|  |  |
| --- | --- |
| Student Name | Asadullah |
| Roll Number | 21SW036 |
| Section # | 03 |
| Lab # | 03 |

**Task#01**

Question statement

Implement linear search on 1D and 2D array.

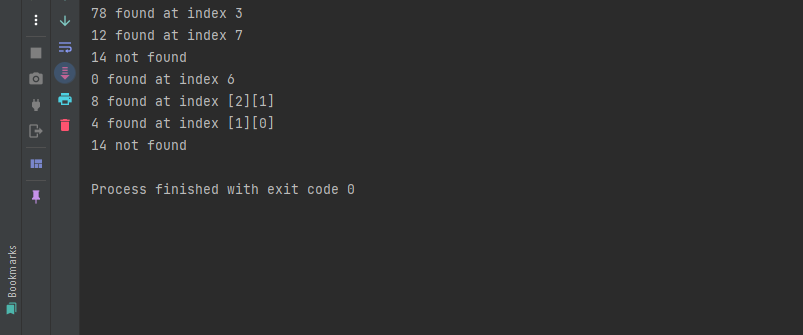
Note: create methods for both 1D and 2D linear search

# LinearSearch.Java

**Code:**

public class LinearSearch {  
  
 public static void linearSearch1D(int [] array, int element){  
 int index = 0;  
 boolean flag = false;  
 for (int i=0; i< array.length; i++){  
 if(array[i]==element){  
 flag = true;  
 index = i;  
 }  
 }  
 if(flag){  
 System.*out*.println(element+" found at index "+index);  
 } else {  
 System.*out*.println(element+" not found");  
 }  
 } // end of linearSearch1D() method  
  
 public static void linearSearch2D(int [][] array, int element) {  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 if (array[i][j] == element) {  
 System.*out*.println(element+" found at index ["+i+"]["+j+"]");  
 return;  
 }  
 }  
 }  
 System.*out*.println(element+" not found");  
 } // end of linearSearch2D  
  
 public static void main(String[] args) {  
  
 int [] array = {4, 5, 6, 78, 98, -5, 0, 12, 25, -8};  
 *linearSearch1D*(array, 78);  
 *linearSearch1D*(array, 12);  
 *linearSearch1D*(array, 14);  
 *linearSearch1D*(array, 0);  
  
 int [][] matrix = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
 *linearSearch2D*(matrix, 8);  
 *linearSearch2D*(matrix, 4);  
 *linearSearch2D*(matrix, 14);  
  
 } // end of main() method  
} // end of program

**Output:**

****

**Task#02**

Question statement

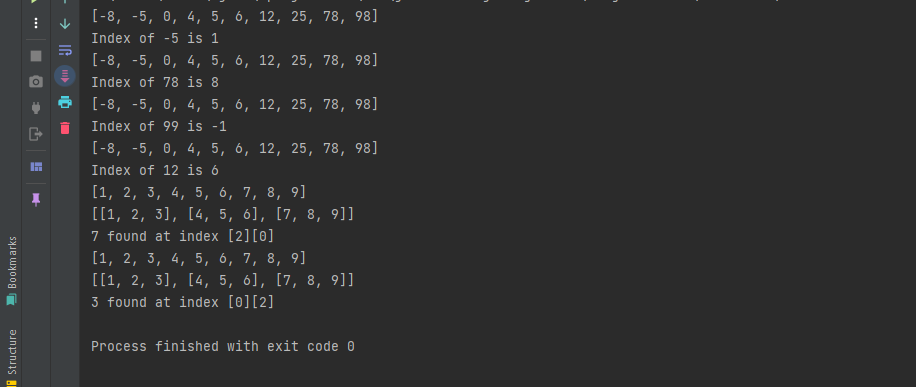
Implement binarysearch on 1D and 2D array.

Note: create methods for both 1D and 2D binarysearch

# BinarySearch.Java

**Code:**

import java.util.Arrays;  
  
public class BinarySearch {  
  
 public static int binarySearch1D(int [] arr, int target){  
 Arrays.*sort*(arr);  
 System.*out*.println(Arrays.*toString*(arr));  
 int p = 0, q=arr.length-1, i = 0;  
 while (p<=q){  
 i= (p+q)/2;  
 if(arr[i]==target){  
 return i;  
 }  
  
 if(arr[i]<target){  
 p = i+1;  
 } else {  
 q = i-1;  
 }  
 }  
 return -1;  
 } // end of binarySearch1D() method  
  
 public static void binarySearch2D(int[][] array, int key) {  
  
 // Conversion of 2D array into 1D array  
 int [] newArray = new int[array.length\*array[0].length];  
  
 int j=0, k=0;  
 for (int i=0; i< newArray.length; i++){  
 if(j>2) {  
 j = 0;  
 k++;  
 }  
 newArray[i] = array[k][j];  
 j++;  
 }  
  
