

ΑΣΚΗΣΕΙΣ

Αλγόριθμοι Αναζήτησης (*Searching Algorithms*)

ΑΣΚΗΣΗ-1^η

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με σειριακή αναζήτηση** ένα αριθμό σε ένα αταξινόμητο πίνακα ακεραίων. Ο αλγόριθμος απαιτεί την διαδοχική σύγκριση του αριθμού που αναζητούμε με τα στοιχεία του πίνακα, ξεκινώντας από το πρώτο στοιχείο. Η αναζήτηση σταματά όταν βρεθεί το στοιχείο που αναζητάμε ή όταν φτάσουμε στο τέλος της λίστας. Να γράψετε μία **στατική μέθοδο**, η οποία θα δέχεται σαν παραμέτρους τον πίνακα των ακεραίων και τον αριθμό που ζητάμε και επιστρέφει τη θέση στην οποία βρέθηκε ο αριθμός που ζητάμε. Π.χ.

```
public static int linerSearch(int[] arr, int key)
```

Η εμφάνιση του αποτελέσματος θα γίνεται στην main().

ΑΣΚΗΣΗ-2^η (παραλλαγή της 1^{ης})

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με σειριακή αναζήτηση** ένα αριθμό σε ένα αταξινόμητο πίνακα ακεραίων. Ο αλγόριθμος απαιτεί την διαδοχική σύγκριση του αριθμού που αναζητούμε με τα στοιχεία του πίνακα, ξεκινώντας από το πρώτο στοιχείο. Η αναζήτηση σταματά όταν βρεθεί το στοιχείο που αναζητάμε ή όταν φτάσουμε στο τέλος της λίστας. Να γράψετε μία **στατική μέθοδο**, η οποία θα δέχεται σαν παραμέτρους τον πίνακα των ακεραίων και τον αριθμό που ζητάμε και επιστρέφει true ή false, ανάλογα αν βρέθηκε ο αριθμός ή όχι. Π.χ.

```
public static boolean linerSearch(int[] a, int b)
```

Η εμφάνιση του αποτελέσματος θα γίνεται στην main().

ΑΣΚΗΣΗ-3^η

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με δυναμική αναζήτηση** ένα αριθμό σε ένα **ταξινομημένο πίνακα** ακεραίων. Ο αλγόριθμος απαιτεί κάθε φορά που κάνουμε μια σύγκριση να μικραίνουμε την λίστα των τιμών στο μισό μέχρις ότου βρούμε το στοιχείο ή δεν το βρούμε στην λίστα των τιμών (δες θεωρία). Να γράψετε μία **στατική μέθοδο**, η οποία θα δέχεται σαν παραμέτρους τον πίνακα των ακεραίων και τον αριθμό που ζητάμε και επιστρέφει τη θέση στην οποία βρέθηκε ο αριθμός που ζητάμε. Π.χ.

```
public static int binarysearch(int[] A, int x)
```

Η εμφάνιση του αποτελέσματος θα γίνεται στην main().

ΑΣΚΗΣΗ-4^η (παραλλαγή της 3^{ης})

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με δυναμική αναζήτηση** ένα αριθμό σε ένα **ταξινομημένο πίνακα** ακεραίων. Ο αλγόριθμος απαιτεί κάθε φορά που κάνουμε μια σύγκριση να μικραίνουμε την λίστα των τιμών στο μισό μέχρις ότου βρούμε το στοιχείο ή δεν το βρούμε στην λίστα των τιμών (δες θεωρία). Να γράψετε μία **στατική μέθοδο**, η οποία θα δέχεται σαν παραμέτρους τον πίνακα των ακεραίων και τον αριθμό που ζητάμε και επιστρέφει true ή false, ανάλογα αν βρέθηκε ο αριθμός ή όχι. Π.χ.

```
public static boolean binarysearch(int[] a, int b)
```

Η εμφάνιση του αποτελέσματος θα γίνεται στην main().

