 = new StringConcator("Open", "Source");  
 str04.StringConcatMethod();  
  
 // Example 5  
 StringConcator str05 = new StringConcator("12345", "6789");  
 str05.StringConcatMethod();  
  
 }  
}



Question 02.

package Question02;  
public class Queue {  
 int front;  
 int count;  
 char[] queue;  
 int maxSize;  
 int rear;  
 public Queue(int maxSize) {  
 this.front = 0;  
 this.count = 0;  
 this.queue = new char[maxSize];  
 this.maxSize = maxSize;  
 this.rear = -1;  
 }  
 boolean IsQueueEmpty(){  
 if (rear<front)  
 return true;  
 else  
 return false;  
 }  
 boolean IsQueueFull() {  
 if (rear == maxSize - 1) {  
 return true;  
 }  
 return false;  
 }  
 void Append(char item) {  
 if (IsQueueFull()) {  
 System.*out*.printf("\nQueue is full\n");  
 } else {  
 queue[++rear] = (char) item;  
 count++;  
 }  
 }  
 char Serve() {  
 if (IsQueueEmpty()) {  
 System.*out*.printf("\nQueue is empty\n");  
 return 0;  
 }  
 else {  
 char x = queue[front++];  
 count--;  
 return x;  
 }  
 }  
}

package Question02;  
public class NumberDevider {  
 int numberSize ;  
 int Number;  
 int midValue;  
 public NumberDevider(int number) {  
 char[] digitList = Integer.*toString*(number).toCharArray();  
 this.Number = number;  
 this.numberSize = digitList.length;  
 this.midValue = numberSize/2;  
 }  
 public int NumberDeviderMethod(){  
 String Numb = Integer.*toString*(Number);  
 char[] digitList = Numb.toCharArray();  
 Queue mainQueueHoldAllNumbers = new Queue(numberSize);  
 while (mainQueueHoldAllNumbers.IsQueueEmpty()) {  
 for (int i = midValue; i < digitList.length; i++) {  
 mainQueueHoldAllNumbers.Append(digitList[i]);  
 }  
 for (int i = 0; i < midValue; i++) {  
 mainQueueHoldAllNumbers.Append(digitList[i]);  
 }  
 }  
 System.*out*.println(mainQueueHoldAllNumbers.queue);  
  
 return Integer.*parseInt*(String.*valueOf*(mainQueueHoldAllNumbers.queue));  
 }  
  
}

package Question02;  
  
public class Test {  
 public static void main(String[] args) {  
 NumberDevider nbdev = new NumberDevider(12345678);  
 nbdev.NumberDeviderMethod();  
  
 NumberDevider nbdev2 = new NumberDevider(7);  
 nbdev2.NumberDeviderMethod();  
 NumberDevider nbdev3 = new NumberDevider(0);  
 nbdev3.NumberDeviderMethod();  
 NumberDevider nbdev5 = new NumberDevider(987654321);  
 nbdev5.NumberDeviderMethod();  
 NumberDevider nbdev6 = new NumberDevider(11111111);  
 nbdev6.NumberDeviderMethod();  
 NumberDevider nbdev7 = new NumberDevider(12321);  
 nbdev7.NumberDeviderMethod();  
 }  
}

