```
                              ؛package pkg3c4_group5

                           ؛import java.util.Scanner

                                         c4Group5٣//
                                   } public class Main

        ؛static Scanner input = new Scanner(System.in)

              } public static void main(String[] args)

                                  ؛()displayMenu
                  ؛()int selection = input.nextInt

                           } switch (selection)
                                    :case 1
                            ؛()mergeArrays
                                   ؛break

                                    :case 2
                             ؛()mergeLists
                                   ؛break

                                    :case 3
                                ؛()inStack
                                   ؛break

                                    :case 4
                             ؛()MergeQueue
                                   ؛break

                                    :case 5
                              ؛()MergeTree
                                   ؛break
                                  :default
    ؛System.out.println("Invalid Choice!")
                                          {
                                              {

         Display Menu NOURA ALQASEM 442001119//
             } ()public static void displayMenu

   System.out.print("*Welcome to the Data structure Merging
    "Assienment*\n" + "Please select one of the following options:\n
 Mergin two Arrays data structure type\n" + "2:  :١" +
                "Mergin two Single Linked Lists data structure type\n
 Mergin two Stacks data structure type\n" + "4:  :٣" +
                             "Mergin two Queues data structure type\n
Mergin two Trees data structure type\n" + "Your  :٥" +
                                                ؛("selection is:\n

                                              {

               ARRAY NOURA ALQASEM 442001119 //
         } public static int readSize(String str)
System.out.print("Please enter The " + str + " Data set's size:
                                                          ؛")
                   ؛()int size = input.nextInt
```

```
                                        ؛return size
                                                    {

        } public static int[] createArray(String str, int size)
                              ؛int[] arr = new int[size]

    System.out.println("Please enter The " + str + " Data set
                                                   ؛elements: ")
                    } for (int i = 0; i < arr.length; i++)
                            ؛()arr[i] = input.nextInt
                                                         {

                                        ؛return arr

                                                             {

                             } ()public static void mergeArrays

                      ؛int firstSize = readSize("First")
        ؛int[] firstArray = createArray("First", firstSize)

                     ؛int secondSize = readSize("Second")
    ؛int[] secondArray = createArray("Second", secondSize)

      ؛int[] mergedArray = new int[firstSize + secondSize]
                                          ؛int i = 0

            } for (int j = 0; j < firstArray.length; j++)
                   ؛mergedArray[i++] = firstArray[j]
                                                         {

           } for (int j = 0; j < secondArray.length; j++)
                  ؛mergedArray[i++] = secondArray[j]
                                                         {

            ؛System.out.print("The First given Array: ")
                               ؛printArray(firstArray)

           ؛System.out.print("The Second given Array: ")
                              ؛printArray(secondArray)

      ؛System.out.print("The Resultant Mergined Arrays: ")
                              ؛printArray(mergedArray)

                                                             {

                 } public static void printArray(int[] array)
                               ؛("]")System.out.print
                 } for (int i = 0; i < array.length; i++)
                          ؛System.out.print(array[i])
           if (i != array.length - 1) { //for commas
                          ؛(" ،")System.out.print
                                                   {
                                                         {
                              ؛("[")System.out.println
                                                             {

                     Linked List Rawan Alotaibi 442005266//
```

```
                                    } ()public static void mergeLists

                                                 The first list//
    ؛System.out.print("Please enter The first Data set's size: ")
                                 ؛()int firstSize = input.nextInt
              ؛()<>Linkedlist<Integer> firstList = new Linkedlist
؛System.out.println("Please enter The First Data set elements: ")
                            } for (int i = 0; i < firstSize; i++)
                                   ؛()int ele = input.nextInt
                                      ؛firstList.addLast(ele)
                                                                {

                                                The second list//
   ؛System.out.print("Please enter The Second Data set's size: ")
                                ؛()int secondSize = input.nextInt
             ؛()<>Linkedlist<Integer> secondList = new Linkedlist
