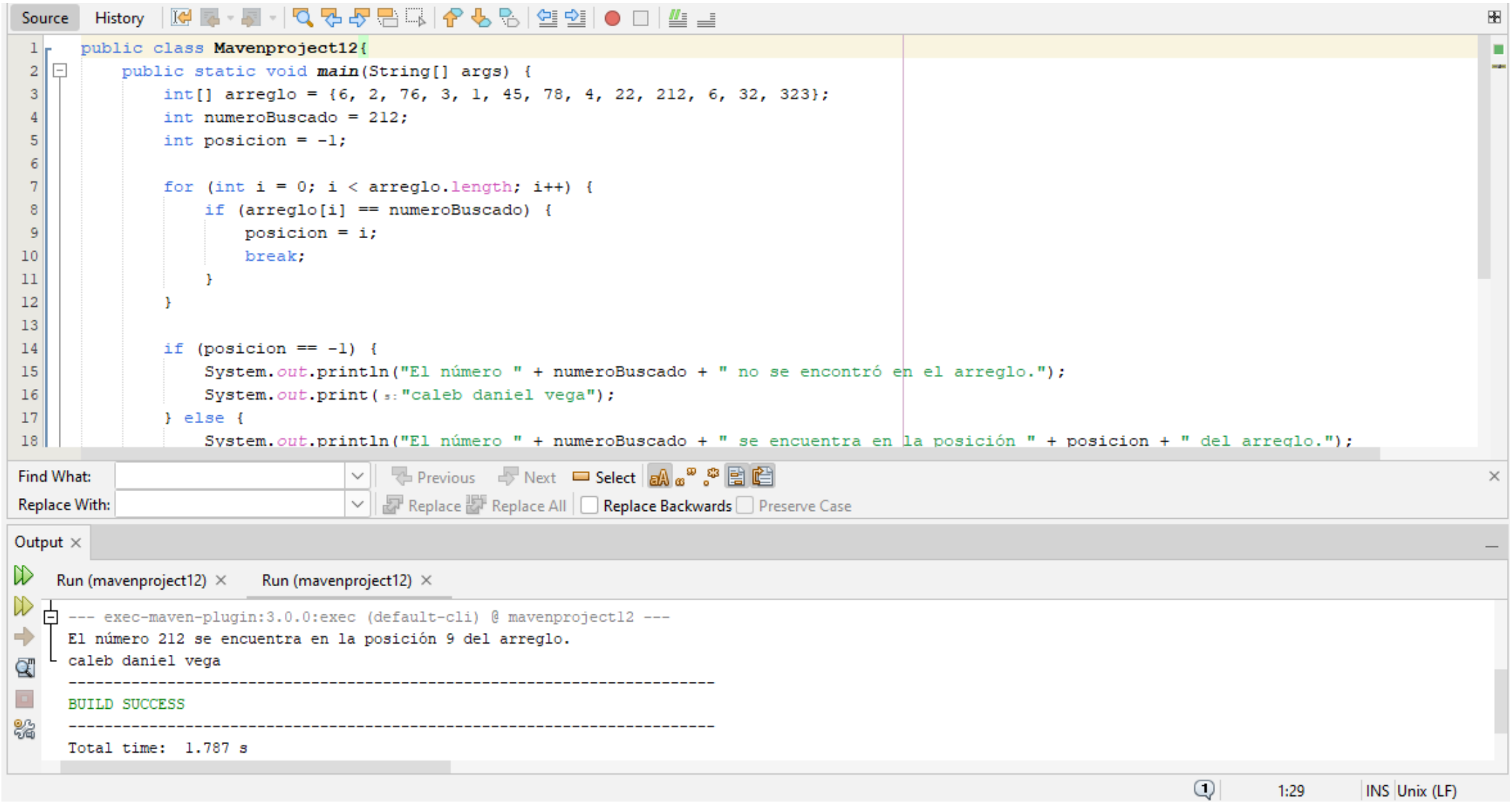
