

## I. ΑΛΓΟΡΙΘΜΟΣ

1. Τα πιο συνηθισμένα σενάρια παραβίασης αλγοριθμικών κριτηρίων είναι:
  - i. **Είσοδος** : χρήση μιας μεταβλητής που δεν έχει πάρει προηγουμένως τιμή.
  - ii. **Έξοδος** : ο αλγόριθμος δεν εμφανίζει κανένα αποτέλεσμα
  - iii. **Περατότητα** : κάποια επαναληπτική δομή δεν τερματίζεται ποτέ (ατέρμων βρόχος).
  - iv. **Καθοριστικότητα**: σε κάποια εντολή πραγματοποιείται διαίρεση χωρίς να έχει διασφαλιστεί ότι ο παρονομαστής είναι διάφορος του μηδέν ή επιχειρείται ο υπολογισμός τετραγωνικής ρίζας χωρίς να έχει διασφαλιστεί ότι το υπόριζο είναι μεγαλύτερο ή ίση του μηδενός.
  - v. **Αποτελεσματικότητα**: κάποια εντολή δίδεται με περιγραφικό τρόπο και μπορεί να αναλυθεί σε δύο ή περισσότερα απλούστερα βήματα.
2. **Εκτελεστές εντολές**: ΔΙΑΒΑΣΕ, ΓΡΑΨΕ κ.λ.π.
3. **Δηλωτικές εντολές**: ΠΡΟΓΡΑΜΜΑ, ΑΛΓΟΡΙΘΜΟΣ, ΤΕΛΟΣ\_ΑΝ κ.λ.π.

## II. ΠΡΑΞΕΙΣ - ΣΥΝΑΡΤΗΣΕΙΣ

1. Η απόδοση αρχικών τιμών στις μεταβλητές βοηθάει στην καλύτερη κατανόηση του προγράμματος και στην ευκολότερη συντήρησή του.
2. Οι συναρτήσεις  $HM()$ ,  $ΣΥΝ()$  και  $ΕΦ()$  δέχονται παράμετρο σε **μοίρες**.
3. Αν  $a < \beta$  τότε  $a \bmod \beta = a$ .
4. Οι πράξεις:  $a \bmod 0$ ,  $a \div 0$  και  $a/0$  ανάγουν στην διαίρεση δια μηδέν (καθοριστικότητα)
5. Στη συνάρτηση  $ΛΟΓ(X)$  το  $X$  είναι θετικό άρα  $ΛΟΓ(X)$  με  $X \leq 0$  δεν ορίζεται (καθοριστικότητα).
6. Ισχύει ότι  $A\_M(x) \leq x < A\_M(x) + 1$ , με  $x$  οποιονδήποτε αριθμό (ακέραιο ή πραγματικό)
7. **ΑΝ  $A\_M(x)=x$  ΤΟΤΕ** |"ο αριθμός  $x$  είναι ακέραιος"
8. Η απόλυτη τιμή  $A\_T()$  μπορεί να πάρει ως παράμετρο, είτε ακέραιο αριθμό και να επιστρέψει ακέραιο, είτε πραγματικό αριθμό και να επιστρέψει πραγματικό
9. Η ιεραρχία των λογικών πράξεων είναι: 1. όχι , 2. και 3. ή.
10. Αντί για **ΓΡΑΨΕ  $2*A-3$**  καλύτερα είναι να γράφουμε  **$X \leftarrow 2*A-3$**  και **ΓΡΑΨΕ  $X$** .

