```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this t
 */
package com.mycompany.arra;

/**
 *
 * @author Renato G Rojas
 */
public class ArrA {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int lim;
        int op;
        int num;
        int ev = 0;
        int encontrado;

        Metodos_ArrA obj = new Metodos_ArrA();

        System.out.println("Ingrese el tamaño del arreglo: ");
        lim = obj.entrada.nextInt();

        int[] myArreglo = new int [lim];

        do{
            op = obj.menu();
            switch(op) {
                case 1 -> ev = obj.insFull(myArreglo);
                case 2 -> ev = obj.insFinal(myArreglo, ev);
                case 3 -> ev = obj.insInicio(myArreglo, ev);
                case 4 -> {
                            System.out.println("Ingrese el numero:");
                            num = obj.entrada.nextInt();
                            ev = obj.insOrden(myArreglo, ev, num);
                }
```

```
                case 5 -> {
                    System.out.println("Numero a buscar:");
                    num = obj.entrada.nextInt();
                    encontrado = obj.secuencial(myArreglo, ev, num);
                    if(encontrado != 1){
                        System.out.println("Esta en la posicion: " + encontrado);
                    }
                }
                case 6 -> obj.presenta(myArreglo, ev);
                default -> {}
            }
        } while (op != 0);
    }

}
```

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
 */
package com.mycompany.arra;

import java.util.Scanner;

/**
 *
 * @author renat
 */
public class Metodos_ArrA {

    Scanner entrada;

    public Metodos_ArrA() {
        this.entrada = new Scanner(System.in);
    }

    public int menu() {
        System.out.println("Insertar completo   [1]:");
        System.out.println("Insertar al final   [2]:");
        System.out.println("Insertar al inicio  [3]:");
        System.out.println("Insertar ordenado   [4]:");
        System.out.println("Buscar              [5]:");
        System.out.println("Presentar           [6]:");
        System.out.println("Salir               [0]:");
        return entrada.nextInt();
    }

    public int insFull(int[] X) {
        for (int i = 0; i < X.length; i++) {
            System.out.println("Ingrese elemento [" + i + "]:");
            X[i] = entrada.nextInt();
        }
        return X.length;
    }

    public void presenta(int[] X, int _ev) {
        for (int i = 0; i < _ev; i++) {
```

```
            System.out.println(X[i] + " ");
        }
        System.out.println("");
    }

    public int insFinal(int[] X, int _ev) {
        if (_ev < X.length) {
            System.out.println("Ingrese elemento:");
            X[_ev] = entrada.nextInt();
        } else {
            System.out.println("Arreglo Lleno...");
        }
        return _ev;
    }

    public void recorreDer(int[] X, int pos, int _ev){
        for(int i = _ev; i > pos; i--){
            X[i] = X[i-1];
        }
    }

    public int insInicio(int[] X, int _ev) {
        if(_ev < X.length){
            recorreDer(X, 0, _ev);
            X[0] = entrada.nextInt();
            _ev++;
        } else {
            System.out.println("Arrelgo lleno...");
        }
        return _ev;
    }

    public int insOrden(int[] X, int _ev, int num) {
        if(_ev < X.length) {
            int pos = 0;
            while((pos < _ev) && (num > X[pos]))
                pos++;
            if(pos != _ev)
                recorreDer(X, pos, _ev);
            X[pos] = num;
            _ev++;
```

```
        } else {
                System.out.println("Arreglo lleno...");
        }
        return _ev;
    }

    public int secuencial(int[] X, int _ev, int num){
        for (int i = 0; i < _ev; i++) {
            if (num == X[i]){
                return i;
            }
        }
        return -1;
    }

    public void recorrerIzq(int[] X, int _ev, int pos){
        for (int i = pos; i < _ev-1; i++) {
            X[i] = X[ i + 1 ];
        }
    }

    public int elimina (int[] X, int _ev, int num) {
        int pos = secuencial (X, _ev, num);
        if (pos != -1) {
        recorrerIzq (X, _ev, pos);
        _ev --;
        } else {
            System.out.println("Elemento no existente...");
        }
        return _ev;
    }

    public void burbuja(int[] X, int _ev) {
        int aux = 0;
        for (int i = 0; i < _ev-1; i++) {
            for (int j = i + 1; j < _ev; j++) {
                if (X[i] > X[j])
                    aux = X[i];
                    X[i] = X[j];
                    X[j] = aux;
            }
```

