INFORMATICS INSTITUTE OF TECHNOLOGY

Informatics Inststute of Technology

Arificial Intellignece & Data Science

Name : A.M Sanduhske De Alwis

IIT number : 20200244

RGU number : 2117852

Module CM1602 Data Structures and algorithms for Artifiacial Intelligence

Assignement : Individual Coursework

Module leader : Ragu Sivaraman

Tutorial leader : Ragu Sivaraman

Handout Date : 7th of March 2022

Submission date : 19th of April 2022

Acknowledge

I would like to take this opportunity to thank all the IIT staff for giving us the opportunity to follow the degree artificial intelligence & data science and also l specially thank my lecturer and tutor Mr. Ragu Sivaraman for giving me the knowledge to do this report. Rather than the academic knowledge I got a lot of information while gathering information. So, at last, I would like to take the opportunity to thank for all the feedbacks that you provided for me and it really helped me to correct all our mistakes. Thank you

Contents

Task 1

I. Select a Data Structure for this scenario. Justify your answer

- Using queues with linked list data structure, the loan system application was implemented

Reasons to use queues with Linked List :

- Queues can be very useful data structures in cases like when only insertion and deletion happens or where there are rare modifications
- Linked lists are easier to implement if you have a structured memory allocation which means it doesn't need to have a size limit when storing data in the memory.
- It can also create a queue that can enlarge and contract. There's no capability of overflowing the heap until the memory is exhausted.
- Therefore with the use of queues with linked list it will be easy to add customer details one after another.

II. Implement the Data Structure

```
class loanSystem{
    QNode frontNode, rearNode;

    public loanSystem(){
        this.frontNode = this.rearNode = null;
    }
    //Method to add an key to the queue
    void enqueue(int NIC, int accountNumber, String loanType,
                 String reasonOfLoanApplication, String
descriptionOfCollaterals){

        //Create a new LL node
        QNode loanApplicationSystem = new QNode(NIC, accountNumber, loanType,
                reasonOfLoanApplication, descriptionOfCollaterals);

        //If queue is empty, then the new node is front and rear both
        if(this.rearNode == null){
            this.frontNode = this.rearNode = loanApplicationSystem;
            return;
        }
        //Add the new node at the end of the queue and change rear
        this.rearNode.next = loanApplicationSystem;
        this.rearNode = loanApplicationSystem;
    }

    //Method to remove an key from queue
    void dequeue(){

        //If queue is empty, return null
        if(this.frontNode == null)
            return;

        //Store previous front and move front a=one node ahead
        QNode temp = this.frontNode;
        this.frontNode = this.frontNode.next;

        //If front becomes null, then change rear also as null
        if(this.frontNode == null)
            this.rearNode = null;
    }
}
```

III. Implement Loan system

loanSystemApplication.java :

```
package sample;

public class loanSystemApplication {
    loanSystemApplication.QNode fNode, rNode;

    public loanSystemApplication() {
        this.fNode = this.rNode = null;
    }

    // Creating a Method to add loan details to the queue
    void enqueue(String Name, int NIC, int accountNumber, String loanType,
String reasonForApplyingTheLoan, String descriptionOfCollateral) {

        // Creating a new Linked List node
        QNode loanAppSystem = new QNode(Name, NIC, accountNumber, loanType,
reasonForApplyingTheLoan, descriptionOfCollateral);

        // If queue is empty, then new node is front and rear both
        if (this.rNode == null) {
            this.fNode = this.rNode = loanAppSystem;
            return;
        }

        // Add the new node at the end of the queue and change rear
        this.rNode.next = loanAppSystem;
        this.rNode = loanAppSystem;
    }

    // Creating a Method to delete a loan application from queue.
    public void dequeue() {
        // If queue is empty, return Null
        if (this.fNode == null)
            return;

        // Store previous front and move front one node ahead
        QNode temp = this.fNode;
        this.fNode = this.fNode.next;

        // If front becomes nULL, then change rear also as NULL
        if (this.fNode == null)
            this.rNode = null;
    }

    public int displayDetailsFrontNode() {
        System.out.println("Customers Name: " + this.fNode.Name);
        System.out.println("Customers NIC No: " + this.fNode.NIC);
        System.out.println("Customers bank account No: " +
this.fNode.accountNumber);
        System.out.println("Type of the Loan: " + this.fNode.loanType);
        System.out.println("Reason for taking the loan: " +
this.fNode.reasonForApplyingTheLoan);
```

```
        System.out.println("Description Collateral: " +
this.fNode.descriptionCollateral);
        return 0;
    }

    public void displayDetailsRearNode() {
        System.out.println("Customers NIC No: " + this.rNode.Name);
        System.out.println("Customers NIC No: " + this.rNode.NIC);
        System.out.println("Customers bank account No: " +
this.rNode.accountNumber);
        System.out.println("Type of the Loan: " + this.rNode.loanType);
        System.out.println("Reason for taking the loan: " +
this.rNode.reasonForApplyingTheLoan);
        System.out.println("Description Collateral: " +
this.rNode.descriptionCollateral);
    }

    // Creating a Linked list Node
    class QNode {
        String Name;
        int NIC;
        int accountNumber;
        String loanType;
        String reasonForApplyingTheLoan;
        String descriptionCollateral;
        QNode next;

        // Creating a constructor to create a new linked list node
        public QNode(String name, int NIC, int accNumber, String loanType,
String reasonForApplyingTheLoan, String descriptionCollateral) {
            this.Name = name;
            this.NIC = NIC;
            this.accountNumber = accNumber;
            this.loanType = loanType;
            this.reasonForApplyingTheLoan = reasonForApplyingTheLoan;
            this.descriptionCollateral = descriptionCollateral;
            this.next = null;
        }
    }
}
```

mainLoanSystemProgramme:

```
package sample;

import java.util.Scanner;

public class mainLoanSystemProgramme {
    public static void main(String[] args) {
        String enterChoice;
        String decision = "";
        int count= 1;
        Scanner sc = new Scanner(System.in);

        System.out.println("..................Welcome to Hatton National
Bank Loan Systems.............");
        System.out.print("Normal customer - [A] \nInner customer -  [B]
\nHigh priority customer - [C] ");
        System.out.println("");
        System.out.println("");
        System.out.print("Select customer type: ");
        enterChoice = sc.next();

        if (enterChoice.equals("A")){
            normalCustomer();
        }else if(enterChoice.equals("B")){
            innerCircleCustomers();
        }else if(enterChoice.equals("C")){
            highPriorityCustomers();
        }else{
            System.out.println("Invalid Input!");
        }
    }

    public static void normalCustomer(){
        loanSystemApplication normalCustomers = new loanSystemApplication();
        System.out.println("..................Normal Customer's
Loans..................... ");
        normalCustomers.enqueue("Sandushke De Alwis",
752148963,1034569893,"Personal loan","Debt consolidation", "The home or real
estate you purchase is often used as collateral when you take out a
mortgage");
        normalCustomers.enqueue("Imaya Perera", 795136595,1203654798,"Home
equity loan","Moving costs", "The vehicle you purchase is typically used as
collateral when you take out a car loan");
        normalCustomers.enqueue("Ishma
Perera",853126978,1031965378,"Mortgage","Alternative to a payday loan"," A
cash deposit is used as collateral for secured credit cards.");
        normalCustomers.enqueue("Hirushke De
Alwis",945783622,1023024589,"Auto loan","Home remodeling", "A valuable item
that the lender agrees to use as collateral, such as your home or your
savings account");
        String decision;
        int count = 1;
        Scanner sc = new Scanner(System.in);

        System.out.println("To enqueue you need to dequeue first");
```

```
        System.out.print("Do you want to enqueue: ");
        decision = sc.next();
        if (decision.equals("yes")){
            normalCustomers.dequeue();
            System.out.println("");
            System.out.println("Next Normal Customer's details: ");
            normalCustomers.displayDetailsFrontNode();
        }
        System.out.println("");
        System.out.print("Do you want to dequeue: ");
        decision = sc.next();
        if(decision.equals("yes")){
            System.out.println();
            System.out.println("Previous normal Customer's details: ");
            normalCustomers.displayDetailsRearNode();
        }else if(decision.equals("no")){
            System.out.println("");
        }
        else{
            System.out.println("Error!");
        }
    }


    public static int innerCircleCustomers(){
        loanSystemApplication innerCircleCustomers = new
loanSystemApplication ();
        System.out.println("...................Inner Circle Customer's
Loans..................... ");
        innerCircleCustomers.enqueue("Ruwinika
Perera",876432169,2031265479,"Debt Consolidation loan","Vehicle Financing",
"The home or real estate you purchase is often used as collateral when you
take out a mortgage");
        innerCircleCustomers.enqueue("Shiran De
Alwis",953164789,1023456298,"Boat loan","Wedding Expenses", "The vehicle you
purchase is typically used as collateral when you take out a car loan");
        innerCircleCustomers.enqueue("Samantha
Perera",984626863,1023459233,"Payday loan","Emergency Expenses", "A cash
deposit is used as collateral for secured credit cards");
        innerCircleCustomers.enqueue("Ishani
Perera",789432692,2031697854,"Small business loan","Home Improvement and
Repairs", "A valuable item that the lender agrees to use as collateral, such
as your home or your savings account");
        innerCircleCustomers.enqueue("Shamika
Weudagedara",201345695,1110236549,"Title loan","Emergency Cash Assistance",
"A cash deposit is used as collateral for secured credit cards");
        innerCircleCustomers.dequeue();
        System.out.println("Next Inner circle Customer's details: ");
        innerCircleCustomers.displayDetailsFrontNode();
        innerCircleCustomers.dequeue();
        System.out.println("Next Inner circle Customer's details: ");
        System.out.println();
        System.out.println("Previous Inner circle Customer's details: ");
        innerCircleCustomers.displayDetailsRearNode();
        System.out.println("");
        System.out.println("Next Normal Customer's details: ");
        innerCircleCustomers.displayDetailsFrontNode();
```

