ANÍBAL SIDNEY DOS SANTOS JÚNIOR

==========

PROGRAMA – A ou 1

```
import java.util.Scanner;
import java.util.Stack;

public class main {

//================================================================================
//================================================================================

    public static void main(String[] args) {

        Scanner leia = new Scanner(System.in);


        // Criando Pilha Principal e Pilha Reversa
        Stack PA = new Stack(); //PILHA PRINCIPAL
        Stack PB = new Stack(); //PILHA REVERSA


        int QTD;
```

```
        int Num;
        int TopoPA = 0;
        int TopoPB = 0 ;
        int VarComp = 0 ;

        int cont = 0;
//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============

System.out.println("\n=====================================
==========");
        System.out.printf("\nDigite a quantidade de números para
empilhar: ");

        QTD = leia.nextInt();

System.out.println("\n=====================================
==========");

//=========================================================
=========================================================
===============
```

```
//=====================================================
=====================================================
===============

        while(cont < QTD){

        Num = leia.nextInt();



        //Se pilha A estiver vazia, então empilhar o primeiro
número
        if(PA.empty()) {
           PA.push(Num);

           System.out.printf("\nPilha que estava vazia agora têm:");
           System.out.println(PA);

System.out.println("================================");

        }



        //Senão
        else {
```

```
while(!PA.isEmpty()) {

    if(!PA.isEmpty()) {
        TopoPA = (int) PA.peek();   //----- Se a pilha A não estiver vazia, então é pq existe TOPO
    }                               //----- aí estou pegando o topo e jogando o valor dentro de uma variável



    if(TopoPA != Num ) {    //&& !PA.isEmpty()
        PB.push(PA.pop());

    }


    if(!PA.isEmpty()) {
        TopoPA = (int) PA.peek();
    }

    if(TopoPA == Num && !PA.isEmpty()) {
        PB.push(PA.pop());
        VarComp = 1;
    }
```

```
            }



        //-----  Se a Variável de comparação estiver em 0 assim como começou, então é pq não teve número repetido
        //-----  se não têm número repetido, terei que empilhá-lo
           if(VarComp == 0) {
              while(!PB.isEmpty()) {
                 PA.push(PB.pop());  // desempilha tudo da PILHA B e empilha na PILHA A
              }
              PA.push(Num); // depois de PILHA toda empilhada, empilhar também o número digitado
           }



        //-----  Se a Variável de comparação estiver em 1 é pq essa variável foi alterada
        //-----  e ela só é alterada quando encontra um TOPO da pilha A igual ao valor digitado
           if(VarComp == 1) {
```

```
            while(!PB.isEmpty()) {     // desempilha tudo da PILHA B e empilha na PILHA A
                PA.push(PB.pop());
            }
        VarComp = 0;    //----- aqui a variável recebe novamente o valor que ela começou, pois se não ela
        }               //----- iria sempre ficar em 1, assim, sempre acusando que teria número igual
        }
        cont++;
        }

        System.out.println("Imprimindo as duas PILHAS");
        System.out.println(PA);
        System.out.println(PB);
```

```
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============


        leia.close();
        }

    }
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============


=====================================================
```

PROGRAMA – B ou 2

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;
```

```
public class main {

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


//              // Criando Pilha Principal e Pilha Reversa
//              Stack PA = new Stack(); //PILHA PRINCIPAL
//              Stack PB = new Stack(); //PILHA REVERSA


            Queue filaA = new LinkedList();
            Queue filaB = new LinkedList();

            int QTD;
            int Num;
            int InicioPA = 0;
```

```
        int VarComp = 0 ;

        int cont = 0;
//=================================================================================================
//=================================================================================================

System.out.println("\n=======================================");
        System.out.printf("\nDigite a quantidade de números para enfileirar: ");

        QTD = leia.nextInt();

System.out.println("\n=======================================");

//=================================================================================================
//=================================================================================================

        while(cont < QTD){
```

```
            Num = leia.nextInt();