## III. ΕΠΑΝΑΛΗΨΕΙΣ

1. Οι μεταβλητές που ελέγχουν την επανάληψη του βρόχου **ΟΣΟ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ** πρέπει υποχρεωτικά να αλλάζουν τιμή μέσα στο σώμα του βρόχου, αλλιώς η δεν εκτελείται ποτέ ή συνηθέστερα δεν σταματάει η εκτέλεση του (ατέρμων βρόχος).
2. Ποτέ μη χρησιμοποιείς εντολές που αλλάζουν την αρχική τιμή, την τελική τιμή, το βήμα ή τη μεταβλητή που ελέγχει την επανάληψη μέσα σε ένα βρόχο **ΓΙΑ**. Αυτό μπορεί να οδηγήσει σε προγράμματα δυσνόητα και συνήθως λαναθασμένα.
3. Στην εντολή επανάληψης: **Για** <μεταβλητή> **ΑΠΟ** < $t_1$ > **ΜΕΧΡΙ** < $t_2$ > **ΜΕ\_ΒΗΜΑ** < $t_3$ >
  - i. αν  $t_3=0$ , τότε ο βρόχος εκτελείται άπειρες φορές (ατέρμονας βρόχος).
  - ii. Αν  $t_1 > t_2$  και  $\beta > 0$  ή  $t_1 < t_2$  και  $\beta < 0$  τότε ο βρόχος δεν εκτελείται καμία φορά.
  - iii. Αν  $t_1 = t_2$  και  $\beta \neq 0$  τότε ο βρόχος θα εκτελεστεί ακριβώς μία φορά.

4. Ενδείκνυται η **ΟΣΟ** όταν:

i. η επανάληψη καθορίζεται από την τιμή της μεταβλητής που διαβάζεται

Παραδείγματα.

a. Να διαβάσει αριθμούς από το πληκτρολόγιο και να υπολογίζει ..... Η διαδικασία να ολοκληρώνεται αν δώσουμε είσοδο 0. **Το 0 λέγεται τιμή φρουρός.**

b. Να διαβάσει ακέραιους από το πληκτρολόγιο και να υπολογίζει ..... Η διαδικασία να ολοκληρώνεται αν δώσουμε είσοδο μη θετικό.

5. Ενδείκνυται η **ΜΕΧΡΙΣ\_ΟΤΟΥ** όταν:

i. η επανάληψη καθορίζεται από απάντηση μετά από κατάλληλη ερώτηση τότε.

Παράδειγμα.

a. Να διαβάσει ..... από το πληκτρολόγιο και να υπολογίζει ..... Η διαδικασία να ολοκληρώνεται αν απαντήσουμε ΟΧΙ στην ερώτηση «Θα συνεχίσεις;».

ii. Απαιτείται έλεγχος εισόδου

Παράδειγμα.

a. Να διαβάσει έναν ακέραιο από το πληκτρολόγιο ελέγχοντας ώστε να είναι θετικός, άρτιος και διαφορετικός του 10.

iii. Μενού επιλογών

Παράδειγμα.

a. Να εμφανίζει το παρακάτω μενού και να ζητάει την επιλογή του χρήστη. Ανάλογα με την επιλογή να γίνεται ..... Η διαδικασία ολοκληρώνεται όταν ο χρήστης πληκτρολογήσει το 5.

1. Πρόσθεση

2. Αφαίρεση

3. Πολλαπλασιασμός

4. Διάρθρωση

5. Τέλος

Δώσε την επιλογή σου.

## 6. Μετατροπή της ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ σε ΟΣΟ

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

<εντολές>

ΜΕΧΡΙΣ\_ΟΤΟΥ <συνθήκη>



<εντολές>

ΟΣΟ Όχι <συνθήκη> ΕΠΑΝΑΛΑΒΕ

<εντολές>

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

## 7. Μετατροπή της ΟΣΟ σε ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΟΣΟ <συνθήκη> ΕΠΑΝΑΛΑΒΕ

<εντολές>

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ



ΑΝ <συνθήκη> ΤΟΤΕ

ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

<εντολές>

ΜΕΧΡΙΣ\_ΟΤΟΥ όχι <συνθήκη>

ΤΕΛΟΣ\_ΑΝ

ή



ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ <συνθήκη> ΤΟΤΕ

<εντολές>

ΤΕΛΟΣ\_ΑΝ

ΜΕΧΡΙΣ\_ΟΤΟΥ όχι <συνθήκη>

## 8. Μετατροπή από ΓΙΑ... σε ΟΣΟ... ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Περίπτωση τιμή1 <= τιμή2 και β > 0	Περίπτωση τιμή1 >= τιμή2 και β < 0
ΓΙΑ <μεταβλητή> ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ_ΒΗΜΑ β <εντολές> ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	



<μεταβλητή> ← τιμή1 ΟΣΟ <μεταβλητή> <=τιμή2 ΕΠΑΝΑΛΑΒΕ <εντολές> <μεταβλητή> ← <μεταβλητή> + β ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	<μεταβλητή> ← τιμή1 ΟΣΟ <μεταβλητή> >= τιμή2 ΕΠΑΝΑΛΑΒΕ <εντολές> <μεταβλητή> ← <μεταβλητή> + β ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
--	---