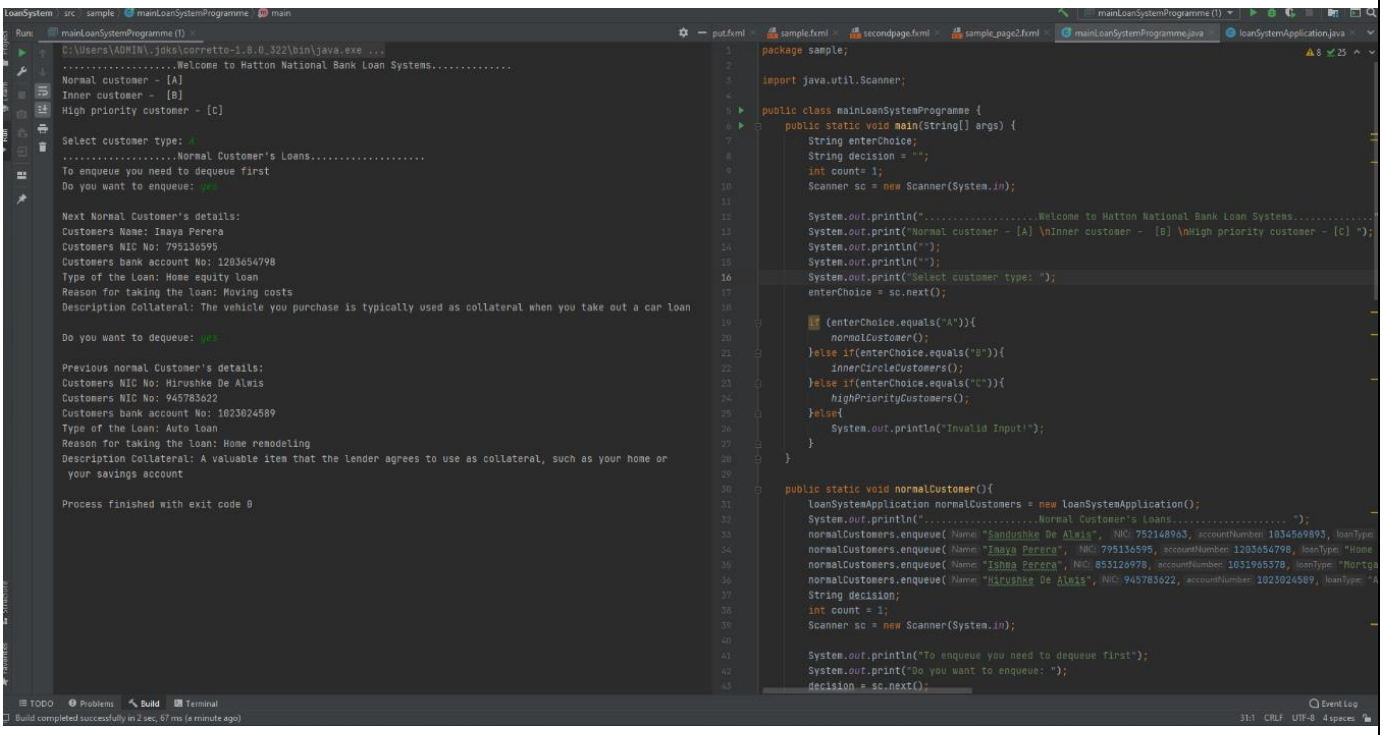
```
        System.out.println();
    }


    public static void highPriorityCustomers(){
        loanSystemApplication highPriorityCustomers = new
loanSystemApplication();
        System.out.println("..................High Priority Customer's
Loans.....................");
        highPriorityCustomers.enqueue("Irushi
Perera",935387624,203126578,"Recreational vehicle Loan","Vacation costs", "A
valuable item that the lender agrees to use as collateral, such as your home
or your savings account");
        highPriorityCustomers.enqueue("Chryshall
Fonseka",543125978,251345632,"Gold loan","Purchasing of a Land", "The vehicle
you purchase is typically used as collateral when you take out a car loan");
        highPriorityCustomers.enqueue("Rushealie
Fonseka",726459835,201236546,"Family loan","Relocation to a New City", "A
cash deposit is used as collateral for secured credit cards");
        highPriorityCustomers.enqueue("Rushinie
Ranasinghe",726459835,201236546,"Family loan","Relocation to a New City", "A
cash deposit is used as collateral for secured credit cards");
        highPriorityCustomers.dequeue();
        System.out.println("Next high priority Customer's details: ");
        highPriorityCustomers.displayDetailsFrontNode();
        System.out.println();
        System.out.println("Previous high priority Customer's details: ");
        highPriorityCustomers.displayDetailsRearNode();
        System.out.println("");
        System.out.println("Next Normal Customer's details: ");
        highPriorityCustomers.displayDetailsFrontNode();
    }
}
```

Explanation: The inputs are being hardcoded in the loan system application which will enqueue and dequeue and gives the output once the user selects the customer type.

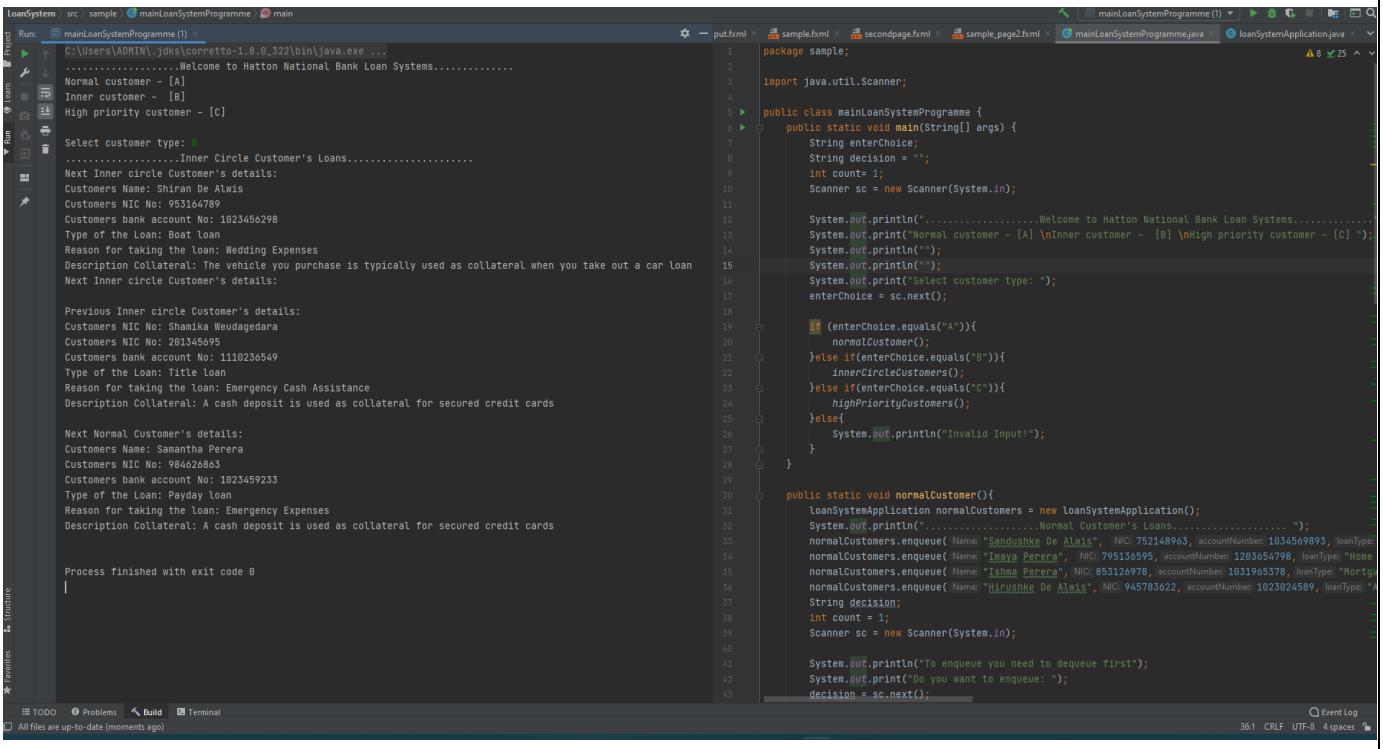
Note: In the above loan system application code the output results of A(normalcustomers) is different than B(innercirclecustomers) and C(highprioritycustomers). In scenario [A] it asks for enqueuing or dequeuing from user while in B and C it straight away displays all enqueue and dequeue results at once