 // Call binary Search method of 1D array  
 int arr = *binarySearch1D*(newArray, key);  
  
 // Convert 1D array back to 2D array  
 int [][] matrix = *conversionIn2D*(newArray, 3, 3);  
 System.*out*.println(Arrays.*deepToString*(matrix));  
 // Switch statement case checking for arr index retrieved from binarySearch()  
 switch (arr){  
 case 0 -> System.*out*.println(key+" found at index [0][0]");  
 case 1 -> System.*out*.println(key+" found at index [0][1]");  
 case 2 -> System.*out*.println(key+" found at index [0][2]");  
 case 3 -> System.*out*.println(key+" found at index [1][0]");  
 case 4 -> System.*out*.println(key+" found at index [1][1]");  
 case 5 -> System.*out*.println(key+" found at index [1][2]");  
 case 6 -> System.*out*.println(key+" found at index [2][0]");  
 case 7 -> System.*out*.println(key+" found at index [2][1]");  
 case 8 -> System.*out*.println(key+" found at index [2][2]");  
 } // end of switch statement  
 } // end of binarySearch2D() method  
  
 public static int[][] conversionIn2D(int [] array, int rows, int cols){  
 int[][] matrix = new int[rows][cols];  
 int k = 0;  
 for (int i=0; i< rows; i++){  
 for (int j=0; j< cols; j++){  
 matrix[i][j] = array[k];  
 k++;  
 }  
 }  
 return matrix;  
 } // end of conversionIn2D() method  
  
  
 public static void main(String[] args) {  
  
 int [] array = {4, 5, 6, 78, 98, -5, 0, 12, 25, -8};  
 System.*out*.println("Index of -5 is "+*binarySearch1D*(array, -5));  
 System.*out*.println("Index of 78 is "+*binarySearch1D*(array, 78));  
 System.*out*.println("Index of 99 is "+*binarySearch1D*(array, 99));  
 System.*out*.println("Index of 12 is "+*binarySearch1D*(array, 12));  
  
 int [][] matrix = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
 *binarySearch2D*(matrix, 7);  
 *binarySearch2D*(matrix, 3);  
  
 } // end of main() method  
} // end of program

**Output: **

**Task#03**

Question statement

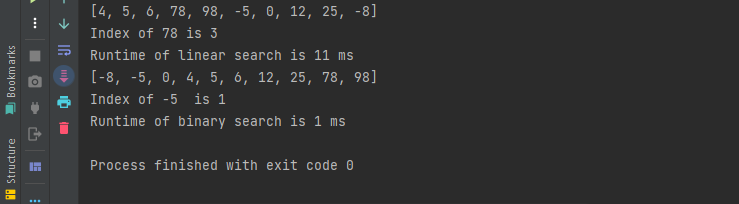
Display the execution time of searching algo (linear and binary both) and examine which one is the faster and explain why?

# ExecutionTimeOfAlgos.Java

**Code:**

import java.util.Arrays;  
  
public class ExecutionTimeOfAlgos {  
  
 public static int linearSearch(int [] array, int element){  
 long diff;  
 long before = System.*currentTimeMillis*();  
 int index = 0;  
 boolean flag = false;  
 for (int i=0; i< array.length; i++){  
 if(array[i]==element){  
 flag = true;  
 index = i;  
 }  
 }  
 if(flag){  
 return index;  
 } else {  
 return -1;  
 }  
  
 } // end of linearSearch  
  
 public static int binarySearch(int [] arr, int target){  
 Arrays.*sort*(arr);  
 System.*out*.println(Arrays.*toString*(arr));  
 int p = 0, q=arr.length-1, i = 0;  
 while (p<=q){  
 i= (p+q)/2;  
 if(arr[i]==target){  
 return i;  
 }  
  
 if(arr[i]<target){  
 p = i+1;  
 } else {  
 q = i-1;  
 }  
 }  
 return -1;  
 }  
  
  
  
 public static void main(String[] args) {  
  
 int [] array = {4, 5, 6, 78, 98, -5, 0, 12, 25, -8};  
 System.*out*.println(Arrays.*toString*(array));  
 long before = System.*currentTimeMillis*();  
 System.*out*.println("Index of 78 is "+*linearSearch*(array, 78));  
 long after = System.*currentTimeMillis*();  
 long runtime = after - before;  
 System.*out*.println("Runtime of linear search is "+runtime+" ms");  
  
 long before1 = System.*currentTimeMillis*();  
 System.*out*.println("Index of -5 is "+*binarySearch*(array, -5));  
 long after1 = System.*currentTimeMillis*();  
 long runtime1 = after1 - before1;  
 System.*out*.println("Runtime of binary search is "+runtime1+" ms");  
  
 } // end of main() method  
} // end of program

**Output:**

****