## IV. ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

1. Στατικές δομές:
  - i. Αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
  - ii. έχουν σταθερό μέγεθος, το οποίο καθορίζεται κατά τη στιγμή του προγραμματισμού τους (στην αρχή του προγράμματος).
  - iii. Οι στατικές δομές υλοποιούνται με πίνακες.
2. Δυναμικές δομές:
  - i. Δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
  - ii. δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους αυξάνεται και μειώνεται, όταν στη δομή αντίστοιχα εισάγονται ή διαγράφονται δεδομένα.
  - iii. Το μέγεθος της μνήμης καθορίζεται κατά την στιγμή της εκτέλεσης του προγράμματος.
3. Μια δομή δεδομένων δεν είναι εγγενώς στατική ή δυναμική, αλλά εξαρτάται από
  - i. τις δυνατότητες της γλώσσας προγραμματισμού που χρησιμοποιούμε και
  - ii. τον τρόπο υλοποίησης της δομής στη γλώσσα αυτή.
4. Οποιαδήποτε γλώσσα προγραμματισμού δεν υποστηρίζει όλες τις δομές δεδομένων, με τις σύγχρονες γλώσσες προγραμματισμού να υποστηρίζουν δυναμικές δομές δεδομένων.
5. Η γλώσσα προγραμματισμού ΓΛΩΣΣΑ, που χρησιμοποιείται στο βιβλίο, υποστηρίζει μόνο στατικές δομές.

## V. ΠΙΝΑΚΕΣ

1. Δύο η περισσότεροι πίνακες λέγονται **παράλληλοι**, αν σε αυτούς έχουμε αποθηκεύσει τα χαρακτηριστικά οντοτήτων με τέτοιο τρόπο ώστε τα δεδομένα κάθε οντότητας να βρίσκονται σε στοιχεία με την ίδια τιμή δείκτη.
2. Δεν μπορούμε να χρησιμοποιήσουμε πίνακα για την επεξεργασία κάποιων τιμών εισόδου όταν δεν γνωρίζουμε ούτε το ακριβές πλήθος ούτε το ανώτερο δυνατό πλήθος των τιμών εισόδου.
3. Στην περίπτωση που **πρέπει** να χρησιμοποιήσω πίνακα αλλά δεν ξέρω το ακριβές μέγεθος του, ορίζω τον πίνακα με αριθμό κελιών το μέγιστο δυνατό.

4. Οι πράξεις (επί των δομών δεδομένων) εισαγωγή και διαγραφή κόμβων δεν υλοποιούνται στους πίνακες.
5. Στον πίνακα, κατά την πράξη της ταξινόμησης, δεν αναδιατάσσονται οι κόμβοι του αλλά το περιεχόμενο των κόμβων.
6. Η λογικής μεταβλητής done (ή εύρηκα ή OK) στον αλγόριθμο της σειριακής αναζήτησης λέγεται «σημαία» και χρησιμοποιείται προκειμένου να αποφευχθούν περιττές επαναλήψεις.
7. Ο αλγόριθμος της δυαδικής αναζήτησης (binary search) εφαρμόζεται μόνο σε πίνακες που έχουν ταξινομημένα στοιχεία. Αν τα στοιχεία δεν είναι ταξινομημένα τότε δεν μπορεί να εφαρμοστεί.
8. Ένας πίνακας λέγεται αραιός αν ένα μεγάλο ποσοστό των στοιχείων (πχ πάνω από 80%) έχουν μηδενική τιμή.

## VI. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

1. Οι διαφορές μεταγλωττιστή και διερμηνευτή είναι:
  - i. Ο μεταγλωττιστής μεταφράζει όλες τις εντολές του **πηγαίου** προγράμματος σε γλώσσα μηχανής δημιουργώντας το **αντικείμενο** πρόγραμμα, το οποίο μετά την παρέμβαση του **συνδότη-φορτωτή** γίνεται **εκτελέσιμο** και εκτελείται. Αντίθετα ο διερμηνευτής μεταφράζει και εκτελεί επί τόπου μία εντολή κάθε φορά.
  - ii. Ο διερμηνευτής παρέχει τη δυνατότητα πιο γρήγορου ελέγχου και διόρθωσης των εντολών σε σχέση με το μεταγλωττιστή.
  - iii. Ο μεταγλωττιστής εξασφαλίζει ταχύτερη εκτέλεση του προγράμματος από το διερμηνευτή.