            //Se pilha A estiver vazia, então empilhar o primeiro
número
            if(filaA.isEmpty()) {
               filaA.add(Num);

               System.out.printf("\nPilha que estava vazia agora têm:");
               System.out.println(filaA);

System.out.println("================================");

            }



            //Senão
            else {



                  while(!filaA.isEmpty()) {

                     if(!filaA.isEmpty()) {
```

```
                InicioPA = (int) filaA.peek();   //----- Se a pilha A não estiver vazia, então é pq existe TOPO
            }                       //----- aí estou pegando o topo e jogando o valor dentro de uma variável



            if(InicioPA != Num ) {    //&& !PA.isEmpty()
               filaB.add(filaA.remove());

            }


            if(!filaA.isEmpty()) {
               InicioPA = (int) filaA.peek();
            }

            if(InicioPA == Num && !filaA.isEmpty()) {
               filaB.add(filaA.remove());
               VarComp = 1;
            }



        }
```

```
        //-----  Se a Variável de comparação estiver em 0 assim como começou, então é pq não teve número repetido
        //-----  se não têm número repetido, terei que enfileirá-lo
           if(VarComp == 0) {
              while(!filaB.isEmpty()) {
                 filaA.add(filaB.remove());  // desenfileira tudo da fila B e emenfileira na fila A
              }
              filaA.add(Num); // depois da fila toda enfileirada, enfileirar também o número digitado
           }


        //-----  Se a Variável de comparação estiver em 1 é pq essa variável foi alterada !!!!!!!!!
        //-----  e ela só é alterada quando encontra um início da fila A igual ao valor digitado !!!!!!!
           if(VarComp == 1) {
              while(!filaB.isEmpty()) {     // desenfilera tudo da fila B e enfilera na fila A
                 filaA.add(filaB.remove());
```

```
            }
        VarComp = 0;    //----- aqui a variável recebe novamente o valor que ela começou, pois se não,
        }               //----- ela iria sempre ficar em 1, assim, sempre acusando que teria número igual.
        }
        cont++;
        }

        System.out.println("Imprimindo as duas PILHAS");
        System.out.println(filaA);
        System.out.println(filaB);

//================================================================================
```

```
//=====================================================
=====================================================
===============


        leia.close();
        }

    }
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============
```

==========

PROGRAMA – C ou 3

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {
```

```
//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


//              // Criando Pilha Principal e Pilha Reversa
//              Stack PA = new Stack(); //PILHA PRINCIPAL
//              Stack PB = new Stack(); //PILHA REVERSA


            LinkedList listaA = new LinkedList();
            LinkedList listaB = new LinkedList();

            int QTD;
            int Num;
            int InicioFA = 0;

            int VarComp = 0 ;

            int cont = 0;
```

```
//=================================================================================================
//=================================================================================================

System.out.println("\n=======================================");
        System.out.printf("\nDigite a quantidade de números para inserir na Lista: ");

        QTD = leia.nextInt();

System.out.println("\n=======================================");

//=================================================================================================
//=================================================================================================

        while(cont < QTD){

        Num = leia.nextInt();
```

```
            //Se lista A estiver vazia, então inserir o primeiro número
            if(listaA.isEmpty()) {
               listaA.add(Num);

               System.out.printf("\nLista que estava vazia agora têm:");
               System.out.println(listaA);

System.out.println("================================");

            }



            //Senão
            else {



                  while(!listaA.isEmpty()) {

                     if(!listaA.isEmpty()) {
                        InicioFA = (int) listaA.peek();   //----- Se a lista A não
estiver vazia, então é pq existe primeiro índice
                           }                        //----- aí estou pegando o
índice e jogando o valor dentro de uma variável
```

```
    if(InicioFA != Num ) {    //&& !PA.isEmpty()
       listaB.add(listaA.remove());

    }


    if(!listaA.isEmpty()) {
       InicioFA = (int) listaA.peek();
    }

    if(InicioFA == Num && !listaA.isEmpty()) {
       listaB.add(listaA.remove());
       VarComp = 1;
    }



}
```

```
        //-----  Se a Variável de comparação estiver em 0 assim como começou, então é pq não teve número repetido

        //-----  se não têm número repetido, terei que enfileirá-lo

           if(VarComp == 0) {

              while(!listaB.isEmpty()) {

                 listaA.add(listaB.remove());  // desenfileira tudo da fila B e emenfileira na fila A

              }

              listaA.add(Num); // depois da fila toda enfileirada, enfileirar também o número digitado

           }



        //-----  Se a Variável de comparação estiver em 1 é pq essa variável foi alterada !!!!!!!!!

        //-----  e ela só é alterada quando encontra um início da fila A igual ao valor digitado !!!!!!!

           if(VarComp == 1) {

              while(!listaB.isEmpty()) {     // desenfilera tudo da fila B e enfilera na fila A

                 listaA.add(listaB.remove());

              }

           VarComp = 0;    //----- aqui a variável recebe novamente o valor que ela começou, pois se não,

           }               //----- ela iria sempre ficar em 1, assim, sempre acusando que teria número igual.
```

```
        }
        cont++;
        }


        System.out.println("Imprimindo as duas Listas");
        System.out.println(listaA);
        System.out.println(listaB);

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============

        leia.close();
        }
```

```
    }
//=========================================================
=========================================================
==============
//=========================================================
=========================================================
==============
```