  System.out.println("Please enter The Second Data set elements:
                                                                      ؛")
                           } for (int i = 0; i < secondSize; i++)
                                   ؛()int ele = input.nextInt
                                     ؛secondList.addLast(ele)
                                                                {

                                                          merge//
             ؛()<>Linkedlist<Integer> mergedList = new Linkedlist

                            } for (int i = 0; i < firstSize; i++)
                           ؛()int ele = firstList.removeFirst
                                      ؛firstList.addLast(ele)
                                     ؛mergedList.addLast(ele)
                                                                {

                           } for (int i = 0; i < secondSize; i++)
                          ؛()int ele = secondList.removeFirst
                                     ؛secondList.addLast(ele)
                                     ؛mergedList.addLast(ele)
                                                                {

              ؛System.out.println("The First given Linked List:")
                                               ؛()firstList.print

             ؛System.out.println("The Second given Linked List:")
                                              ؛()secondList.print

      ؛System.out.println("The Resultant Mergined Linked List: ")
                                              ؛()mergedList.print
                                                                    {

                                 Stack Reham Alshamraani 442001832 //
                                       } ()public static void inStack

                                                  the big Stack//
                        ؛()<>LLstack<Integer> bigst = new LLstack

                                            add the first stack//
    ؛System.out.print("Please enter the First data set's size: ")
                                        ؛()int s1 = input.nextInt
                        ؛()<>LLstack<Integer> inst1 = new LLstack
```

```
؛System.out.println("Please Enter the First data set elements: ")
                                   } for (int i = 0; i < s1; i++)
                                   ؛()int in1 = input.nextInt
                                             ؛inst1.push(in1)
                                             ؛bigst.push(in1)
                                                               {

                                         add the second stack//
   ؛System.out.print("Please enter the Second data set's size: ")
                                       ؛()int s2 = input.nextInt
                        ؛()<>LLstack<Integer> inst2 = new LLstack
  System.out.println("Please enter the Second data set elements:
                                                                    ؛")
                                   } for (int i = 0; i < s2; i++)
                                   ؛()int in2 = input.nextInt
                                             ؛inst2.push(in2)
                                             ؛bigst.push(in2)
                                                               {

                            Print the element in the first Stack//
                      ؛System.out.println("The First given Stack:")
                                                ؛()inst1.display
                          Print the elements in the second Stack//
                     ؛System.out.println("The Second given Stack:")
                                                ؛()inst2.display
                                Print the element in the big Stack//
             ؛System.out.println("The Resultant Margined Stacks :")
                                                ؛()bigst.display

                                                                   {

                                  Queue Noraa Almotairi 442004509 //
                                   } ()public static void MergeQueue

   ؛System.out.print("Please enter the first data set's size : ")
                                      ؛()int size1 = input.nextInt
                    ؛()<>LLQueue<Integer> firstQueue = new LLQueue
  System.out.println("Please enter the first data set elements :
                                                                    ؛")
                                } for (int i = 0; i < size1; i++)
                                  ؛()int elem = input.nextInt
                                    ؛firstQueue.Enqueue(elem)
                                                               {

  ؛System.out.print("Please enter the second data set's size : ")
                                      ؛()int size2 = input.nextInt
                   ؛()<>LLQueue<Integer> secondQueue = new LLQueue
 System.out.println("Please enter the second data set elements :
                                                                    ؛")
                                } for (int i = 0; i < size2; i++)
                                  ؛()int elem = input.nextInt
                                   ؛secondQueue.Enqueue(elem)
                                                               {

                ؛()<LLQueue<Integer> BigQueue = new LLQueue<Integer
                                ؛()long size = firstQueue.getSize
                                 } for (int i = 0; i < size; i++)
                             ؛()int elem = firstQueue.Dequeue
```

```
                                      ؛BigQueue.Enqueue(elem)
