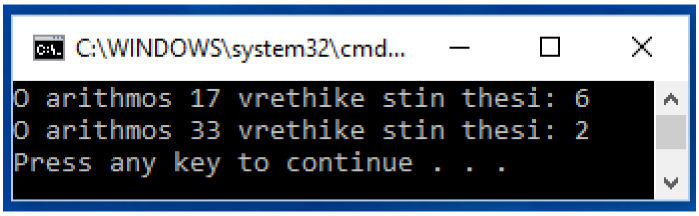
ΠΡΟΤΕΙΝΟΜΕΝΕΣ
ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ

Αναδρομή *(Recursion)*

ΑΣΚΗΣΗ-1η

Σειριακή ή γραμμική αναζήτηση αριθμού σε πίνακα ακεραίων

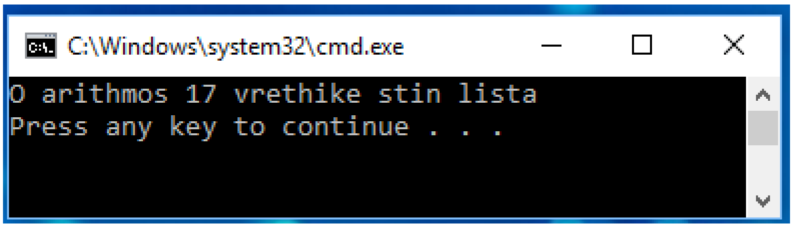
```
public class SequentialSearch {
   public static int linerSearch(int[] arr, int key){
      int size = arr.length;
      for(int i=0;i<size;i++){
         if(arr[i] == key){
            return i; }
      }
      return -1;}

   public static void main(String a[]){
      int[] arr1= {2,35,11,45,80,12,17,44};
      int searchKey = 17;
      System.out.println("O arithmos "+searchKey+" vrethike stin thesi: "+linerSearch(arr1,
               searchKey));
      int[] arr2= {12,51,33,55,134,13,74,56};
      searchKey = 33;
      System.out.println("O arithmos "+searchKey+" vrethike stin thesi: "+linerSearch(arr2,
                searchKey)); } }
```



```
O arithmos 17 vrethike stin thesi: 6
O arithmos 33 vrethike stin thesi: 2
Press any key to continue . . .
```

ΑΣΚΗΣΗ-2η : Παραλλαγή - Σειριακή ή γραμμική αναζήτηση αριθμού σε πίνακα ακεραίων

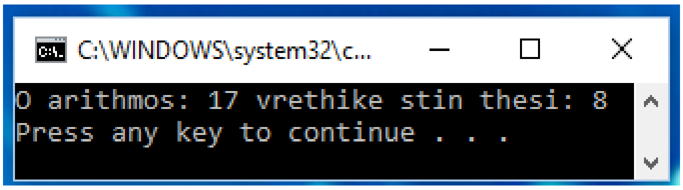
```
public class SequentialSearch1 {
    public static boolean linerSearch(int[] a, int b){
        for (int i : a) {
            if (i==b){
                return true;
            }
        }
        return false;
   }

   public static void main(String a[]){
      int[] arr1= {2,35,11,45,80,12,17,44};
      int searchKey = 17;
      boolean x=linerSearch(arr1, searchKey);
      if (x)
        System.out.println("O arithmos "+searchKey+" vrethike stin lista");
      else
        System.out.println("O arithmos "+searchKey+" den vrethike stin lista");
    } }
```



```
O arithmos 17 vrethike stin lista
Press any key to continue . . .
```

ΑΣΚΗΣΗ-3η Δυαδική αναζήτηση *(Binary Search)*

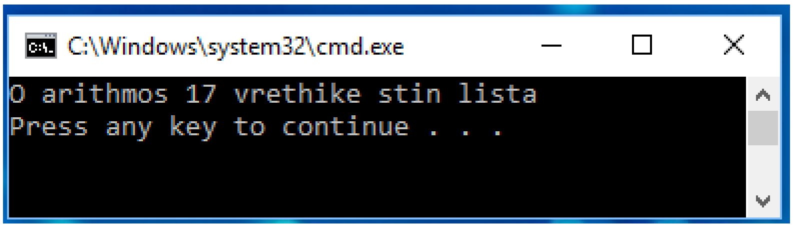
```
class MyBinarySearch1 {
 public static void main (String[] args) {
   int orderednumbers[] = {-31, -22, 1, 3, 4, 5, 9, 10, 17, 23 };
   int key=17;
   System.out.println("O arithmos: "+ key + " vrethike stin
             thesi: "+ MyBinarySearch1.binarysearch(orderednumbers, key) );}
public static int binarysearch(int[] A, int x) {
  int left = 0, right = A.length-1;
  int mid, found = -1;
  while (found == -1 && left <= right) { mid = (left + right) / 2;
   if (x < A[mid]) {  // to x sto 1o miso
     right = mid-1;}
   else if (x > A[mid]) { // to x sto 2o miso
   left = mid + 1;}
   else found = mid; }
   return found; } }
```



```
O arithmos: 17 vrethike stin thesi: 8
Press any key to continue . . .
```