==========

PROGRAMA – D ou 4

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {

//=========================================================
=========================================================
==============
//=========================================================
=========================================================
==============
```

```
        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando Pilha Principal e Pilha Reversa
            Stack PA = new Stack(); //PILHA PRINCIPAL

            Queue filaB = new LinkedList();

            int QTD;
            int cont = 0;
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============


System.out.println("\n=====================================
==========");
            System.out.printf("\nDigite a quantidade de números para
empilhar: ");

            QTD = leia.nextInt();
```

```
System.out.println("\n=====================================
==========");

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============

        while(cont < QTD){

        PA.push(leia.nextInt());

        cont++;

        }




System.out.println("================================");
        System.out.println("Imprimindo a Pilha");
        System.out.println(PA);

        cont = 0;
```

```
        int VarPA;
        while(cont < QTD) {

           filaB.add(PA.pop());

           cont++;
        }


        cont = 0;
        while(cont < QTD) {
           PA.push(filaB.remove());

           cont++;
        }



System.out.println("==================================");
        System.out.println("Imprimindo a Pilha invertida");
        System.out.println(PA);


//=================================================
================================================
===============
```

```
//=====================================================
=====================================================
===============


        leia.close();
        }

    }
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============
```

===

PROGRAMA – E ou 5

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {
```

```
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando Pilha Auxiliar
            Stack PB = new Stack(); //PILHA PRINCIPAL

            LinkedList listaA = new LinkedList();

            int QTD;
            int cont = 0;
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============
```

```
System.out.println("\n=======================================
==========");
        System.out.printf("\nDigite a quantidade de números para
colocar na lista: ");

        QTD = leia.nextInt();

System.out.println("\n=======================================
==========");

//=======================================================
=========================================================
===============
//=======================================================
=========================================================
===============

        while(cont < QTD){

           listaA.add(leia.nextInt());

        cont++;

        }
```

```
System.out.println("================================");
        System.out.println("Imprimindo a Lista");
        System.out.println(listaA);

        cont = 0;
        int VarPA;
        while(cont < QTD) {

           PB.push(listaA.remove());

           cont++;
        }


        cont = 0;
        while(cont < QTD) {
           listaA.add(PB.pop());

           cont++;
        }


System.out.println("================================");
        System.out.println("Imprimindo a Lista invertida");
```

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


//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============


            leia.close();
            }

        }
//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============


=========================================================
```

PROGRAMA – Fou 6

```
import java.util.LinkedList;
```

```
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {

//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando Pilha Principal e Pilha Reversa
            Stack PB = new Stack(); //PILHA PRINCIPAL

