

# Μάθημα 154

Επανάληψη

B.2

Δομές δεδομένων

## Άσκηση 1 / Αναζήτηση όλων

Να γραφεί πρόγραμμα το οποίο θα διαβάζει και θα καταχωρεί σε πίνακα 100 θέσεων τα ονόματα των 100 εργαζομένων μιας επιχείρησης. Στη συνέχεια θα διαβάζει ένα όνομα και θα αναζητεί πόσες φορές και σε ποιες θέσεις βρίσκεται αυτό στον πίνακα και θα εμφανίζει τα κατάλληλα μηνύματα.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_1\_ΕΥΡΕΣΗ\_ΟΣΩΝ\_ΤΑΙΡΙΑΖΟΥΝ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, C

**ΧΑΡΑΚΤΗΡΕΣ:** ON[100], key

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

**ΔΙΑΒΑΣΕ** ON[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** key

C ← 0

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

**ΑΝ** ON[i] = key **ΤΟΤΕ**

**ΓΡΑΨΕ** i

C ← C + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** C

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 2 / Αναζήτηση μοναδικού

Να γραφεί πρόγραμμα το οποίο θα διαβάζει και θα καταχωρεί σε πίνακα 200 θέσεων τους αριθμούς αστυνομικής ταυτότητας 200 πολιτών. Στη συνέχεια θα διαβάζει έναν αριθμό ταυτότητας και αναζητεί και εμφανίζει αν ο αριθμός αυτός βρίσκεται στον πίνακα και αν ναι σε ποια θέση.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_2\_ΣΕΙΡΙΑΚΗ\_ΑΝΑΖΗΤΗΣΗ\_A  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, pos

**ΧΑΡΑΚΤΗΡΕΣ:** AT[200], key

**ΛΟΓΙΚΕΣ:** done

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 200

**ΔΙΑΒΑΣΕ** AT[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** key

i ← 1

pos ← 0

done ← **ΨΕΥΔΗΣ**

**ΟΣΟ** done = **ΨΕΥΔΗΣ** **ΚΑΙ** i ≤ 200 **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** AT[i] = key **ΤΟΤΕ**

pos ← i

done ← **ΑΛΗΘΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

i ← i + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** done = **ΑΛΗΘΗΣ** **ΤΟΤΕ**

**ΓΡΑΨΕ** pos

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν βρέθηκε'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_2\_ΣΕΙΡΙΑΚΗ\_ΑΝΑΖΗΤΗΣΗ\_B  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΧΑΡΑΚΤΗΡΕΣ:** AT[200], key

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 200

**ΔΙΑΒΑΣΕ** AT[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** key

i ← 1

**ΟΣΟ** AT[i] <> key **ΚΑΙ** i < 200 **ΕΠΑΝΑΛΑΒΕ**

i ← i + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** AT[i] = key **ΤΟΤΕ**

**ΓΡΑΨΕ** i

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν βρέθηκε'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

### Άσκηση 3 / Αναζήτηση σε ταξινομημένο πίνακα του τελευταίου που ταιριάζει

Να γραφεί πρόγραμμα το οποίο θα διαβάζει και θα καταχωρεί ακέραιους στον πίνακα A[150], τον οποίο στη συνέχεια και ταξινομεί σε φθίνουσα σειρά. Μετά θα διαβάζει έναν αριθμό τον οποίο στη συνέχεια θα αναζητά στον ταξινομημένο πίνακα. Ο αριθμός αυτός μπορεί να υπάρχει περισσότερες από μία φορές. Η αναζήτηση πρέπει να σταματά αμέσως μόλις βρεθεί στοιχείο μικρότερο από αυτό που αναζητείται. Στο τέλος θα εμφανίζεται αντίστοιχο μήνυμα που αφορά στην εύρεση ή μη του στοιχείου.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_3\_ΣΕΙΡΙΑΚΗ\_ΣΕ\_ΤΑΞΙΝΟΜΗΜΕΝΟ  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, j, A[150], pos, key, temp

**ΛΟΓΙΚΕΣ:** done

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 150

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*! Ταξινόμηση*

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 150

**ΓΙΑ** j **ΑΠΟ** 150 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** A[j - 1] < A[j] **ΤΟΤΕ**

temp ← A[j - 1]

A[j - 1] ← A[j]