Screenshots of outputs

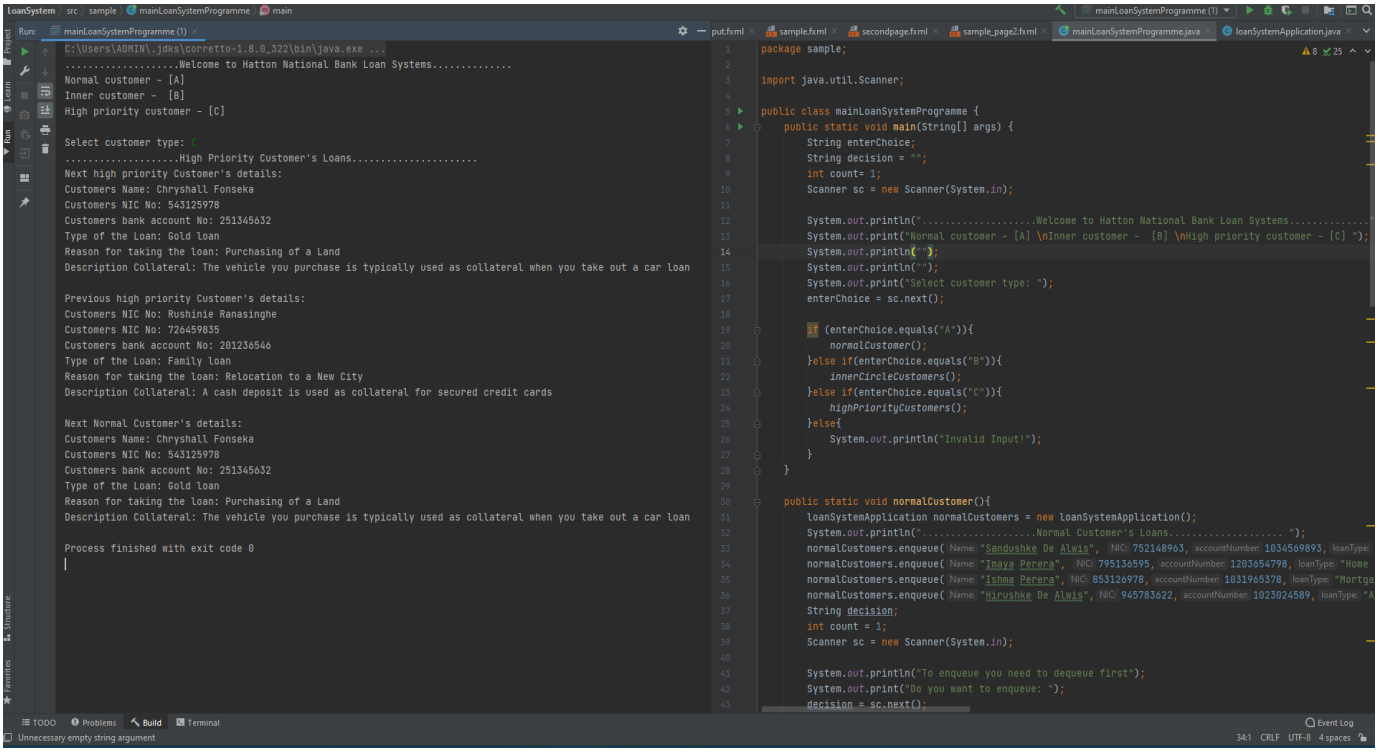
Output 1



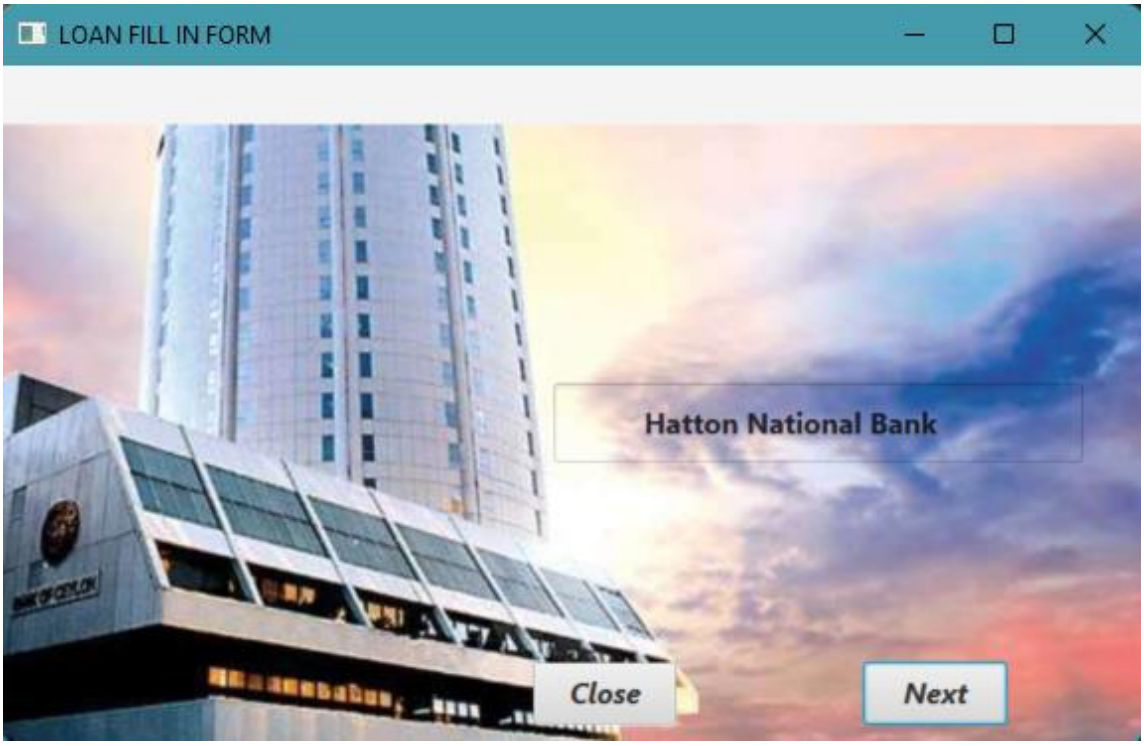
Output 2

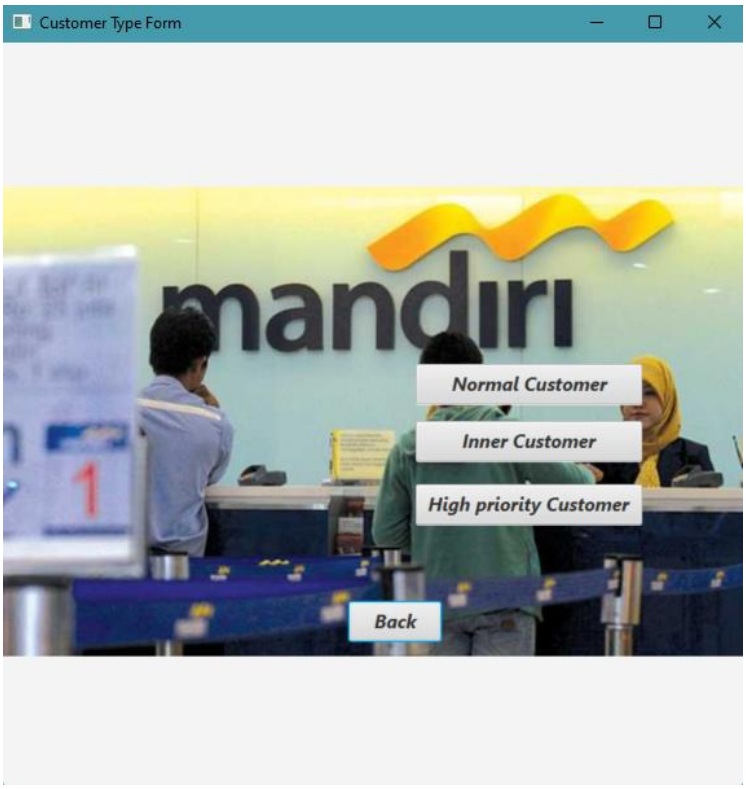


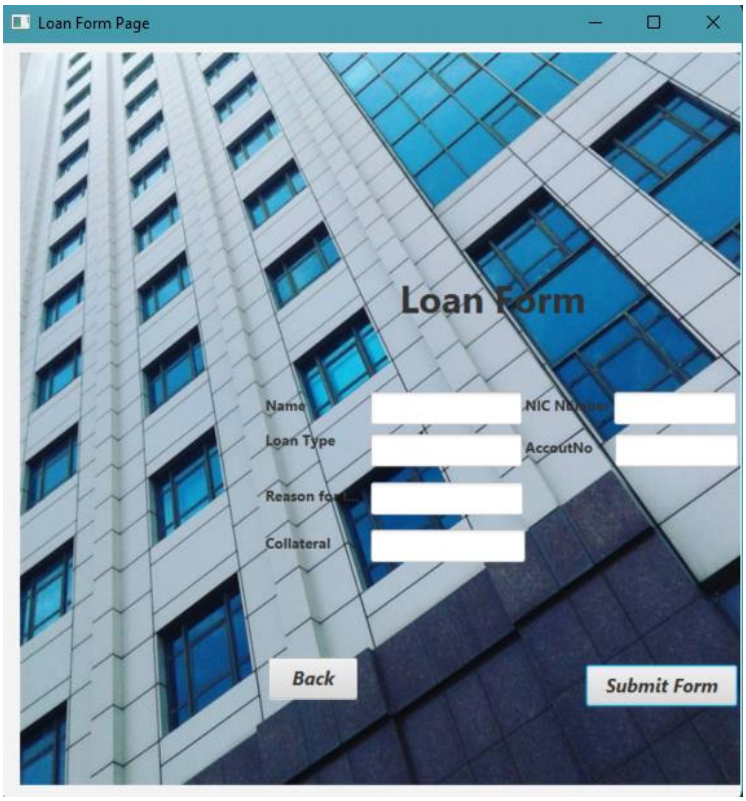
Output 3



Screenshot Outputs of the GUI interfaces




Customer Type Form
mandiri
Normal Customer
Inner Customer
High priority Customer
Back


Loan Form Page
Loan Form
Name
Loan Type
AccoutNo
Collateral
Back
Submit Form

Note: I haven't implemented the loan system using GUI interfaces. Everything in the loan system runs through the command line. It's just a couple of interfaces I made to show how the loan system will look like with the interfaces.