ΑΣΚΗΣΗ-5^η (Αναδρομική Δυναμική Αναζήτηση - Recursive Binary Search)

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με αναδρομική δυναμική αναζήτηση** ένα αριθμό σε ένα **ταξινομημένο πίνακα** ακεραίων. Να γράψετε μία **στατική αναδρομική μέθοδο**, η οποία θα δέχεται σαν παραμέτρους τον πίνακα των ακεραίων, τον αριθμό που ζητάμε, και τα left και right. Η μέθοδος επιστρέφει τη θέση στην οποία βρέθηκε ο αριθμός που ζητάμε. Π.χ.

```
public static int recBinarySearch(int[] arr, int key, int left, int right)
```

Η εμφάνιση του αποτελέσματος θα γίνεται στην main().

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

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με δυναμική αναζήτηση** ένα αριθμό σε ένα **ταξινομημένο πίνακα** ακεραίων με την εντολή της Java:

```
java.util.Arrays.binarySearch(int[] a, int key)
```

Η μέθοδος **binarySearch()** δέχεται σαν παραμέτρους τον πίνακα των ακεραίων και τον αριθμό που ζητάμε και επιστρέφει τη θέση στην οποία βρέθηκε ο αριθμός που ζητάμε. Για την ταξινόμηση του πίνακα χρησιμοποιείτε την εντολή της Java: **Arrays.sort(Array)**.

ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

ΑΣΚΗΣΗ-1^η

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει με σειριακή αναζήτηση** ένα όνομα σε ένα αταξινομητό πίνακα ονομάτων. Ο αλγόριθμος είναι σχεδόν ίδιος με αυτόν των ασκήσεων 1 και 2 μόνο που αντί για ακεραίους θα έχουμε Strings. Η σύγκριση δύο String γίνεται με την μέθοδο **equals()**.

ΑΣΚΗΣΗ-2^η

Να γραφεί το πρόγραμμα Java που **βρίσκει και εμφανίζει** τα στοιχεία ενός υπαλλήλου σε ένα πίνακα υπαλλήλων. Θα γίνουν δύο αναζητήσεις: (1) ως προς το επώνυμο του υπαλλήλου, και (2)

ως προς το ΑΦΜ (*ταξινομημένος πίνακας ως προς το ΑΦΜ*). Το πρόγραμμα ορίζει τουλάχιστον τρεις κλάσεις:

(1) κλάση **Employee** με χαρακτηριστικά: Όνομα(String), Επώνυμο(String), ΑΦΜ(long) και Μισθός(double) και τον πλήρη δομητή, getters() και την μέθοδο toString().

(2) κλάση **MyUtils** που θα περιέχει δύο μεθόδους: (α) **AnazitisiMeEpwnymo()** που θα αναζητά στον πίνακα των υπαλλήλων ένα υπάλληλο ως προς το επώνυμο και θα επιστρέφει τη θέση στην οποία βρέθηκε ή (-1) αν δεν βρεθεί. Η μέθοδος καλείται με δύο παραμέτρους τον *πίνακα των αντικειμένων (υπαλλήλων)* και το *επώνυμο* που αναζητάμε. (β) **AnazitisiMeAFM()** που θα αναζητά στον πίνακα των υπαλλήλων ένα υπάλληλο ως προς το ΑΦΜ και θα επιστρέφει τη θέση στην οποία βρέθηκε ή (-1) αν δεν βρεθεί. Η μέθοδος καλείται με δύο παραμέτρους τον *πίνακα των αντικειμένων (υπαλλήλων)* και το *ΑΦΜ* που αναζητάμε.

(3) κλάση **TestEmployee**, που θα περιέχει την main(). Δημιουργήστε ένα πίνακα **N**-υπαλλήλων και δώστε ένα επώνυμο για σειριακή αναζήτηση και ένα ΑΦΜ για δυαδική αναζήτηση. Φροντίστε ο πίνακας των υπαλλήλων να είναι ταξινομημένος από την αρχή ως προς το ΑΦΜ ή αλλιώς θα πρέπει να τον ταξινομήσετε πριν την αναζήτηση. Τα πλήρη στοιχεία του προς αναζήτηση υπαλλήλου θα εμφανίζονται στην main().

ΑΣΚΗΣΗ-3^η (παραλλαγή της 2^{ης})

Στην προηγούμενη άσκηση προσθέστε στην κλάση **MyUtils** μια τρίτη μέθοδο την **AnadromikiAnazitisiMeAFM()** που θα υλοποιεί την δυαδική αναζήτηση με αναδρομή.