```
        }
    }

    public int binaria (int[] X, int ini, int fin, int num){
        int medio = (ini + fin) / 2;
        if (ini <= fin){
            if (X[medio] == num){
                return medio;
            } else {
                if(num > X[medio]){
                return binaria(X, medio+1, fin, num);
                } else {
                    return binaria (X, ini, medio -1, num);
                }
            }
        } else {
            return -1;
        }
    }
}
```

Start Page | ArrA.java | Metodos_ArrA.java

Source | History

```
4   */
5   package com.mycompany.arra;
6
7   /**
8    *
9    * @author Renato G Rojas
10   */
11  public class ArrA {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18          int lim;
19          int op;
20          int num;
21          int ev = 0;
22          int encontrado;
23
24          Metodos_ArrA obj = new Metodos_ArrA();
25
26          System.out.println("Ingrese el tamaño del arreglo: ");
27          lim = obj.entrada.nextInt();
28
29          int[] myArreglo = new int [lim];
30
31          do{
32              op = obj.menu();
33              switch(op) {
34                  case 1 -> ev = obj.insFull(myArreglo);
35                  case 2 -> ev = obj.insFinal(myArreglo, ev);
36                  case 3 -> ev = obj.insInicio(myArreglo, ev);
37                  case 4 -> {
38                                System.out.println("Ingrese el numero:");
39                                num = obj.entrada.nextInt();
```

```
39                          num = obj.entrada.nextInt();
40                          ev = obj.insOrden(myArreglo, ev, num);
41                  }
42                  case 5 -> {
43                      System.out.println("Numero a buscar:");
44                      num = obj.entrada.nextInt();
45                      encontrado = obj.secuencial(myArreglo, ev, num);
46                      if(encontrado != 1){
47                          System.out.println("Esta en la posicion: " + encontrado);
48                      }
49                  }
50                  case 6 -> obj.presenta(myArreglo, ev);
51                  default -> {}
52              }
53          } while (op != 0);
54      }
55
56  }
57
```

Start Page | ArrA.java | Metodos_ArrA.java

Source | History

```java
package com.mycompany.arra;

import java.util.Scanner;

/**
 *
 * @author renat
 */
public class Metodos_ArrA {

    Scanner entrada;

    public Metodos_ArrA() {
        this.entrada = new Scanner(System.in);
    }

    public int menu() {
        System.out.println("Insertar completo   [1]:");
        System.out.println("Insertar al final   [2]:");
        System.out.println("Insertar al inicio  [3]:");
        System.out.println("Insertar ordenado   [4]:");
        System.out.println("Buscar              [5]:");
        System.out.println("Presentar           [6]:");
        System.out.println("Salir               [0]:");
        return entrada.nextInt();
    }

    public int insFull(int[] X) {
        for (int i = 0; i < X.length; i++) {
            System.out.println("Ingrese elemento [" + i + "]:");
            X[i] = entrada.nextInt();
        }
        return X.length;
    }

    public void presenta(int[] X, int _ev) {
```

Start Page | ArrA.java | Metodos_ArrA.java

Source | History