Task 2

a) Explain Fibonacci number

- A Fibonacci numbers is a sequence of numbers which starts from zero and then followed by one then based on a rule each number gets equal to the sum of proceeding two numbers.

b) Discuss the application of Fibonacci numbers:

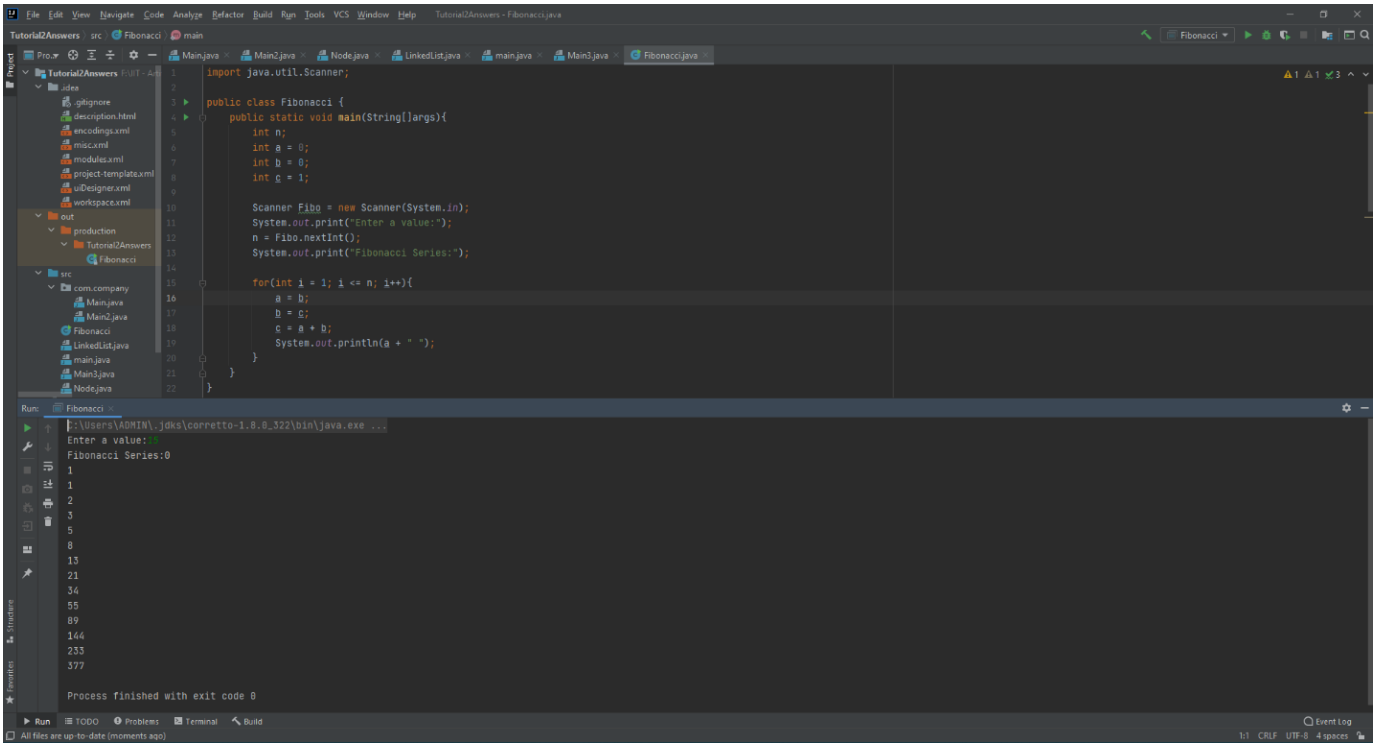
- Fibonacci numbers are used when implementing computer algorithms, when heaping data structures.
- Fibonacci sequences are used in graphs called Fibonacci cubes for interconnecting parallel and distributed systems

c) Implement a program to calculate the Fibonacci value of a given number using loops. Test the solution implemented with sample inputs

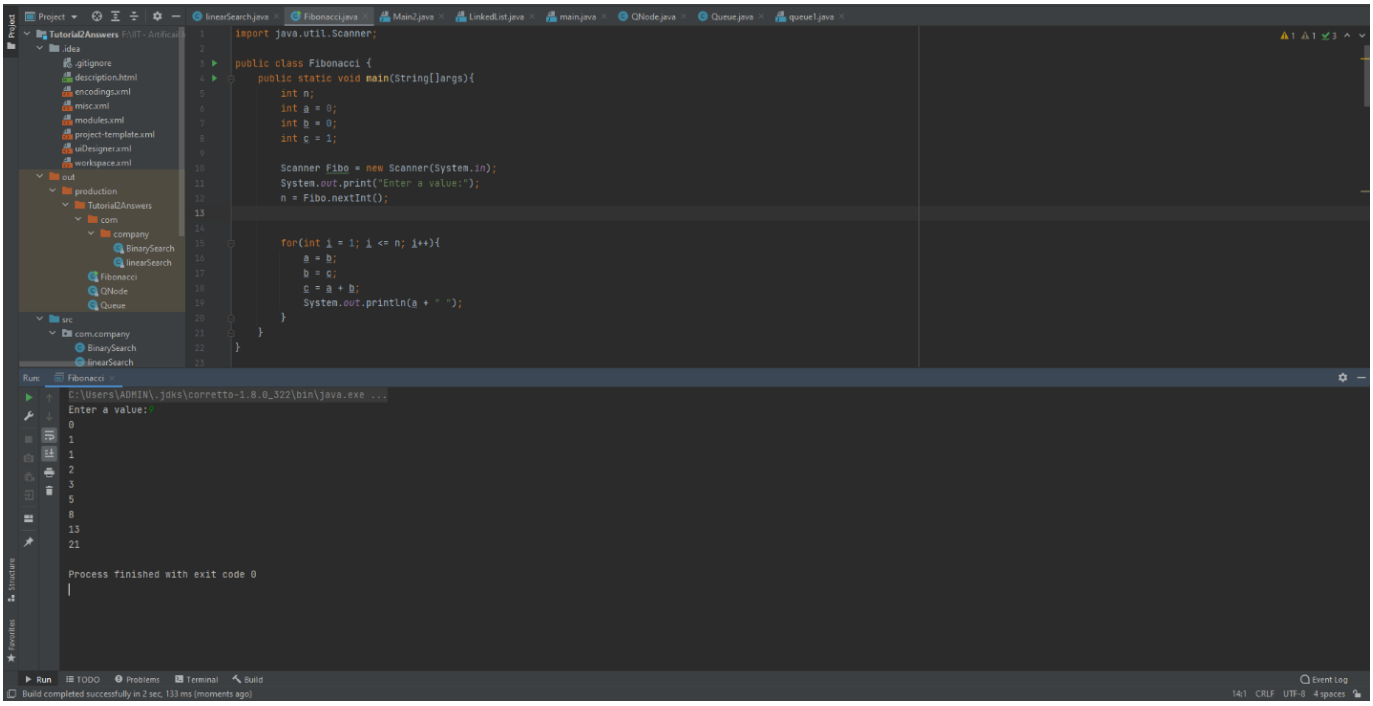
a)
```java
import java.util.Scanner;

public class Fibonacci {
    public static void main(String[]args){
        int n;
        int a = 0;
        int b = 0;
        int c = 1;

        Scanner Fibo = new Scanner(System.in);
        System.out.println("Enter a value:");
        n = Fibo.nextInt();
        System.out.println("Fibonacci Series:");

        for(int i = 1; i <= n; i++){
            a = b;
            b = c;
            c = a + b;
            System.out.println(a + " ");
        }
    }
}
```

Explanation: The codes represents the Fibonacci series using loops. When the user inputs a value it prints all the Fibonacci numbers up to that number. And always in every Fibonacci series, the starting number will be zero while the next value is one.

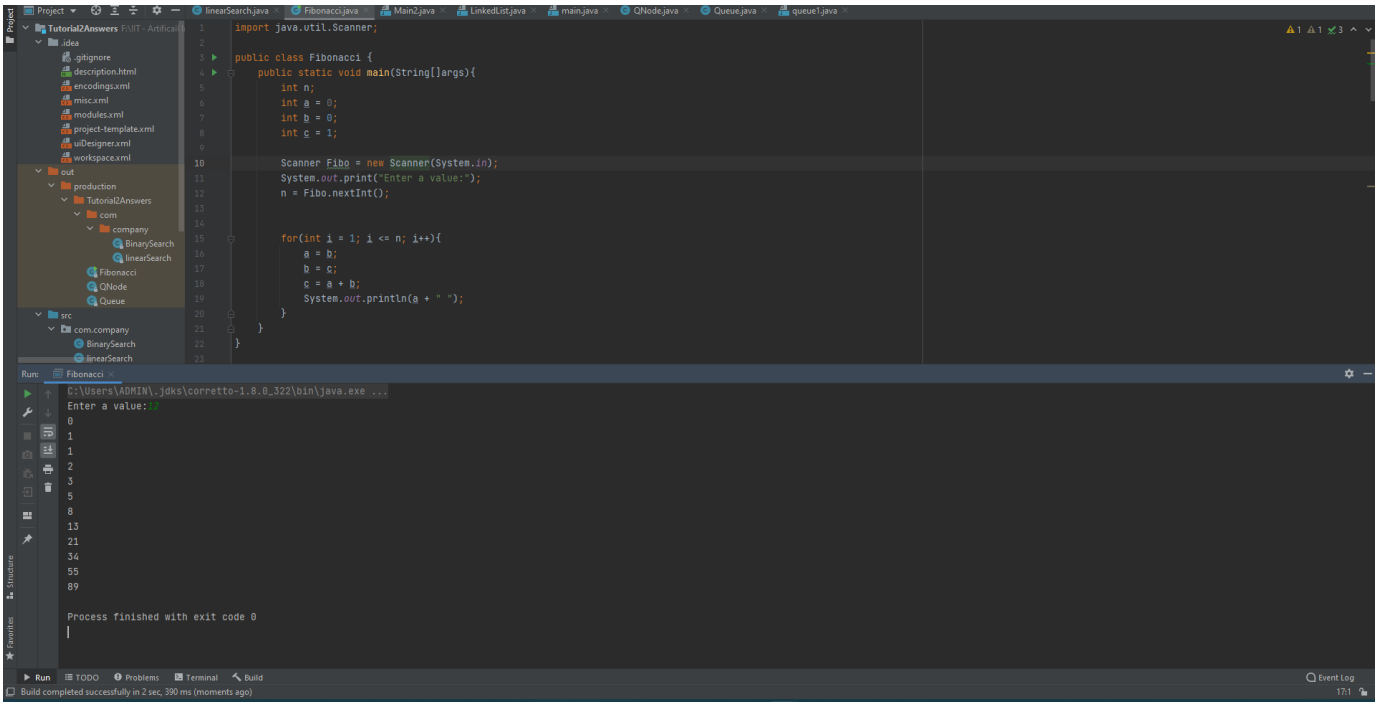
Output 1:



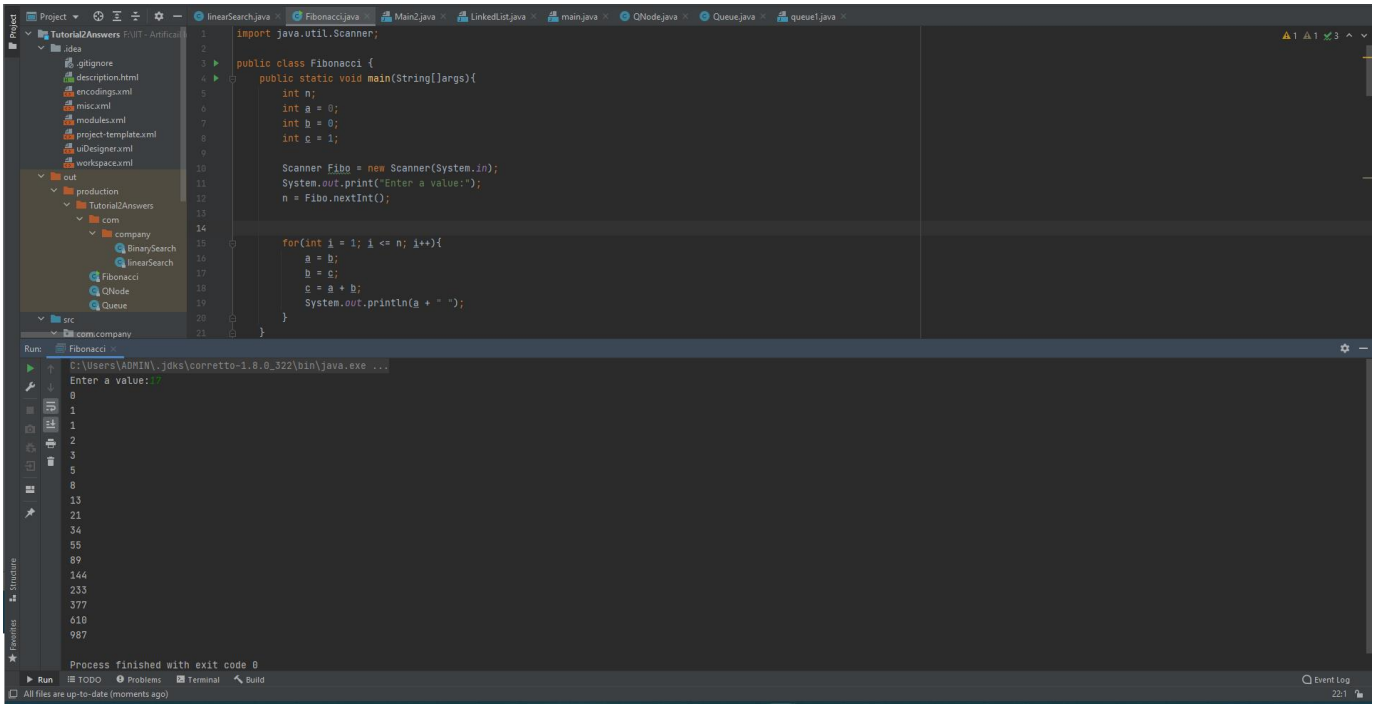
Output 2:



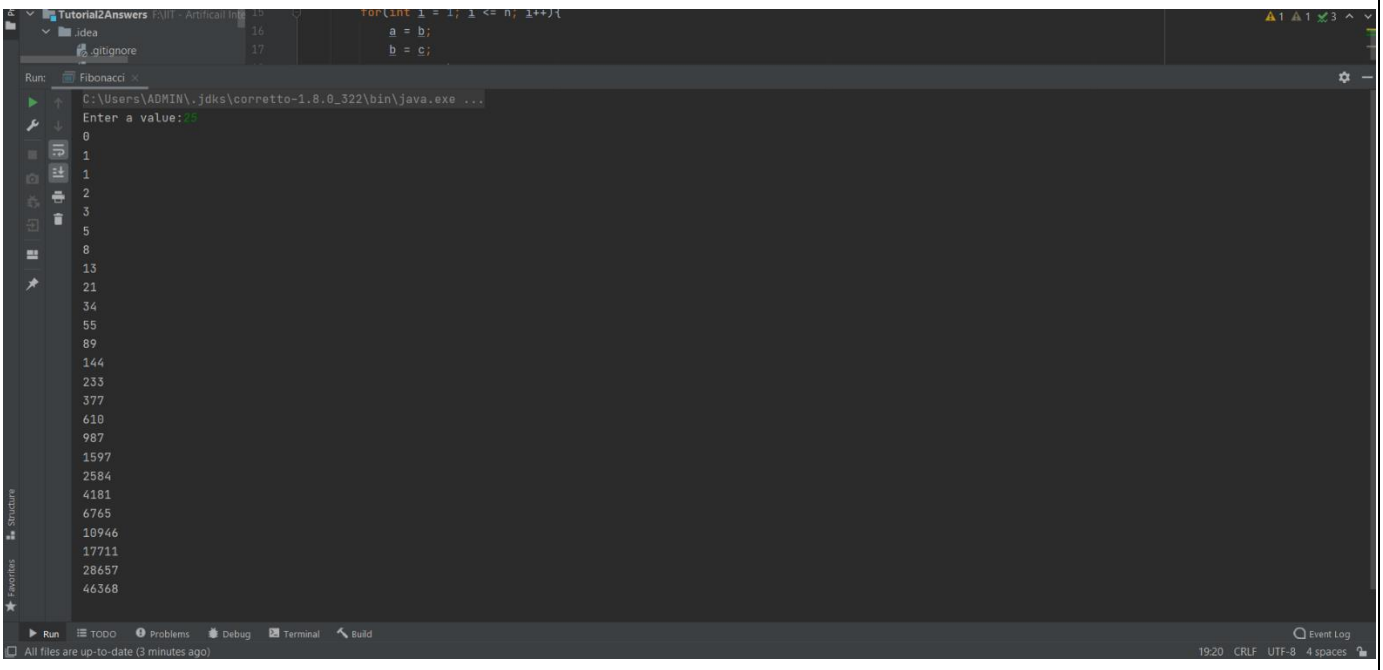
Output 3:



Output 4:



Output 5:



Test Case	userInput	Expected outcome	Actual outcome	Pass/ Fail
Test 1	**15**	**0, 1, 1, 2 ,3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377**	**0, 1, 1, 2 ,3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377**	**Pass**
Test 2	**9**	**0, 1, 1, 2, 3, 5, 8, 13, 21**	**0, 1, 1, 2, 3, 5, 8, 13, 21**	**Pass**
Test 3	**12**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89**	**Pass**
Test 4	**17**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987**	**Pass**
Test 5	**25**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368**	**Pass**