            Queue filaA = new LinkedList();

            int QTD;
            int cont = 0;
```

```
//=================================================================================================
//=================================================================================================

System.out.println("\n=======================================");
        System.out.printf("\nDigite a quantidade de números para enfileirar: ");

        QTD = leia.nextInt();

System.out.println("\n=======================================");

//=================================================================================================
//=================================================================================================

        while(cont < QTD){

        filaA.add(leia.nextInt());
```

```
            cont++;

            }

System.out.println("================================");
            System.out.println("Imprimindo a Fila");
            System.out.println(filaA);

            cont = 0;
            int VarPA;
            while(cont < QTD) {

               PB.push(filaA.remove());

               cont++;
            }


            cont = 0;
            while(cont < QTD) {
               filaA.add(PB.pop());

               cont++;
```

```
            }


System.out.println("==================================");
            System.out.println("Imprimindo a Fila invertida");
            System.out.println(filaA);


//=================================================
================================================
===============
//=================================================
================================================
===============


            leia.close();
            }

        }
//=================================================
================================================
===============
//=================================================
================================================
===============
```

==========

PROGRAMA – G ou 7

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {

//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando as Pilhas
            LinkedList FilaA = new LinkedList();
            LinkedList FilaB = new LinkedList();
```

```
        LinkedList FauxA = new LinkedList();
        LinkedList FauxAcomp = new LinkedList();

        int QTD;
        int cont = 0;
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============

System.out.println("\n=====================================
==========");
        System.out.printf("\nDigite a quantidade de números que
deseja colocar em cada Fila: ");

        QTD = leia.nextInt();

//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============
```

```
System.out.println("\n=====================================
==========");
        System.out.println("\nDigite os números da Fila A");


        while(cont < QTD){

           FilaA.add(leia.nextInt());

        cont++;

        }

//=====================================================
=================================================
===============
//=====================================================
=================================================
===============


System.out.println("\n=====================================
==========");
        System.out.println("\nDigite os números da Fila B");
```

```
            cont = 0;

            while(cont < QTD){

               FilaB.add(leia.nextInt());

            cont++;

            }

//=============================================================================================

//=============================================================================================


System.out.println("\n=======================================");
            System.out.println("Imprimindo a Fila A e Fila B");
            System.out.println(FilaA);
            System.out.println(FilaB);

System.out.println("\n=======================================");
```

```
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============
```

```
            cont = 0;

      while(!FilaB.isEmpty()) { //repita enquanto a PB não estiver vazia

         if(FilaA.peek() == FilaB.peek()) {
            FilaA.remove();
         }


         if(!FilaA.isEmpty() && !FilaB.isEmpty() && FilaA.peek() !=
FilaB.peek()) {

            FauxA.add(FilaA.remove());
         }
```

```
    if(FilaA.isEmpty()) {
        while(!FauxA.isEmpty()) {
            FilaA.add(FauxA.remove());
        }
        FilaA.add(FilaB.remove());
    }









        System.out.println(FilaA);




    cont++;
}



FauxAcomp.add(FilaA.remove());
```

```
    if(!FilaA.isEmpty()) {
       while(!FilaA.isEmpty()) {

          if(FilaA.peek() == FauxAcomp.peek()) {
             FilaA.remove();
          }

          if(!FilaA.isEmpty() && FilaA.peek() != FauxAcomp.peek()){
             FauxAcomp.add(FilaA.remove());
          }
       }
    }

    if(FilaA.isEmpty()){
       while(!FauxAcomp.isEmpty()){
          FilaA.add(FauxAcomp.remove());
       }
    }




//=================================================
================================================
===============
```

```
//=========================================================
=========================================================
================

System.out.println("\n=====================================
==========");
        System.out.println("Imprimindo a União de conjuntos da
Lista A com a Lista B");
        System.out.println(FilaA);
        System.out.println(FilaB);
        System.out.println(FauxA);
        System.out.println(FauxAcomp);

//=========================================================
=========================================================
================
//=========================================================
=========================================================
================

        leia.close();
        }

    }
```