```
40      public void presenta(int[] X, int _ev) {
41          for (int i = 0; i < _ev; i++) {
42              System.out.println(X[i] + " ");
43          }
44          System.out.println("");
45      }
46
47      public int insFinal(int[] X, int _ev) {
48          if (_ev < X.length) {
49              System.out.println("Ingrese elemento:");
50              X[_ev] = entrada.nextInt();
51          } else {
52              System.out.println("Arreglo Lleno...");
53          }
54          return _ev;
55      }
56
57      public void recorreDer(int[] X, int pos, int _ev){
58          for(int i = _ev; i > pos; i--){
59              X[i] = X[i-1];
60          }
61      }
62
63      public int insInicio(int[] X, int _ev) {
64          if(_ev < X.length){
65              recorreDer(X, 0, _ev);
66              X[0] = entrada.nextInt();
67              _ev++;
68          } else {
69              System.out.println("Arrelgo lleno...");
70          }
71          return _ev;
72      }
73
74      public int insOrden(int[] X, int _ev, int num) {
75          if(_ev < X.length) {
```

Start Page | ArrA.java | Metodos_ArrA.java

Source | History

```
74      public int insOrden(int[] X, int _ev, int num) {
75          if(_ev < X.length) {
76              int pos = 0;
77              while((pos < _ev) && (num > X[pos]))
78                  pos++;
79              if(pos != _ev)
80                  recorreDer(X, pos, _ev);
81              X[pos] = num;
82              _ev++;
83          } else {
84                  System.out.println("Arreglo lleno...");
85          }
86          return _ev;
87      }
88
89      public int secuencial(int[] X, int _ev, int num){
90          for (int i = 0; i < _ev; i++) {
91              if (num == X[i]){
92                  return i;
93              }
94          }
95          return -1;
96      }
97
98      public void recorrerIzq(int[] X, int _ev, int pos){
99          for (int i = pos; i < _ev-1; i++) {
100             X[i] = X[ i + 1 ];
101         }
102     }
103
104     public int elimina (int[] X, int _ev, int num) {
105         int pos = secuencial (X, _ev, num);
106         if (pos != -1) {
107         recorrerIzq (X, _ev, pos);
108         _ev --;
109         } else {
```

Start Page | ArrA.java | Metodos_ArrA.java

Source | History

```
108             _ev --;
109             } else {
110                 System.out.println("Elemento no existente...");
111             }
112             return _ev;
113         }
114
115         public void burbuja(int[] X, int _ev) {
116             int aux = 0;
117             for (int i = 0; i < _ev-1; i++) {
118                 for (int j = i + 1; j < _ev; j++) {
119                     if (X[i] > X[j])
120                         aux = X[i];
                        X[i] = X[j];
122                     X[j] = aux;
123                 }
124             }
125         }
126
127         public int binaria (int[] X, int ini, int fin, int num){
128             int medio = (ini + fin) / 2;
129             if (ini <= fin){
130                 if (X[medio] == num){
131                     return medio;
132                 } else {
133                     if(num > X[medio]){
134                     return binaria(X, medio+1, fin, num);
135                     } else {
136                         return binaria (X, ini, medio -1, num);
137                     }
138                 }
139             } else {
140                 return -1;
141             }
142         }
143     }
```

```
Ingrese el tamaño del arreglo:
5
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
1
Ingrese elemento [0]:
1
Ingrese elemento [1]:
2
Ingrese elemento [2]:
3
Ingrese elemento [3]:
4
Ingrese elemento [4]:
5
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
2
Arreglo Lleno...
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
3
Arrelgo lleno...
Insertar completo   [1]:
```

Output - Run (ArrA)

```
Presentar           [6]:
Salir               [0]:
3
Arrelgo lleno...
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
4
Ingrese el numero:
2
Arreglo lleno...
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
5
Numero a buscar:
3
Esta en la posicion: 2
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
5
Numero a buscar:
8
Esta en la posicion: -1
Insertar completo   [1]:
Insertar al final   [2]:
```

Output - Run (ArrA)

```
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
5
Numero a buscar:
8
Esta en la posicion: -1
Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
6
1
2
3
4
5

Insertar completo   [1]:
Insertar al final   [2]:
Insertar al inicio  [3]:
Insertar ordenado   [4]:
Buscar              [5]:
Presentar           [6]:
Salir               [0]:
0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  01:48 min
Finished at: 2023-05-16T13:38:42-05:00
------------------------------------------------------------------------
```