ΑΣΚΗΣΗ-4η Παραλλαγή - Δυαδική αναζήτηση *(Binary Search)*

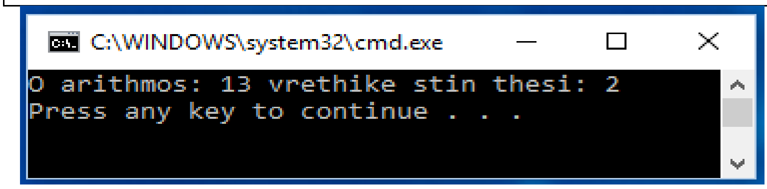
```
class MyBinarySearch1 {
 public static void main (String[] args) {
   int arr1[] = {-31, -22, 1, 3, 4, 5, 9, 10, 17, 23 };
   int key=17;
   boolean x=binarysearch1(arr1, key);
   if (x)
     System.out.println("O arithmos "+key+" vrethike stin lista");
   else
     System.out.println("O arithmos "+key+" den vrethike stin lista"); }

public static boolean binarysearch1(int[] a, int b) {
   if (a.length == 0) {
            return false; }
        int low = 0;
        int high = a.length-1;
        while(low <= high ) {
            int middle = (low+high) /2;
            if (b> a[middle] ){
                low = middle +1;
            } else if (b< a[middle]){
                high = middle -1;
            } else { // The element has been found
                return true;
            }
        }
        return false;
    } }
```



```
O arithmos 17 vrethike stin lista
Press any key to continue . . .
```

ΑΣΚΗΣΗ-5η Αναδρομική Δυαδική αναζήτηση *(Recursive Binary Search)*

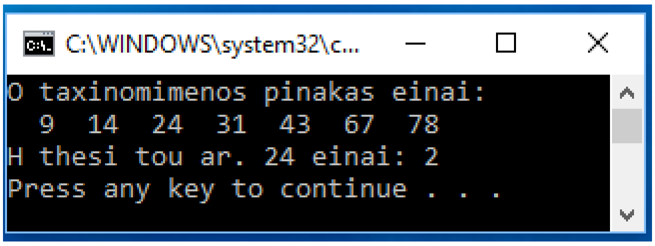
```
class RecursiveBinarySearch {
  public static void main (String[] args)   {
    int arr[] = {11, 12, 13, 14, 15, 16, 17, 18 ,19, 20};
    int key=13;
    int found = recBinarySearch(arr, key, 0, arr.length - 1);
    if (found > -1){
      System.out.println ("O arithmos: ‟ + key + " vrethike stin thesi: " + found);}
    else{System.out.println("O arithmos den vrethike"); } }

public static int recBinarySearch(int[] arr, int key, int left, int right) {
    int mid;
    if (right < left){return -1;}
    mid = (left + right) / 2;
    if (arr[mid] < key)
        return recBinarySearch(arr, key, mid + 1, right);
    else if (arr[mid] > key)
        return recBinarySearch(arr, key, left, mid - 1);
    else return mid; } }
```



```
O arithmos: 13 vrethike stin thesi: 2
Press any key to continue . . .
```

ΑΣΚΗΣΗ-6η (Δυαδική Αναζήτηση με την εντολή της Java - binarySearch())

```
import java.util.Arrays;
public class ArraysBinarySearch {
  public static void main(String[] args) {
  int Array[] = {14,31,67,43,24,9,78};

  Arrays.sort(Array);   // taxinomisi pinaka
  System.out.println("O taxinomimenos pinakas einai: ");
  for (int i : Array) {System.out.print("  "+i);}
  System.out.println();

  // O ar. anazitisis p.x. 24
  int key = 24;
  int thesi = Arrays.binarySearch(Array,key);
  System.out.println("H thesi tou ar. 24 einai: " + thesi);}}
```



```
O taxinomimenos pinakas einai:
  9  14  24  31  43  67  78
H thesi tou ar. 24 einai: 2
Press any key to continue . . .
```