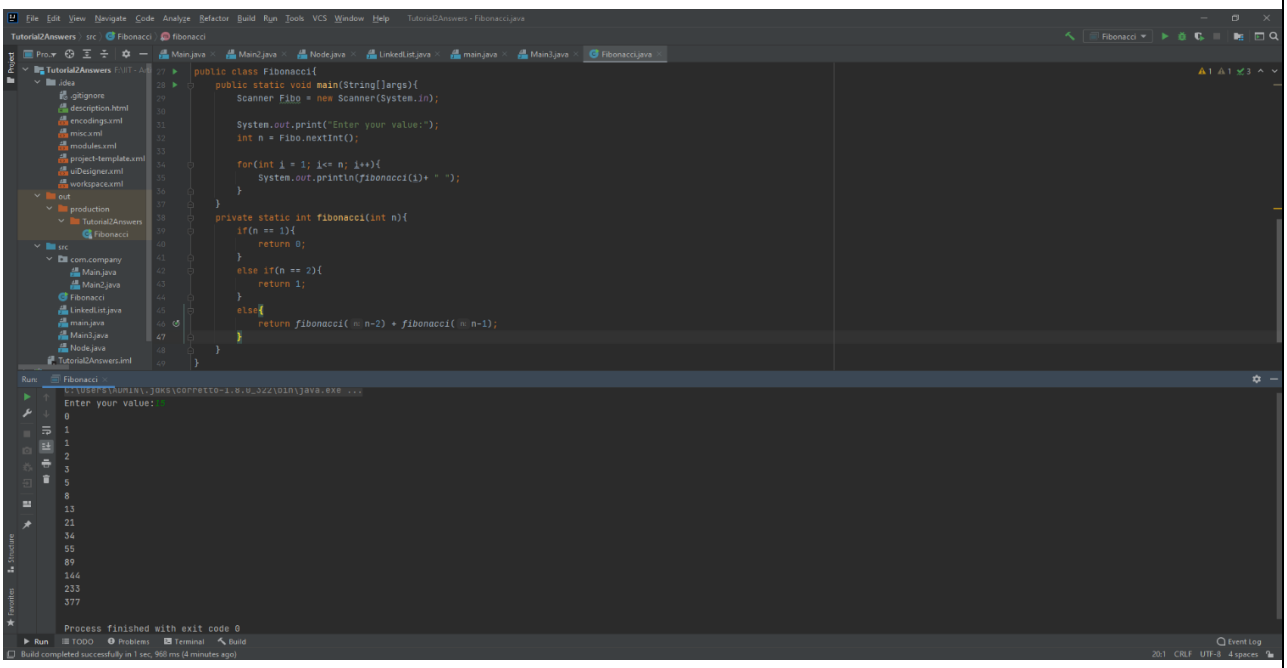
d) Implement a program to calculate the Fibonacci value of a given number using Recursion. Test the solution implemented with the sample inputs

```
import java.util.Scanner;

public class Fibonacci{
    public static void main(String[]args){
        Scanner Fibo = new Scanner(System.in);

        System.out.println("Enter your value:");
        int value = Fibo.nextInt();

        for(int x = 1; x<= value; i++){
            System.out.println(fibonacci(x)+ " ");
        }
    }
    private static int fibonacci(int value){
        if(value == 1){
            return 0;
        }
        else if(value == 2){
```

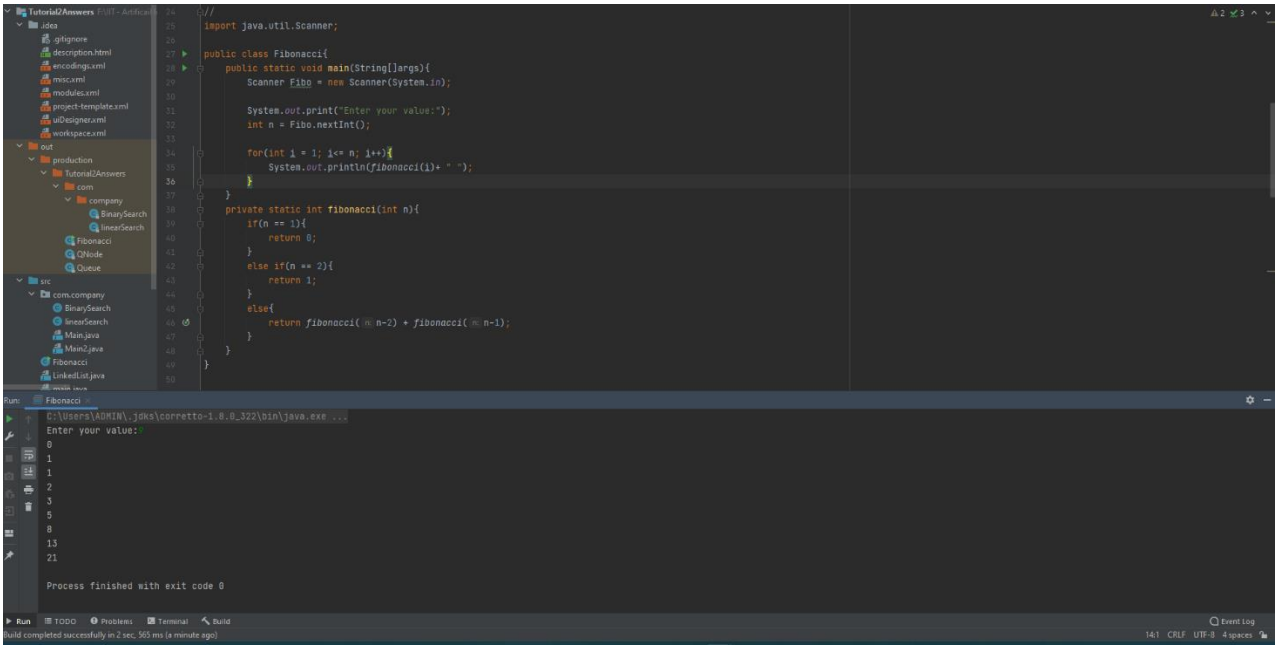
```
            return 1;
        }
        else{
            return fibonacci(value -2) + fibonacci(value -1);
        }
    }
}
```

Explanation: The codes represents the Fibonacci series using loops. When the user inputs a value it prints all the Fibonacci numbers up to that number. And always in every Fibonacci series, the starting number will be zero while the next value is one.

Output 1:



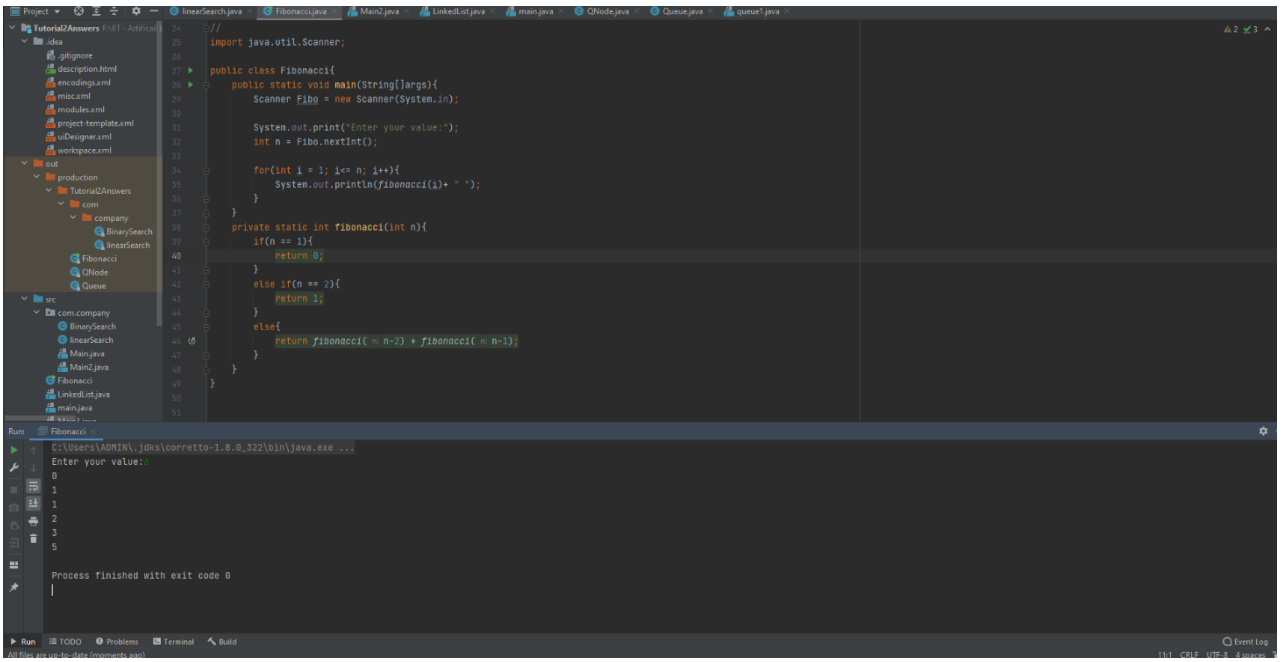
Output 2:



```
//
import java.util.Scanner;

public class Fibonacci{
    public static void main(String[]args){
        Scanner Fibo = new Scanner(System.in);

        System.out.print("Enter your value:");
        int n = Fibo.nextInt();

        for(int i = 1; i<= n; i++){
            System.out.println(fibonacci(i)+ " ");
        }
    }
    private static int fibonacci(int n){
        if(n == 1){
            return 0;
        }
        else if(n == 2){
            return 1;
        }
        else{
            return fibonacci( n-2) + fibonacci( n-1);
        }
    }
}
```

```
C:\Users\ADMIN\.jdks\corretto-1.8.0_322\bin\java.exe ...
Enter your value:
0
1
1
2
3
5
8
13
21

Process finished with exit code 0
```