## VII. ΔΙΑΔΙΚΑΣΙΕΣ - ΣΥΝΑΡΤΗΣΕΙΣ

1. Οι συναρτήσεις δεν μπορούν να έχουν εντολές εισόδου (ΔΙΑΒΑΣΕ)- εξόδου (ΓΡΑΨΕ).
2. ΠΡΟΣΟΧΗ:  
Στην διαδικασία: αλλαγές των τιμών των **τυπικών** παραμέτρων αλλάζουν και τις τιμές των **πραγματικών** παραμέτρων (μετά το τέλος της διαδικασίας).  
Στην Συνάρτηση αυτό δεν γίνεται. Δηλαδή αν αλλάξει η τιμή μιας τυπικής παραμέτρου (μέσα στο σώμα της συνάρτησης), αυτό δεν θα επηρεάσει την τιμή της πραγματικής παραμέτρου.
3. Μια συνάρτηση δεν μπορεί να καλέσει διαδικασία αλλά μόνο συνάρτηση.
4. Μια διαδικασία μπορεί να καλέσει μια άλλη διαδικασία ή και συνάρτηση.

## VIII. ΑΛΓΟΡΙΘΜΟΙ - ΠΡΟΓΡΑΜΜΑΤΑ

1. Έξυπνη φουσαλίδα (δηλαδή σταματάει η διαδικασία ταξινόμησης φουσαλίδας αν ο πίνακας ταξινομηθεί σε ενδιάμεσο στάδια)

Αλγόριθμος	Πρόγραμμα
<p><b>Αλγόριθμος</b> Εξυπνη_Φουσαλίδα</p> <p><b>Δεδομένα</b> // table, n //</p> <p><math>i \leftarrow 2</math></p> <p>οκ ← αληθής</p> <p><b>οσο</b> <math>i \leq n</math> και οκ <b>επανάλαβε</b></p> <p style="padding-left: 20px;">οκ ← ψευδής</p> <p style="padding-left: 20px;"><b>Για</b> j από n μέχρι i με_βήμα -1</p> <p style="padding-left: 40px;"><b>Αν</b> table[j-1] &gt; table[j] <b>Τότε</b></p> <p style="padding-left: 60px;">βοηθ ← table[j-1]</p> <p style="padding-left: 60px;">table[j-1] ← table[j]</p> <p style="padding-left: 60px;">table[j] ← βοηθ</p> <p style="padding-left: 40px;">οκ ← αληθής</p> <p style="padding-left: 20px;">τέλος_αν</p> <p style="padding-left: 20px;"><b>Τέλος_Επανάληψης</b></p> <p><b>Τέλος_επανάληψης</b></p> <p><b>Αποτελέσματα</b> //table//</p> <p><b>Τέλος</b> Εξυπνη_Φουσαλίδα</p>	<p><b>ΠΡΟΓΡΑΜΜΑ</b> Εξυπνη_Φουσαλίδα</p> <p><b>ΜΕΤΑΒΛΗΤΕΣ</b></p> <p style="padding-left: 20px;"><b>ΑΚΕΡΑΙΕΣ:</b> A[20], βοηθ, i, j</p> <p style="padding-left: 20px;"><b>ΛΟΓΙΚΕΣ:</b> οκ</p> <p><b>ΑΡΧΗ</b></p> <p style="padding-left: 20px;"><b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> 20</p> <p style="padding-left: 40px;"><b>ΔΙΑΒΑΣΕ</b> A[i]</p> <p style="padding-left: 20px;"><b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b></p> <p><math>i \leftarrow 2</math></p> <p>οκ ← ΑΛΗΘΗΣ</p> <p><b>ΟΣΟ</b> <math>i \leq 20</math> <b>ΚΑΙ</b> οκ <b>ΕΠΑΝΑΛΑΒΕ</b></p> <p style="padding-left: 20px;">οκ ← ΨΕΥΔΗΣ</p> <p style="padding-left: 20px;"><b>ΓΙΑ</b> j <b>ΑΠΟ</b> 20 <b>ΜΕΧΡΙ</b> i <b>ΜΕ ΒΗΜΑ</b> -1</p> <p style="padding-left: 40px;"><b>ΑΝ</b> table[j-1] &gt; A[j] <b>ΤΟΤΕ</b></p> <p style="padding-left: 60px;">βοηθ ← table[j-1]</p> <p style="padding-left: 60px;">table[j-1] ← table[j]</p> <p style="padding-left: 60px;">table[j] ← βοηθ</p> <p style="padding-left: 40px;">οκ ← αληθής</p> <p style="padding-left: 20px;"><b>ΤΕΛΟΣ_ΑΝ</b></p> <p style="padding-left: 20px;"><b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b></p> <p><b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b></p> <p><b>ΓΡΑΨΕ</b> 'Εκτύπωση με ταξινομημένα τα στοιχεία'</p> <p style="padding-left: 20px;"><b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> 20</p> <p style="padding-left: 40px;"><b>ΓΡΑΨΕ</b> A[i]</p> <p style="padding-left: 20px;"><b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b></p> <p><b>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</b></p>

