JAVA.UTIL.ARRAYS.BINARYSEARCH METHOD

Description

The **java.util.Arrays.binarySearch***char[]a, charkey* method searches the specified array of chars for the specified value using the binary search algorithm.The array must be sorted before making this call.If it is not sorted, the results are undefined.

Declaration

Following is the declaration for **java.util.Arrays.binarySearch** method

```
public static int binarySearch(char[] a, char key)
```

Parameters

- **a** -- This is the array to be searched.
- **key** -- This is the value to be searched for.

Return Value

This method returns index of the search key, if it is contained in the array, else it returns $-(insertionpoint - 1)$. The insertion point is the point at which the key would be inserted into the array: the index of the first element greater than the key, or a.length if all elements in the array are less than the specified key.

Exception

- **NA**

Example

The following example shows the usage of java.util.Arrays.binarySearch method.

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

   public static void main(String[] args) {

   // initializing unsorted char array
   char charArr[] = {'a', 'c', 'b', 'e','d'};

   // sorting array
   Arrays.sort(charArr);

   // let us print all the elements available in list
   System.out.println("The sorted char array is:");
   for (char number : charArr) {
   System.out.println("Number = " + number);
   }

   // entering the value to be searched
   char searchVal = 'e';

   int retVal = Arrays.binarySearch(charArr, searchVal);

   System.out.println("The index of e is : " + retVal);
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
The sorted char array is:
Number = a
Number = b
Number = c
Number = d
Number = e
The index of e is : 4
```