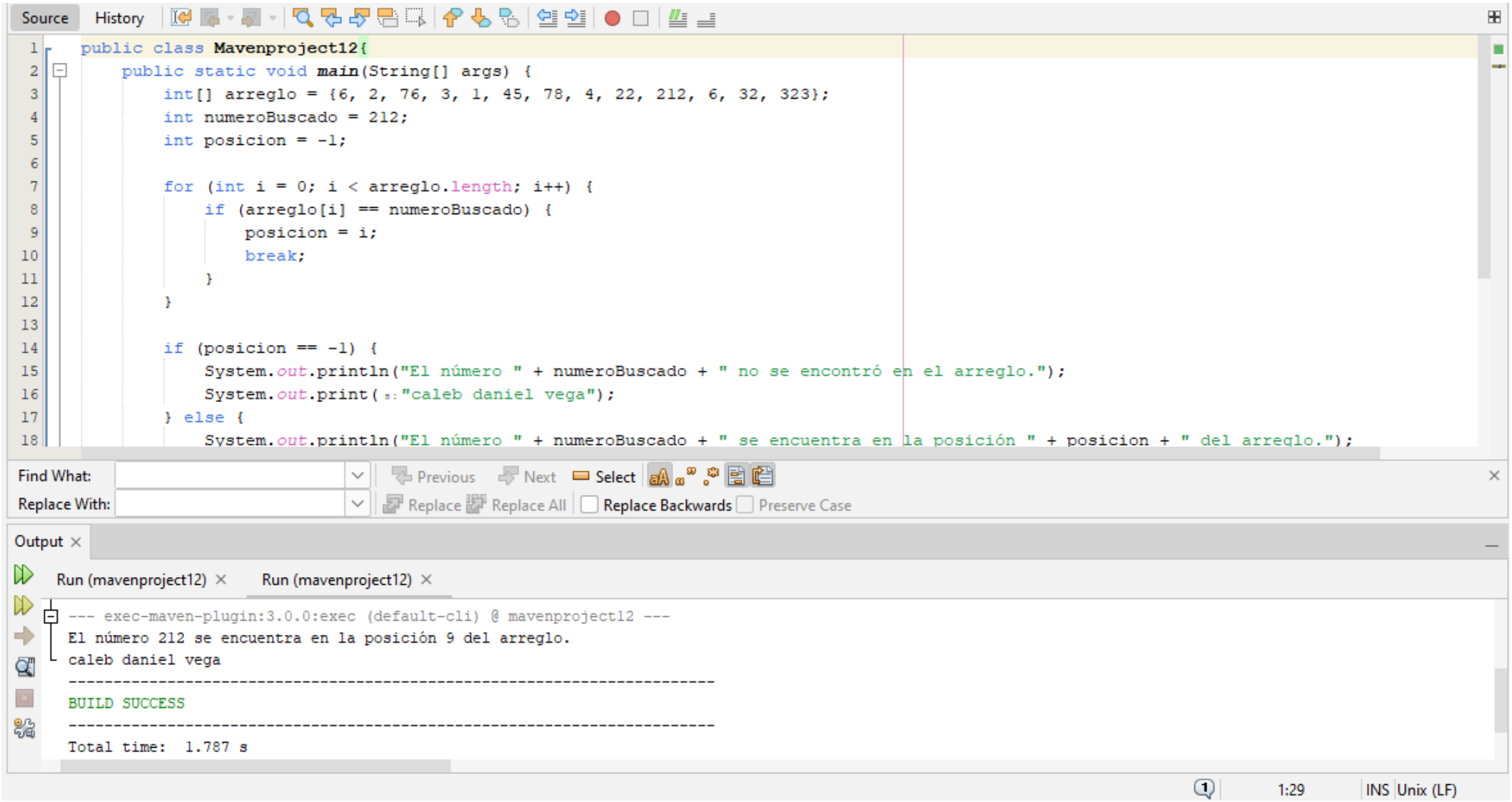