## 2. Ο Αλγόριθμος και το πρόγραμμα ταξινόμησης με επιλογή.

Αλγόριθμος	Πρόγραμμα
<b>Αλγόριθμος Selection_Sort</b> <b>Δεδομένα // table, n //</b> <b>Για i από 1 μέχρι n-1</b> $\theta \leftarrow i$ <b>Για j από i+1 μέχρι n</b> <b>Αν table[<math>\theta</math>] &gt; table[j] Τότε</b> $\theta \leftarrow j$ <b>τέλος_αν</b> <b>Τέλος_Επανάληψης</b> $\beta\omicron\eta\theta \leftarrow \text{table}[\theta]$ $\text{table}[\theta] \leftarrow \text{table}[i]$ $\text{table}[i] \leftarrow \beta\omicron\eta\theta$ <b>Τέλος_επανάληψης</b> <b>Αποτελέσματα //table//</b> <b>Τέλος selection_Sort</b>	<b>ΠΡΟΓΡΑΜΜΑ Selection_Sort</b> <b>ΜΕΤΑΒΛΗΤΕΣ</b> <b>ΑΚΕΡΑΙΕΣ: A[20], <math>\theta</math>, <math>\beta\omicron\eta\theta</math>, i, j</b> <b>ΑΡΧΗ</b> <b>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20</b> <b>ΔΙΑΒΑΣΕ A[i]</b> <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b> <b>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 19</b> $\theta \leftarrow i$ <b>ΓΙΑ j ΑΠΟ i + 1 ΜΕΧΡΙ 20</b> <b>ΑΝ table[<math>\theta</math>] &gt; A[j] ΤΟΤΕ</b> $\theta \leftarrow j$ <b>ΤΕΛΟΣ_ΑΝ</b> <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b> $\beta\omicron\eta\theta \leftarrow A[\theta]$ $A[\theta] \leftarrow A[i]$ $A[i] \leftarrow \beta\omicron\eta\theta$ <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b> <b>ΓΡΑΨΕ 'Εκτύπωση με ταξινομημένα τα στοιχεία'</b> <b>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20</b> <b>ΓΡΑΨΕ A[i]</b> <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b> <b>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</b>