//==========
==========
==========

//==========
==========
==========

==========

PROGRAMA – H ou 8

==========

PROGRAMA – I ou 9

==========

PROGRAMA – J ou 10

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;
```

```
public class main {

//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando as Pilhas
            Stack PA = new Stack();
            Stack PB = new Stack();
            Stack PauxA = new Stack();

            Stack PauxAcomp = new Stack();

            int QTD;
            int cont = 0;
//=========================================================
=========================================================
===============
```

```
//=================================================================================================

System.out.println("\n=========================================");

        System.out.printf("\nDigite a quantidade de números que deseja colocar em cada Pilha: ");

        QTD = leia.nextInt();

//=================================================================================================
//=================================================================================================

System.out.println("\n=========================================");

        System.out.println("\nDigite os números da Pilha A");

        while(cont < QTD){
```

```
            PA.push(leia.nextInt());

        cont++;

        }

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============



System.out.println("\n=================================
==========");
        System.out.println("\nDigite os números da Pilha B");

        cont = 0;

        while(cont < QTD){

            PB.push(leia.nextInt());

        cont++;

        }
```

```
//=================================================================================================

//=================================================================================================


System.out.println("\n=========================================");
            System.out.println("Imprimindo a Pilha A e Pilha B");
            System.out.println(PA);
            System.out.println(PB);

System.out.println("\n=========================================");

//=================================================================================================

//=================================================================================================


            cont = 0;
```

```
while(!PB.isEmpty()) { //repita enquanto a PB não estiver vazia

   if(PA.peek() == PB.peek()) {
      PA.pop();
   }



   if(!PA.isEmpty() && !PB.isEmpty() && PA.peek() != PB.peek()) {

      PauxA.push(PA.pop());
   }




   if(PA.empty()) {
      while(!PauxA.isEmpty()) {
         PA.push(PauxA.pop());
      }
      PA.push(PB.pop());
   }
```

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



    cont++;
}



PauxAcomp.push(PA.pop());

if(!PA.isEmpty()) {
    while(!PA.isEmpty()) {

        if(PA.peek() == PauxAcomp.peek()) {
            PA.pop();
        }

        if(!PA.isEmpty() && PA.peek() != PauxAcomp.peek()){
            PauxAcomp.push(PA.pop());
```

```
            }
        }
    }

    if(PA.isEmpty()){
        while(!PauxAcomp.isEmpty()){
            PA.push(PauxAcomp.pop());
        }
    }




//=====================================================
=====================================================
================
//=====================================================
=====================================================
================




//
//              while(!PB.isEmpty()) {
```

```
//
//
//          PA.push(PB.pop());
//
//
//          }



System.out.println("\n=====================================
==========");
            System.out.println("Imprimindo a União de conjuntos da
Pilha A com a Pilha B");
            System.out.println(PA);
            System.out.println(PB);
            System.out.println(PauxA);
            System.out.println(PauxAcomp);

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============


        leia.close();
```

```
        }

    }
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============
```

===

PROGRAMA – K ou 11

===

PROGRAMA – L ou 12

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {
```

```
//=========================================================
==========================================================
===============
//=========================================================
==========================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando as Pilhas
            Stack PA = new Stack();
            Stack PB = new Stack();
            Stack PC = new Stack();

            Stack PauxA = new Stack();

            int QTD;
            int cont = 0;

            int flag = 0;
//=========================================================
==========================================================
===============
```

```
//=================================================================================================

System.out.println("\n=========================================");

        System.out.printf("\nDigite a quantidade de números que deseja colocar em cada Pilha: ");

        QTD = leia.nextInt();

//=================================================================================================

//=================================================================================================

System.out.println("\n=========================================");