A[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** key

*! Αναζήτηση*

i ← 1

pos ← 0

done ← **ΨΕΥΔΗΣ**

**ΟΣΟ** done = **ΨΕΥΔΗΣ** **ΚΑΙ** i <= 150 **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** A[i] = key **ΤΟΤΕ**

pos ← i

**ΑΛΛΙΩΣ\_ΑΝ** A[i] < key **ΤΟΤΕ**

done ← **ΑΛΗΘΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

i ← i + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** pos <> 0 **ΤΟΤΕ**

**ΓΡΑΨΕ** 'Βρέθηκε τελευταία φορά στη θέση ', pos

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν βρέθηκε'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 4 / Δυαδική αναζήτηση

Να γραφεί πρόγραμμα το οποίο θα διαβάζει και θα καταχωρεί στους πίνακες ΑΦΜ[1000] και ΟΝ[1000] τον Αριθμό Φορολογικού Μητρώου και το όνομα 1000 πολιτών και θα τους ταξινομεί με βάση το ΑΦΜ από το παλιότερο προς το νεότερο (αύξουσα ταξινόμηση). Να σημειωθεί πως τα παλαιότερα ΑΦΜ ξεκινούσαν με μηδέν. Στη συνέχεια θα διαβάζει ένα ΑΦΜ και θα εμφανίζει το όνομα του κατόχου του ή κατάλληλο μήνυμα εάν δεν είναι καταχωρημένο.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_4\_ΔΥΑΔΙΚΗ\_ΑΝΑΖΗΤΗΣΗ  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, j, L, M, R

**ΧΑΡΑΚΤΗΡΕΣ:** ΑΦΜ[1000], ΟΝ[1000], temp, key

**ΛΟΓΙΚΕΣ:** done

**ΑΡΧΗ**

*!καταχώρηση στοιχείων*

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 1000

**ΔΙΑΒΑΣΕ** ΑΦΜ[i], ΟΝ[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!ταξινόμηση*

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 1000

**ΓΙΑ** j **ΑΠΟ** 1000 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** ΑΦΜ[j - 1] > ΑΦΜ[j] **ΤΟΤΕ**

temp ← ΑΦΜ[j - 1]

ΑΦΜ[j - 1] ← ΑΦΜ[j]

ΑΦΜ[j] ← temp

temp ← ΟΝ[j - 1]

ΟΝ[j - 1] ← ΟΝ[j]

ΟΝ[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!στοιχείο προς αναζήτηση*

**ΔΙΑΒΑΣΕ** key

*!δυαδική αναζήτηση*

L ← 1

R ← 1000

done ← **ΨΕΥΔΗΣ**

**ΟΣΟ** L ≤ R **ΚΑΙ** done = **ΨΕΥΔΗΣ** **ΕΠΑΝΑΛΑΒΕ**

M ← (L + R) div 2

**ΑΝ** ΑΦΜ[M] = key **ΤΟΤΕ**

done ← **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ\_ΑΝ** ΑΦΜ[M] < key **ΤΟΤΕ**

L ← M + 1

**ΑΛΛΙΩΣ**

R ← M - 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!αποτελέσματα αναζήτησης*

**ΑΝ** done = **ΑΛΗΘΗΣ** **ΤΟΤΕ**

**ΓΡΑΨΕ** ΟΝ[M]

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν βρέθηκε'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 5 / Συγχώνευση πινάκων

Να γράψετε πρόγραμμα που θα διαβάσει τα στοιχεία δύο πινάκων A[350] και B[150] με ακέραιους, ταξινομεί τον A σε φθίνουσα σειρά και τον B σε αύξουσα σειρά, στη συνέχεια τους συγχωνεύει στον πίνακα Γ[500] σε φθίνουσα σειρά και τελικά εμφανίζει τα στοιχεία του πίνακα Γ.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_154\_5\_ΤΑΞΙΝΟΜΗΣΗ\_ΣΥΓΧΩΝΕΥΣΗ  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, j, δ, δ1, δ2

**ΑΚΕΡΑΙΕΣ:** A[350], B[150], Γ[500], temp

**ΑΡΧΗ**

*!καταχώρηση τιμών*

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 350

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 150

**ΔΙΑΒΑΣΕ** B[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

*!ταξινόμηση πινάκων*

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 350

**ΓΙΑ** j **ΑΠΟ** 350 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** A[j - 1] < A[j] **ΤΟΤΕ**

temp ← A[j - 1]

A[j - 1] ← A[j]

A[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 150

**ΓΙΑ** j **ΑΠΟ** 150 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** B[j - 1] > B[j] **ΤΟΤΕ**

temp ← B[j - 1]