## 3. Ο Αλγόριθμος και το πρόγραμμα δυαδικής αναζήτησης.

Αλγόριθμος	Πρόγραμμα
<p><b>αλγόριθμος</b> Δυαδική_αναζήτηση  !A μονοδιάστατος πίνακας N θέσεων,  !S το στοιχείο που αναζητούμε  <b>δεδομένα</b> // N, A, S //</p> <p>Left ← 1 ! αριστερό όριο  Right ← N ! δεξιό όριο  K ← 0 ! θέση του στοιχείου  F ← FALSE</p> <p><b>όσο</b> (Left&lt;=Right) και (f=FALSE) <b>επανάλαβε</b>  Mid ← (Left+Right) div 2  <b>αν</b> A[Mid]=S <b>τότε</b>  K ← M;  F ← TRUE;</p> <p><b>αλλιώς</b>  <b>αν</b> A[M]&lt;S <b>τότε</b>  Left ← Mid+1;</p> <p><b>αλλιώς</b>  Right← Mid-1;</p> <p><b>Τέλος_αν</b>  <b>Τέλος_αν</b>  <b>Τέλος_επανάληψης</b>  <b>Αν</b> F = TRUE <b>τότε</b>  <b>Εμφάνισε</b> "Το", S , "υπάρχει στη θέση:", M</p> <p><b>Αλλιώς</b>  <b>Εμφάνισε</b> "Το", S , " δεν υπάρχει "</p> <p><b>Τέλος_αν</b>  <b>Τέλος</b> Δυαδική_Αναζήτηση</p>	<p><b>ΠΡΟΓΡΑΜΜΑ</b> δυαδική_αναζήτηση  <b>ΜΕΤΑΒΛΗΤΕΣ</b>  <b>ΑΚΕΡΑΙΕΣ:</b> A[20], Left, Right, Mid, k, S, i  <b>ΛΟΓΙΚΕΣ:</b> f</p> <p><b>ΑΡΧΗ</b>  <b>ΓΡΑΨΕ</b> 'Οι αριθμοί που θα δοθούν πρέπει να είναι &amp; ταξινομημένοι κατά αύξουσα τάξη'  <b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> 20  <b>ΓΡΑΨΕ</b> 'Δώσε το', i, ' στοιχείο του πίνακα'  <b>ΔΙΑΒΑΣΕ</b> A[i]  <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>  <b>ΓΡΑΨΕ</b> 'Δώσε τιμή για αναζήτηση:'  <b>ΔΙΑΒΑΣΕ</b> S  Left ← 1  Right ← 20  k ← 0  f ← ΨΕΥΔΗΣ</p> <p><b>ΟΣΟ</b> (Left &lt;= Right) <b>ΚΑΙ</b> (f = ΨΕΥΔΗΣ) <b>ΕΠΑΝΑΛΑΒΕ</b>  Mid ← (Left + Right) DIV 2  <b>ΑΝ</b> A[M] = S <b>ΤΟΤΕ</b>  k ← Mid  f ← <b>ΑΛΗΘΗΣ</b></p> <p><b>ΑΛΛΙΩΣ</b>  <b>ΑΝ</b> A[Mid] &lt; S <b>ΤΟΤΕ</b>  Left ←- Mid + 1</p> <p><b>ΑΛΛΙΩΣ</b>  Right ← Mid - 1</p> <p><b>ΤΕΛΟΣ_ΑΝ</b>  <b>ΤΕΛΟΣ_ΑΝ</b>  <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>  <b>ΑΝ</b> f = <b>ΑΛΗΘΗΣ</b> <b>ΤΟΤΕ</b>  <b>ΓΡΑΨΕ</b> "Το στοιχείο,", S, "υπάρχει στη θέση:", M</p> <p><b>ΑΛΛΙΩΣ</b>  <b>ΓΡΑΨΕ</b> "Το στοιχείο,", S, " δεν υπάρχει"</p> <p><b>ΤΕΛΟΣ_ΑΝ</b>  <b>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</b></p>