        System.out.println("\nDigite os números da Pilha A");

        while(cont < QTD){
```

```
            PA.push(leia.nextInt());

        cont++;

        }

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============



System.out.println("\n=================================
==========");
        System.out.println("\nDigite os números da Pilha B");

        cont = 0;

        while(cont < QTD){

            PB.push(leia.nextInt());

        cont++;

        }
```

```
//=================================================================================================

//=================================================================================================


System.out.println("\n=======================================");
            System.out.println("Imprimindo a Pilha A e Pilha B");
            System.out.println(PA);
            System.out.println(PB);

System.out.println("\n=======================================");

//=================================================================================================

//=================================================================================================


        while(!PB.isEmpty()) {
```

```
        if(!PA.isEmpty() && !PB.isEmpty() && PA.peek() ==
PB.peek()){
            PA.pop();
            PB.pop();
        }

        if(!PA.isEmpty() && !PB.isEmpty() && PA.peek() !=
PB.peek()){
            PauxA.push(PA.pop());
            System.out.println(PauxA);
        }

        if(PA.isEmpty()) {
            PC.push(PB.pop());
            System.out.println("\n=========");
            System.out.println(PC);
            System.out.println("=========");

            while(!PauxA.isEmpty()) {
                PA.push(PauxA.pop());
            }
        }
```

```
        }

        while(!PC.isEmpty()) {
            if(!PA.isEmpty() && !PC.isEmpty() && PA.peek() ==
PC.peek()) {
                PA.pop();
                PC.pop();
            }

            if(!PA.isEmpty() && !PC.isEmpty() && PA.peek() !=
PC.peek()) {
                PauxA.push(PA.pop());
//                  PC.pop();
            }


            if(PA.isEmpty()) {
                PC.pop();
            }

            System.out.println(PauxA);

            if(PA.isEmpty()) {
```

```
        while(!PauxA.isEmpty()) {


        PA.push(PauxA.pop());
        System.out.println("\nPA");
        System.out.println(PA);
        System.out.println("PA");
    }
    }

}
```

```
//       while(!PB.isEmpty()) {
//
//          if(!PB.isEmpty() && !PA.isEmpty() && PA.peek() !=
PB.peek()) {
//
//             PauxA.push(PA.pop());
//
//             System.out.println(PauxA);
//
//          }
//
//
//
//          if(!PB.isEmpty()&& !PA.isEmpty() && PA.peek() ==
PB.peek()) {
//             PA.pop();
//             PB.pop();
//
//
//
```

```
//          }
//    //=============================================
//          if(PA.isEmpty()) {
//              System.out.println(PB.peek());
//              PC.push(PB.pop());
//
//
//
//
//              while(!PauxA.isEmpty()) {
//                  PA.push(PauxA.pop());
//              }
//
//
//
//          }
//
//    //=========================================
//
//       }


       //while(!PB.isEmpty()) {
       //    if(!PB.isEmpty()) {
       //        if(PA.peek() == PB.peek()) {
```

```
//          //PC.push(PA.pop());
//          PA.pop();
//          PB.pop();
//
//      }
//
//
//      if(!PA.isEmpty() && !PB.isEmpty() && PA.peek() !=
PB.peek()) {
//          PauxA.push(PA.pop());
//      }
//
//      if(PA.isEmpty()) {
//          while(!PauxA.isEmpty()) {
//              PA.push(PauxA.pop());
//
//          }
//
//          PB.pop();
//
//      }
```

```
    //    }



    //if(PB.isEmpty() && !PC.isEmpty()) {

    //    while(!PC.isEmpty()){
    //        if(!PA.isEmpty() && PA.peek() == PC.peek()) {
    ///            PA.pop();
     //        }

    //        if(!PA.isEmpty() && PA.peek() != PC.peek()) {
       //         PB.push(PA.pop());
          //}

    //        if(PA.isEmpty()) {
    //            PC.pop();
    //        while(!PB.isEmpty()){
    //            PA.push(PB.pop());
    //        }
    //        }
    //    }
    //}