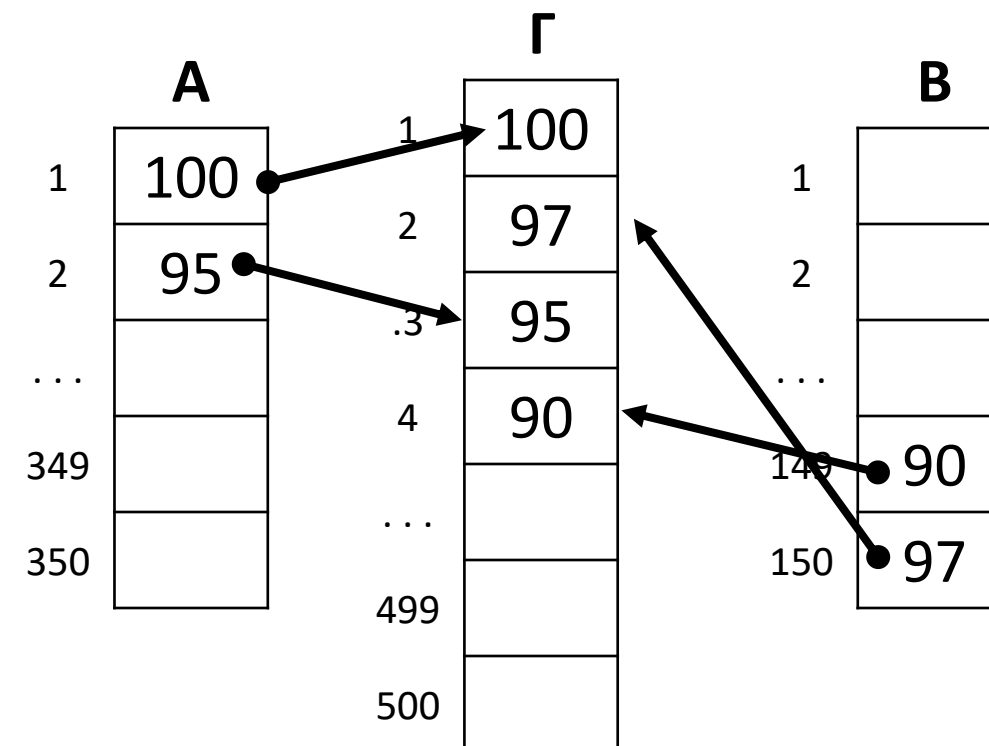
B[j - 1] ← B[j]

B[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**



*!συγχώνευση πινάκων*

δ1 ← 1

δ2 ← 150

δ ← 1

**ΟΣΟ** δ1 ≤ 350 **ΚΑΙ** δ2 ≥ 1 **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** A[δ1] > B[δ2] **ΤΟΤΕ**

Γ[δ] ← A[δ1]

δ1 ← δ1 + 1

**ΑΛΛΙΩΣ**

Γ[δ] ← B[δ2]

δ2 ← δ2 - 1

**ΤΕΛΟΣ\_ΑΝ**

δ ← δ + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** δ1 ≤ 350 **ΤΟΤΕ**

**ΓΙΑ** i **ΑΠΟ** δ1 **ΜΕΧΡΙ** 350

Γ[δ] ← A[i]

δ ← δ + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΛΛΙΩΣ**

**ΓΙΑ** i **ΑΠΟ** δ2 **ΜΕΧΡΙ** 1 **ΜΕ\_ΒΗΜΑ** - 1

Γ[δ] ← B[i]

δ ← δ + 1

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

*!εκτύπωση τελικού πίνακα*

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 500

**ΓΡΑΨΕ** Γ[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Να γραφεί αλγόριθμος που διαβάζει και καταχωρεί 100 τιμές (έστω διαφορετικές μεταξύ τους) στον πίνακα A, στη συνέχεια διαβάζει μία τιμή στη μεταβλητή key την οποία αναζητά στον πίνακα A (με χρήση pos και χωρίς χρήση done) και τελικά εμφανίζει τη θέση του πίνακα στην οποία βρέθηκε το key. Αν δεν βρεθεί εμφανίζει 0. Ακολουθώς να σχεδιαστεί το διάγραμμα ροής του αλγόριθμου αυτού.
2. Να γράψετε πρόγραμμα που θα διαβάζει τα στοιχεία δύο πινάκων A[350] και B[150] με ονόματα και τελικά θα δημιουργεί έναν πίνακα Γ[500] με όλα τα ονόματα ταξινομημένα αλφαβητικά, τα στοιχεία του οποίου και θα εμφανίζει.  
Η υλοποίηση να γίνει με δύο τρόπους και συγκεκριμένα:
  - α. Πρώτα ενώνει τους δύο πίνακες και μετά ταξινομεί τον πίνακα Γ.
  - β. Πρώτα ταξινομεί τους πίνακες A και B και στην πορεία τους συγχωνεύει.