## 4. Ο Αλγόριθμος και το πρόγραμμα συγχώνευσης 2 ταξινομημένων πινάκων.

Αλγόριθμος	Πρόγραμμα
<p><b>αλγόριθμος</b> Συγχώνευση</p> <p>!Α μονοδιάστατος πίνακας Ν θέσεων, !Β μονοδιάστατος πίνακας Μ θέσεων, !Γ μονοδιάστατος πίνακας Μ+Ν θέσεων</p> <p><b>δεδομένα</b> // Α,Β, Μ, Ν //</p> <p><math>i \leftarrow 1</math> <math>j \leftarrow 1</math> <math>k \leftarrow 1</math></p> <p><b>όσο</b> (<math>i \leq N</math>) και (<math>j \leq M</math>) <b>επανάλαβε</b>   <b>αν</b> <math>A[i] &lt; B[j]</math> <b>τότε</b>     <math>\Gamma[k] \leftarrow A[i]</math>     <math>k \leftarrow k+1</math>     <math>i \leftarrow i+1</math>   <b>αλλιώς</b>     <math>\Gamma[k] \leftarrow B[j]</math>     <math>k \leftarrow k+1</math>     <math>j \leftarrow j+1</math>   <b>Τέλος_αν</b></p> <p><b>Τέλος_επανάληψης</b></p> <p><b>Αν</b> <math>i &gt; N</math> <b>τότε</b>   <b>Για</b> λ από k μέχρι Μ+Ν     <math>\Gamma[\lambda] \leftarrow B[j]</math>     <math>j \leftarrow j+1</math>   <b>τέλος_επανάληψης</b></p> <p><b>αλλιώς</b>   <b>Για</b> λ από k μέχρι Μ+Ν     <math>\Gamma[\lambda] \leftarrow A[i]</math>     <math>i \leftarrow i+1</math>   <b>τέλος_επανάληψης</b></p> <p><b>Τέλος_αν</b></p> <p><b>Τέλος</b> Συγχώνευση</p>	<p><b>ΠΡΟΓΡΑΜΜΑ</b> Συγχώνευση</p> <p><b>ΜΕΤΑΒΛΗΤΕΣ</b>   <b>ΑΚΕΡΑΙΕΣ:</b> Α[100], Β[100], Γ[200], k, j, i, λ   <b>ΛΟΓΙΚΕΣ:</b> f</p> <p><b>ΑΡΧΗ</b>   <b>ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ</b>     <b>ΓΡΑΨΕ</b> 'Δώσε πλήθος στοιχείων πίνακα Α'     <b>ΔΙΑΒΑΣΕ</b> Ν   <b>ΜΕΧΡΙΣ_ΟΤΟΥ</b> <math>N \leq 100</math> ΚΑΙ <math>N &gt; 0</math>   <b>ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ</b>     <b>ΓΡΑΨΕ</b> 'Δώσε πλήθος στοιχείων πίνακα Β'     <b>ΔΙΑΒΑΣΕ</b> Μ   <b>ΜΕΧΡΙΣ_ΟΤΟΥ</b> <math>M \leq 100</math> ΚΑΙ <math>M &gt; 0</math>   <b>ΓΡΑΨΕ</b> 'Οι αριθμοί που θα δοθούν πρέπει να είναι &amp; ταξινομημένοι κατά αύξουσα τάξη'   <b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> Ν     <b>ΓΡΑΨΕ</b> 'Δώσε το', i, ' στοιχείο του πίνακα Α'     <b>ΔΙΑΒΑΣΕ</b> Α[i]   <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> Μ     <b>ΓΡΑΨΕ</b> 'Δώσε το', i, ' στοιχείο του πίνακα Β'     <b>ΔΙΑΒΑΣΕ</b> Β[i]   <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <math>i \leftarrow 1</math>   <math>j \leftarrow 1</math>   <math>k \leftarrow 1</math>   <b>ΟΣΟ</b> (<math>i \leq N</math>) <b>ΚΑΙ</b> (<math>j \leq M</math>) <b>ΕΠΑΝΑΛΑΒΕ</b>     <b>ΑΝ</b> <math>A[i] &lt; B[j]</math> <b>ΤΟΤΕ</b>       <math>\Gamma[k] \leftarrow A[i]</math>       <math>k \leftarrow k+1</math>       <math>i \leftarrow i+1</math>     <b>ΑΛΛΙΩΣ</b>       <math>\Gamma[k] \leftarrow B[j]</math>       <math>k \leftarrow k+1</math>       <math>j \leftarrow j+1</math>     <b>ΤΕΛΟΣ_ΑΝ</b>   <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <b>ΑΝ</b> <math>i &gt; N</math> <b>ΤΟΤΕ</b>     <b>ΓΙΑ</b> λ <b>ΑΠΟ</b> k <b>ΜΕΧΡΙ</b> Μ+Ν       <math>\Gamma[\lambda] \leftarrow B[j]</math>       <math>j \leftarrow j+1</math>     <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <b>ΑΛΛΙΩΣ</b>     <b>ΓΙΑ</b> λ <b>ΑΠΟ</b> k <b>ΜΕΧΡΙ</b> Μ+Ν       <math>\Gamma[\lambda] \leftarrow A[i]</math>       <math>i \leftarrow i+1</math>     <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <b>ΤΕΛΟΣ_ΑΝ</b>   <b>ΓΙΑ</b> λ <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> Μ+Ν     <b>ΓΡΑΨΕ</b> Γ[λ]   <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>   <b>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</b></p>