                                    ؛firstQueue.Enqueue(elem)

                                                                 {

                                    ؛()size = secondQueue.getSize
                                 } for (int i = 0; i < size; i++)

                            ؛()int elem = secondQueue.Dequeue
                                      ؛BigQueue.Enqueue(elem)
                                   ؛secondQueue.Enqueue(elem)

                                                                 {

                  ؛System.out.println("The firste given Queu : ")
                                       ؛()firstQueue.DisplayQueue
                  ؛System.out.println("The second given Queu : ")
                                      ؛()secondQueue.DisplayQueue
           ؛System.out.println("The Resultant Margined Queue : ")
                                         ؛()BigQueue.DisplayQueue
                                                                     {

                                 TREE ALL STUDENT IN GROUP WRITE IT//
                                     } ()public static void MergeTree

   ؛System.out.print("Please enter the first data set's size : ")
                                     ؛()int size1 = input.nextInt
                                 ؛int[] firstArr = new int[size1]
  System.out.println("Please enter the first data set elements :
                                                                       ؛")
                      } for (int i = 0; i < firstArr.length; i++)
                                  ؛()int elem = input.nextInt
                                          ؛firstArr[i] = elem
                                                                 {

  ؛System.out.print("Please enter the second data set's size : ")
                                     ؛()int size2 = input.nextInt
                                ؛int[] secondArr = new int[size2]
 System.out.println("Please enter the second data set elements :
                                                                       ؛")
                     } for (int i = 0; i < secondArr.length; i++)
                                  ؛()int elem = input.nextInt
                                         ؛secondArr[i] = elem
                                                                 {

                ؛()BinaryTree<Integer> firstTree = new BinaryTree
               ؛()BinaryTree<Integer> secondTree = new BinaryTree

                                  ؛firstTree.addRoot(firstArr[0])
                      } for (int i = 1; i < firstArr.length; i++)

                                    ؛int perant = (i - 1) / 2
                            ؛int LeftChild = (perant * 2) + 1
                           ؛int RightChild = (perant * 2) + 2

                              } if (firstArr[LeftChild] != 0)
؛firstTree.addLeft(firstArr[LeftChild], firstArr[perant])
                                                             {
                        } if ((RightChild) < firstArr.length)
                         } if (firstArr[RightChild] != 0)
            firstTree.addRight(firstArr[RightChild],
                                                       ؛firstArr[perant])
                                                        {
                                                            {
                                                         ؛++i

                                                                {

                                ؛secondTree.addRoot(secondArr[0])
                     } for (int i = 1; i < secondArr.length; i++)

                                    ؛int perant = (i - 1) / 2
                            ؛int LeftChild = (perant * 2) + 1
                           ؛int RightChild = (perant * 2) + 2

                             } if (secondArr[LeftChild] != 0)
                secondTree.addLeft(secondArr[LeftChild],
                                                      ؛secondArr[perant])
                                                            {

                       } if ((RightChild) < secondArr.length)
                        } if (secondArr[RightChild] != 0)
          secondTree.addRight(secondArr[RightChild],
                                                      ؛secondArr[perant])
                                                        {
                                                            {
                                                         ؛++i

                                                                {

  System.out.println("The First given Tree in in Order traversal
                                                                     ؛:")
                               ؛firstTree.InOrder(firstTree.Root)
         System.out.println("\nThe Second given Tree in in Order
                                                           ؛traversal :")
                             ؛secondTree.InOrder(secondTree.Root)

  System.out.println("\nEnter the parant at which the margirn is
                                                           ؛required : ")
                                    ؛()int perant = input.nextInt
                      ؛firstTree.MergeTwoTree(perant, secondTree)

            ؛System.out.println("The Resultant Margined Tree : ")
                               ؛firstTree.InOrder(firstTree.Root)
    System.out.println("\nThe Resultant Margined Tree'size : " +
                                                         ؛firstTree.size)

                                                                    {
                                                                        {
```