//    }
```

```
//    if(!PauxA.isEmpty()) {
      //while(!PauxA.isEmpty()) {
         //PA.push(PauxA.pop());
         //    }
      //       }

      if(!PauxA.isEmpty()) {
         while(!PauxA.isEmpty()){
            PA.push(PauxA.pop());
         }
      }



//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============



System.out.println("\n=====================================
==========");
            System.out.println("Imprimindo a diferença de conjuntos da
Pilha A com a Pilha B");
            System.out.println(PA);
```

```
            System.out.println(PB);
            System.out.println(PauxA);
            System.out.println(PC);

//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============


        leia.close();
        }

    }
//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============




=========================================================
```

PROGRAMA – M ou 13

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class main {

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============

        public static void main(String[] args) {

            Scanner leia = new Scanner(System.in);


            // Criando as Pilhas
            Queue FilaA = new LinkedList();
            Queue FilaB = new LinkedList();
            Queue FauxA = new LinkedList();
            Queue FauxAcomp = new LinkedList();
```

```
        int QTD;
        int cont = 0;
//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============

System.out.println("\n=====================================
==========");
        System.out.printf("\nDigite a quantidade de números que
deseja colocar em cada Fila: ");

        QTD = leia.nextInt();

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============
```

```
System.out.println("\n=====================================
==========");
        System.out.println("\nDigite os números da Fila A");


        while(cont < QTD){

           FilaA.add(leia.nextInt());

        cont++;

        }

//=====================================================
=============================================================
===============
//=====================================================
=============================================================
===============


System.out.println("\n=====================================
==========");
        System.out.println("\nDigite os números da Fila B");

        cont = 0;
```

```
            while(cont < QTD){

               FilaB.add(leia.nextInt());

            cont++;

            }

//=====================================================
=====================================================
===============
//=====================================================
=====================================================
===============


System.out.println("\n=================================
==========");
            System.out.println("Imprimindo a Fila A e Fila B");
            System.out.println(FilaA);
            System.out.println(FilaB);

System.out.println("\n=================================
==========");
```

```
//=========================================================
=========================================================
===============
//=========================================================
=========================================================
===============



        cont = 0;

    while(!FilaB.isEmpty()) { //repita enquanto a PB não estiver vazia

        if(FilaA.peek() == FilaB.peek()) {
            FilaA.remove();
        }



        if(!FilaA.isEmpty() && !FilaB.isEmpty() && FilaA.peek() !=
FilaB.peek()) {

            FauxA.add(FilaA.remove());
        }
```

```
    if(FilaA.isEmpty()) {
        while(!FauxA.isEmpty()) {
            FilaA.add(FauxA.remove());
        }
        FilaA.add(FilaB.remove());
    }









        System.out.println(FilaA);




    cont++;
}



FauxAcomp.add(FilaA.remove());
```

```
        if(!FilaA.isEmpty()) {
            while(!FilaA.isEmpty()) {

                if(FilaA.peek() == FauxAcomp.peek()) {
                    FilaA.remove();
                }

                if(!FilaA.isEmpty() && FilaA.peek() != FauxAcomp.peek()){
                    FauxAcomp.add(FilaA.remove());
                }
            }
        }

        if(FilaA.isEmpty()){
            while(!FauxAcomp.isEmpty()){
                FilaA.add(FauxAcomp.remove());
            }
        }




//================================================================================
```

```
//=========================================================
=========================================================
================

System.out.println("\n=====================================
==========");
        System.out.println("Imprimindo a União de conjuntos da
Fila A com a Fila B");
        System.out.println(FilaA);
        System.out.println(FilaB);
        System.out.println(FauxA);
        System.out.println(FauxAcomp);

//=========================================================
=========================================================
================
//=========================================================
=========================================================
================

        leia.close();
        }

    }
```

//==========
==========
==========

//==========
==========
==========

==========

PROGRAMA – N ou 14

==========

PROGRAMA – O ou 15