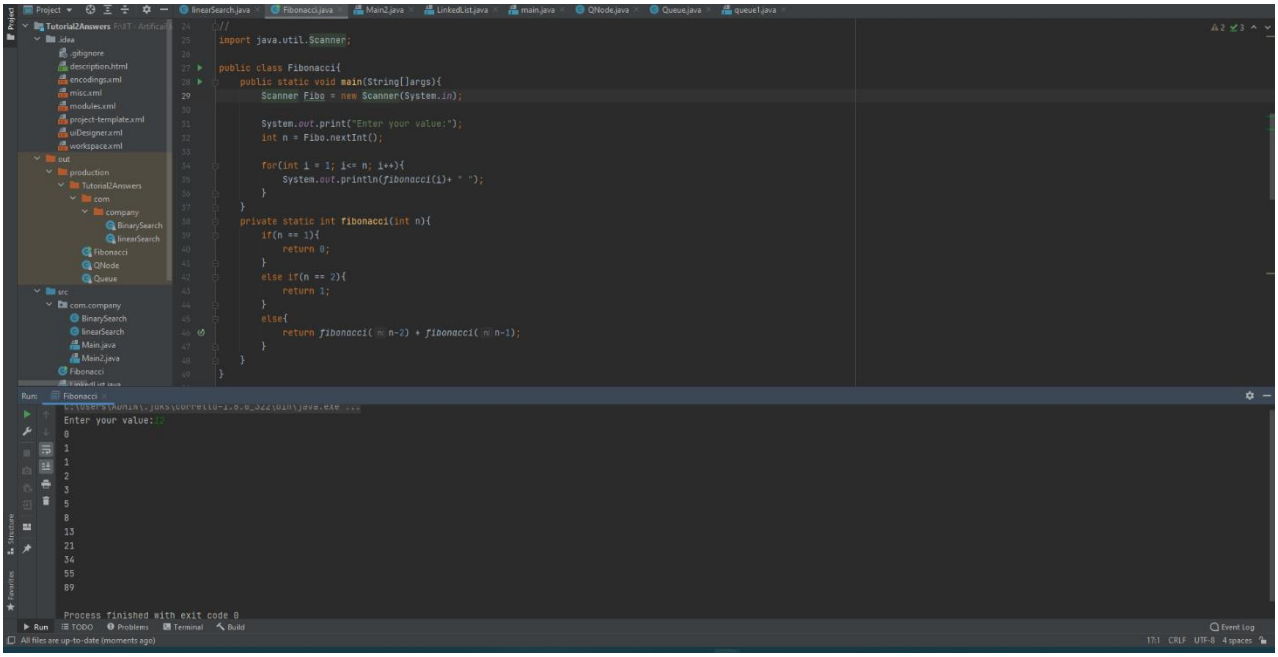
Output 3:



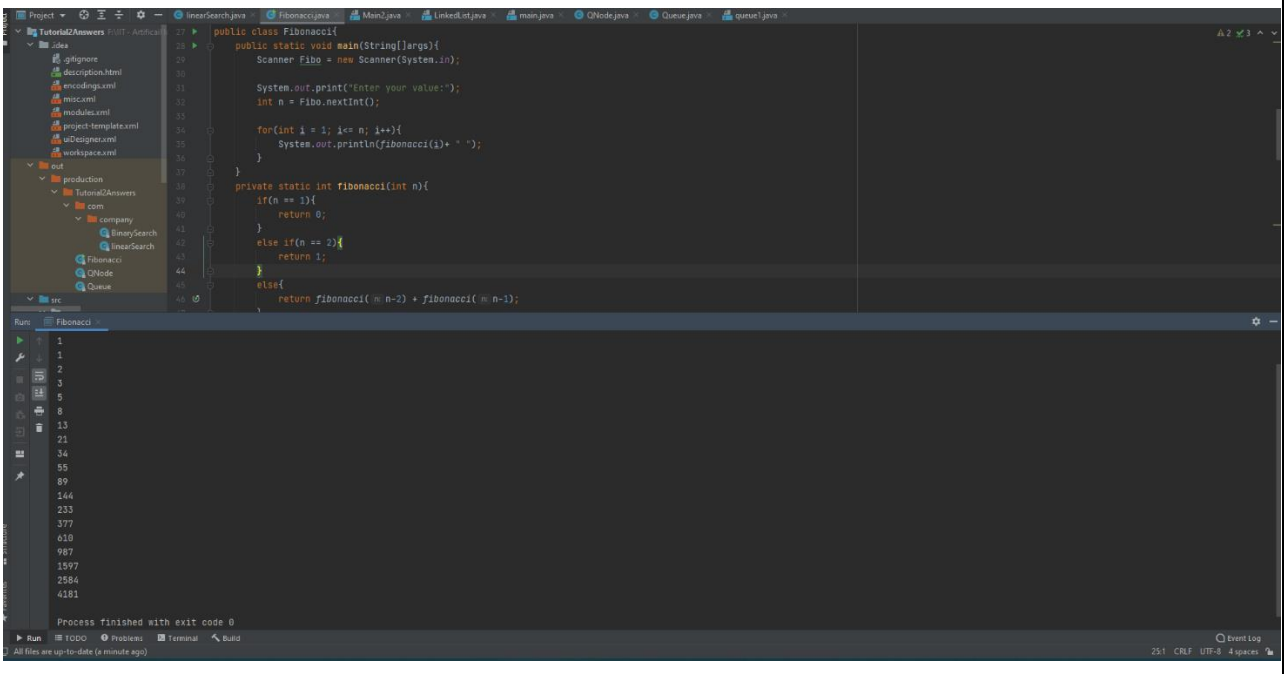
```
//
import java.util.Scanner;

public class Fibonacci{
    public static void main(String[]args){
        Scanner Fibo = new Scanner(System.in);

        System.out.print("Enter your value:");
        int n = Fibo.nextInt();

        for(int i = 1; i<= n; i++){
            System.out.println(fibonacci(i)+ " ");
        }
    }
    private static int fibonacci(int n){
        if(n == 1){
            return 0;
        }
        else if(n == 2){
            return 1;
        }
        else{
            return fibonacci( n-2) + fibonacci( n-1);
        }
    }
}
```

```
C:\Users\ADMIN\.jdks\corretto-1.8.0_322\bin\java.exe ...
Enter your value:
0
1
1
2
3
5

Process finished with exit code 0
```

Output 4:



Output 5:



Test Case	userInput	Expected outcome	Actual outcome	Pass/ Fail
Test 1	**15**	**0, 1, 1, 2 ,3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377**	**0, 1, 1, 2 ,3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377**	**Pass**
Test 2	**9**	**0, 1, 1, 2, 3, 5, 8, 13, 21**	**0, 1, 1, 2, 3, 5, 8, 13, 21**	**Pass**
Test 3	**6**	**0, 1, 1, 2, 3, 5**	**0, 1, 1, 2, 3, 5**	**Pass**
Test 4	**12**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89**	**Pass**
Test 5	**30**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229**	**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229**	**Pass**

e) Implement linear search and binary search algorithms

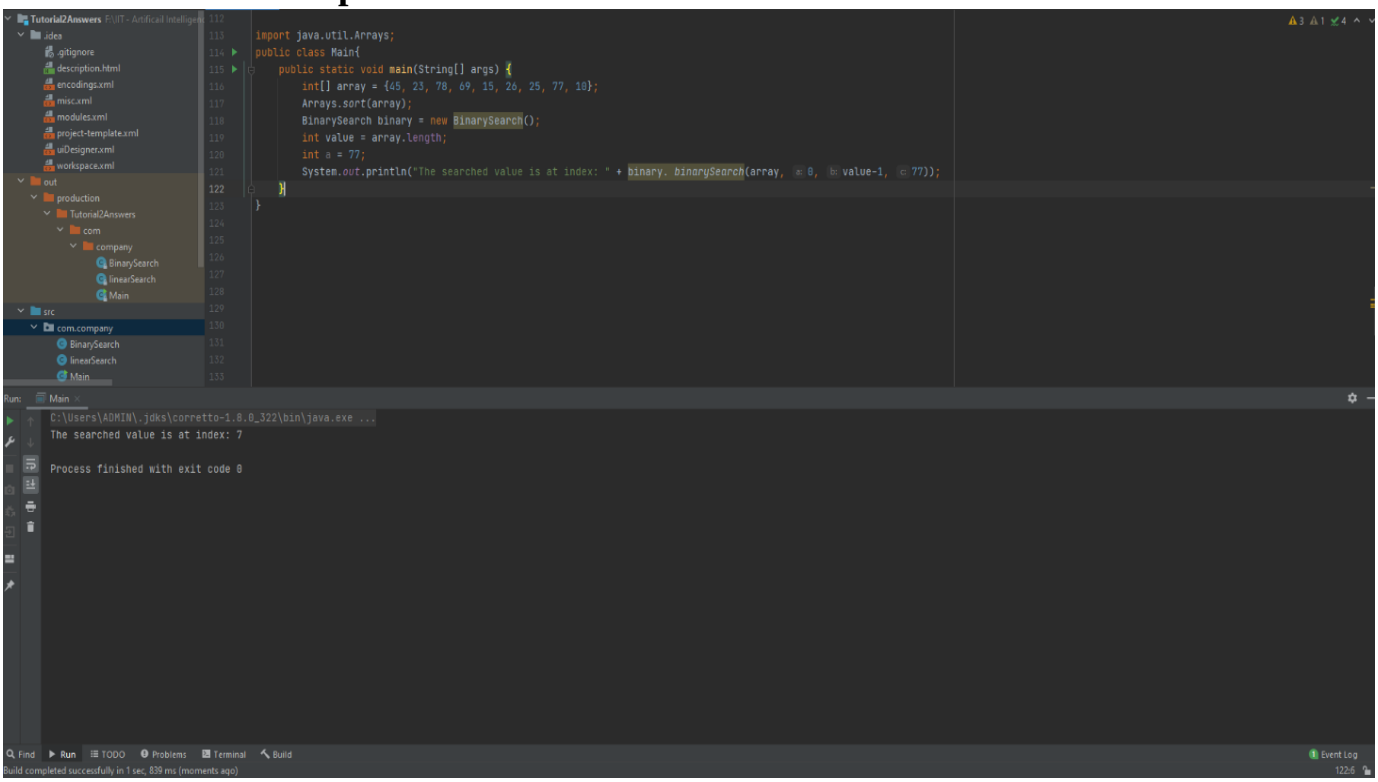
Linear Search:

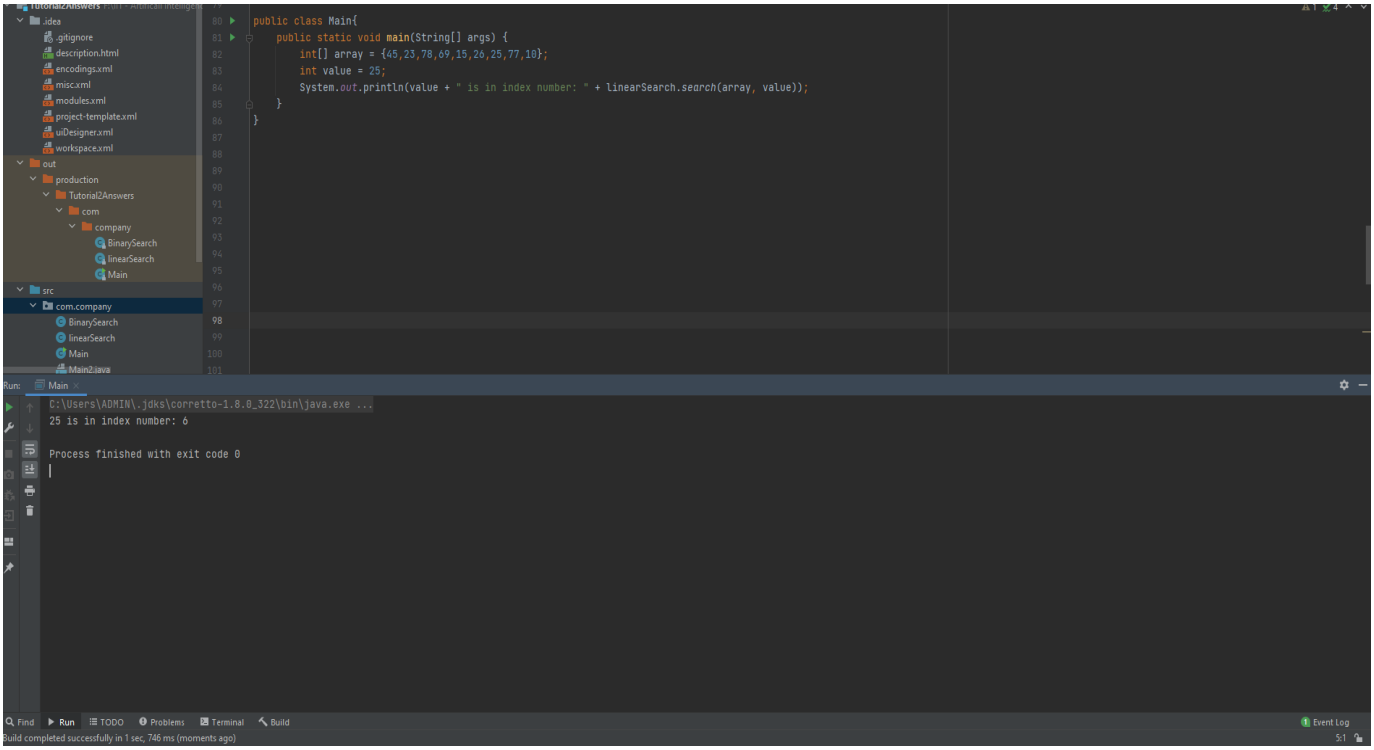
```
package com.company;

public class linearSearch {
    public static int search(int array[], int a) {
        int x = array.length;
        for(int i = 0; i < x; i++){
            if(array[i] == a){
                return i;
            }
        }
        return -1;
    }
}
```

```
public class Main{
    public static void main(String[] args) {
        int[] array = {45,23,78,69,15,26,25,77,10};
        int value = 25;
        System.out.println(value + " is in index number: " +
linearSearch.search(array, value));
    }
}
```

Explanation: In linear search algorithm the sequance of array numbers are sorted one by one over the time. Once the item is found and matched, it will returened or else it will keep contuing until the end of the array.

Screenshot of the output of the code:





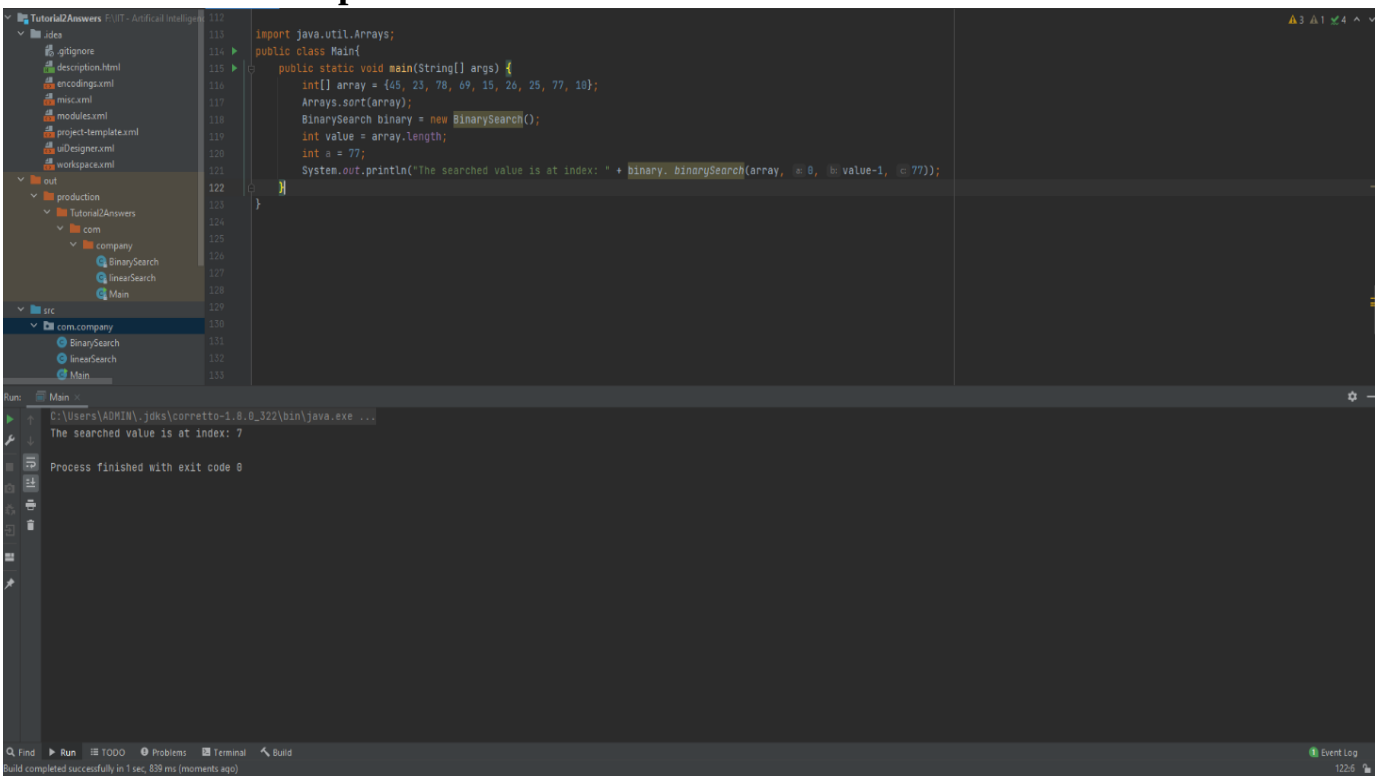
Binary Search:

```
package com.company;

public class BinarySearch {
    static int binarySearch(int array[], int a, int b, int c){
        if(b >= a){
            int midVal = (a + b) /2;
            if(array[midVal] ==  c){
                return midVal;
            }
            if(array[midVal] > c){
                return binarySearch(array, a, midVal -1, c);
            }
            return binarySearch(array, midVal + 1, b, c);
        }else{
            return -1;
        }
    }
}
```

```java
import java.util.Arrays;
public class Main{
    public static void main(String[] args) {
        int[] array = {45, 23, 78, 69, 15, 26, 25, 77, 10};
        Arrays.sort(array);
        BinarySearch binary = new BinarySearch();
        int value = array.length;
        int a = 77;
        System.out.println("The searched value is at index: " + binary.
binarySearch(array, 0, value-1, 77));
    }
}
```

Explnantion: Binary search algorithm is used in sorted arrays which divides the search interval by half. This algorithm method is used to get information of the array and reduce time complexity O(log n)

Screenshot of the output of the code:



f) Compare the performance of the two algorithms you implemented in Task 2 Part a)

Linear Search	Binary Search
• **Linear search is preferred to use for small sized data sets because it is less efficient in handling large data sets**	• Binary search is preferred to use for use large size data sets because it has more efficiency in handling it than small sized data sets
• **Linear search can be implemented for both single are multidimensional array**	• Binary search can be implemented only for a multidimensional array
• **Linear search is iterative which uses sequential approach for its functionalities**	• Binary search make use of divide and get the best approach in its functionalities
• **Time complexity of linear search is O(n) which is not that efficient**	• Time complexity of binary search is O(log2n) which is efficient than linear search

References:

Sanfoundry. 2022. *Java Program to Generate Fibonacci Numbers - Sanfoundry*. [online] Available at: <https://www.sanfoundry.com/java-program-generate-fibonacci-numbers/> [Accessed 3 April 2022].

Pencil Programmer. 2022. *Java Program for Fibonacci Series [Loop, Recursion]*. [online] Available at: <https://pencilprogrammer.com/java-programs/fibonacci-series/> [Acessed 3 April 2022].

GeeksforGeeks. 2022. *Binary Search - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/binary-search/> [Accessed 3 April 2022].

GeeksforGeeks. 2022. *Binary Search - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/binary-search/> [Accessed 3 April 2022].