Source
History
public class Mavenproject12{
public static void main(String[] args) {
int[] arreglo = {6, 2, 76, 3, 1, 45, 78, 4, 22, 212, 6, 32, 323};
int numeroBuscado = 212;
int posicion = -1;
for (int i = 0; i < arreglo.length; i++) {
if (arreglo[i] == numeroBuscado) {
posicion = i;
break;
}
}
if (posicion == -1) {
System.out.println("El número " + numeroBuscado + " no se encontró en el arreglo.");
System.out.print( s: "caleb daniel vega");
} else {
System.out.println("El número " + numeroBuscado + " se encuentra en la posición " + posicion + " del arreglo.");
Find What:
Previous
Next
Select
Replace With:
Replace
Replace All
Replace Backwards
Preserve Case
Output
Run (mavenproject12)
Run (mavenproject12)
--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject12 ---
El número 212 se encuentra en la posición 9 del arreglo.
caleb daniel vega
BUILD SUCCESS
Total time: 1.787 s
1:29
INS
Unix (LF)


Source
History
1 public class Mavenproject12{
2 public static void main(String[] args) {
3 int[] arreglo = {6, 2, 76, 3, 1, 45, 78, 4, 22, 212, 6, 32, 323};
4 int numeroBuscado = 212;
5 int posicion = -1;
6
7 for (int i = 0; i < arreglo.length; i++) {
8 if (arreglo[i] == numeroBuscado) {
9 posicion = i;
10 break;
11 }
12 }
13
14 if (posicion == -1) {
15 System.out.println("El número " + numeroBuscado + " no se encontró en el arreglo.");
16 System.out.print(s:"caleb daniel vega");
17 } else {
18 System.out.println("El número " + numeroBuscado + " se encuentra en la posición " + posicion + " del arreglo.");
Find What:
Replace With:
Previous
Next
Select
Replace
Replace All
Replace Backwards
Preserve Case
Output
Run (mavenproject12)
Run (mavenproject12)
--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject12 ---
El número 212 se encuentra en la posición 9 del arreglo.
caleb daniel vega
BUILD SUCCESS
Total time: 1.787 s
1:29
INS
Unix (LF)

Source History

```java
public class Mavenproject11 {
    public static void main(String[] args) {
        int[] arreglo1 = {1, 3, 5, 7, 9};
        int[] arreglo2 = {2, 4, 6, 8, 10};
        int[] resultado = intercalar(arreglo1, arreglo2);
        for (int i = 0; i < resultado.length; i++) {
            System.out.print(resultado[i] + " ");
            System.out.print( s: "\nCaleb Daniel Vega");
        }
    }

    public static int[] intercalar(int[] arreglo1, int[] arreglo2) {
        int[] resultado = new int[arreglo1.length + arreglo2.length];
        int indice1 = 0;
        int indice2 = 0;
        for (int i = 0; i < resultado.length; i++) {
            if (indice1 < arreglo1.length && indice2 < arreglo2.length) {
                if (arreglo1[indice1] < arreglo2[indice2]) {
                    resultado[i] = arreglo1[indice1];
                    indice1++;
```

Output - Run (mavenproject11)

```
Caleb Daniel Vega9
Caleb Daniel Vega10
Caleb Daniel Vega------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  12.175 s
Finished at: 2023-04-16T16:38:13-05:00
------------------------------------------------------------------------
```

Unpacking index for Central Repository

10:6 INS Unix (LF)

Start Page | Mavenproject9.java | Mavenproject10.java | Mavenproject11.java | Mavenproject12.java | **Mavenproject13.java**

Source | History

```
22
23      public static int[] dijkstra(int[][] graph, int start) {
24          int n = graph.length;
25          int[] distances = new int[n];
26          Arrays.fill( a: distances,  val: Integer.MAX_VALUE);
27          distances[start] = 0;
28
29          PriorityQueue<Node> pq = new PriorityQueue<>();
30          pq.offer(new Node( id: start,  distance: 0));
31
32          while (!pq.isEmpty()) {
33              Node curr = pq.poll();
34
35              for (int i = 0; i < n; i++) {
36                  int distance = graph[curr.id][i];
37                  if (distance > 0 && curr.distance + distance < distances[i]) {
38                      distances[i] = curr.distance + distance;
39                      pq.offer(new Node( id: i, distances[i]));
40                  }
41              }
42          }
```

Output - Run (mavenproject13)

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject13 ---
[0, 4, 12, 19, 21, 11, 9, 8, 14]
Caleb daniel vega gonzalez
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.237 s
```

58:56 INS

Start Page | Mavenproject9.java | Mavenproject10.java | Mavenproject11.java | Mavenproject12.java | Mavenproject13.java

Source | History

```
37                  if (distance > 0 && curr.distance + distance < distances[i]) {
38                      distances[i] = curr.distance + distance;
39                      pq.offer(new Node( id:i, distances[i]));
40                  }
41              }
42          }
43
44          return distances;
45      }
46
47      public static void main(String[] args) {
48          int[][] graph = {
49                  {0, 7, 0, 5, 0, 0, 0},
50                  {7, 0, 8, 9, 7, 0, 0},
51                  {0, 8, 0, 0, 5, 0, 0},
52                  {5, 9, 0, 0, 12, 6, 0},
53                  {0, 7, 5, 12, 0, 8, 9},
54                  {0, 0, 0, 6, 8, 0, 11},
                    {0, 0, 0, 0, 9, 11, 0},
56          };
57          int[] distances = dijkstra(graph,  start:0);
```

Output - Run (mavenproject13)

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject13 ---
[0, 7, 15, 5, 14, 11, 22]
Caleb daniel vega gonzalez
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.037 s
```