

# Λίστα με προτεινόμενα Σ/Λ για εξέταση στις Πανελλήνιες στο μάθημα «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον»

Created by TuxHouse by Dimitris Vagiakakos SV1SJP

<https://www.youtube.com/@TuxHouseEdu>

<https://sv1sjp.github.io/>

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον:

270+ Προτεινόμενα Λυμένα Σ/Λ για τις Πανελλήνιες

5<sup>η</sup> έκδοση

Δημήτρης Βαγιακάκος SV1SJP

[www.youtube.com/@TuxHouseEdu](https://www.youtube.com/@TuxHouseEdu)  
<https://sv1sjp.github.io/whoami/>

ΔΩΡΕΑΝ Hardcore λυμένες Ασκήσεις, η θεωρία και βοηθητικές σημειώσεις για Πανελλήνιες

## Δικαιώματα χρήσης Panellinies\_Aepp :

Η χρήση του Panellinies\_AEPP είναι δωρεάν για όλους τους αναγνώστες. Επιπλέον, επιτρέπεται η χρήση του Panellinies\_AEPP για διδασκαλία εφόσον συνοδεύεται με το logo του TuxHouse όπως και έχει δημιουργηθεί από το [επίσημο site του Panellinies AEPP](#).

Απαγορεύεται οποιαδήποτε πώληση οποιουδήποτε μέρους του συγκεκριμένου έργου, όπως επίσης και η μεταφόρτωση του σε servers τρίτων χωρίς την γραπτή συγκατάθεση μου. Για οποιαδήποτε περαιτέρω πληροφορία, μπορείτε να επικοινωνήσετε μαζί μου στα social media του TuxHouse οπού θα βρείτε εδώ: <https://sv1sjp.github.io/whoami/>

**(Όπου 0 τα λάθος και 1 τα σωστά):**

- 1) Σε μία γλώσσα προγραμματισμού η σημασιολογία των λέξεων της γλώσσας καθορίζεται από το δημιουργό της γλώσσας. 1
- 2) Η κλήση μίας συνάρτησης γίνεται με την εντολή ΚΑΛΕΣΕ. 0
- 3) Η λειτουργία επί των δομών δεδομένων "Εισαγωγή" αποτελεί μια από τις τυπικές λειτουργίες των πινάκων. 0
- 4) Η δομή επανάληψης ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ\_ΒΗΜΑ 10 θα εκτελεστεί ακριβώς 2 φορές. 0
- 5) Ο μεταγλωττιστής, εφόσον δεν βρει συντακτικά λάθη σε ένα πρόγραμμα, παράγει το εκτελέσιμο πρόγραμμα. 0
- 6) Η απεριόριστη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να δηλώνονται σε αυτό το τμήμα. 0
- 7) Μια συνάρτηση μπορεί να εκτελέσει τις ίδιες ακριβώς λειτουργίες με μια διαδικασία. 0
- 8) Οι τελεστές DIV και MOD μπορούν να χρησιμοποιηθούν με οποιοδήποτε αριθμητικό τύπο δεδομένων. 0
- 9) Η δυαδική αναζήτηση έχει μικρότερη χρονική πολυπλοκότητα από τη σειριακή. 1
- 10) Η επανάληψη ΓΙΑ i ΑΠΟ 5 ΜΕΧΡΙ 1 θα εκτελεστεί 5 φορές. 0
- 11) Κατά την δημιουργία ενός προγράμματος χρησιμοποιούνται κατά σειρά 1.συντάκτης 2. συνδέτης – φορτωτής 3. μεταγλωττιστής ή διερμηνευτής. 0
- 12) Η συνθήκη στην εντολή «Όσο...επανάλαβε» ελέγχεται τουλάχιστον μια φορά. 1.
- 13) Η συνθήκη στην εντολή «Όσο...επανάλαβε» εκτελείται τουλάχιστον μια φορά. 0
- 14) Η σειριακή αναζήτηση έχει μικρότερη χρονική πολυπλοκότητα από τη δυαδική. 0
- 15) Η συνθήκη στην εντολή «Αρχή Επανάληψης ... Μέχρις\_Οτου» ελέγχεται τουλάχιστον μια φορά. 1
- 16) Η συνθήκη στην εντολή «Αρχή Επανάληψης ... Μέχρις\_Οτου» εκτελείται τουλάχιστον μια φορά. 1
- 17) Η συνθήκη στην εντολή «Αρχή Επανάληψης ... Μέχρις\_Οτου» ελέγχεται το πολύ μια φορά. 0
- 18) Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος. 1
- 19) Μετά από την εκτέλεση της εντολής  $\Sigma \leftarrow \Sigma + A$ , αν A θετικό, η τιμή της μεταβλητής  $\Sigma$  είναι πάντοτε μεγαλύτερη από την τιμή που είχε πριν από την εκτέλεση της εντολής. 1

- 20) Μετά από την εκτέλεση της εντολής  $\Sigma \leftarrow \Sigma + A$ , η τιμή της μεταβλητής  $\Sigma$  είναι πάντοτε μεγαλύτερη από την τιμή που είχε πριν από την εκτέλεση της εντολής. 0 (μπορεί το  $A$  να είναι αρνητικό).
- 21) Πάντα μία ΟΣΟ θα εκτελεστεί τουλάχιστον 1 φορά. 0
- 22) Μετά από την εκτέλεση της εντολής  $\Sigma \leftarrow \Sigma * A$ , η τιμή της μεταβλητής  $\Sigma$  είναι πάντοτε μεγαλύτερη από την τιμή που είχε πριν από την εκτέλεση της εντολής. 0
- 23) Κάθε μεταβλητή παίρνει τιμή μόνο με εντολή εκχώρησης. 0
- 24) Η έκφραση  $\text{OXI}(K=10 \text{ ΚΑΙ } X>7)$  είναι ισοδύναμη με την έκφραση  $(K \neq 10 \vee X \leq 7)$ . 1
- 25) Η έκφραση  $\text{OXI}(K=10 \text{ ΚΑΙ } X>7)$  είναι ισοδύναμη με την έκφραση  $(\text{OXI}(K=10) \vee X \leq 7)$ . 1
- 26) Οι συγκριτικοί τελεστές δεν έχουν μεγαλύτερη ιεραρχία από τους λογικούς τελεστές. 0.
- 27) Πολύ συχνά οι εντολές που έχουν γραφτεί με εμφωλευμένα AN μπορούν να γραφτούν πιο απλά χρησιμοποιώντας σύνθετες εκφράσεις ή την εντολή επιλογής AN... ΤΟΤΕ... ΑΛΛΙΩΣ\_ΑΝ. 1
- 28) Οι λειτουργίες ώθηση και απώθηση είναι οι κύριες λειτουργίες σε μια ουρά. 0
- 29) Οι πίνακες είναι δομές δεδομένων της κύριας μνήμης. 1
- 30) Τα συντακτικά λάθη ενός προγράμματος μπορούν να διορθωθούν κατά την εκτέλεση του προγράμματος, ταυτόχρονα με τα λογικά λάθη. 0
- 31) Ένα άλυτο πρόβλημα δεν μπορεί μελλοντικά να χαρακτηριστεί επιλύσιμο. 0
- 32) Αν σε έναν ακέραιο αριθμό εφαρμοστεί ολίσθηση προς τα δεξιά και στον αριθμό που προκύψει εφαρμοστεί ολίσθηση προς τα αριστερά, τότε ο τελικός αριθμός θα είναι πάντα ίσος με τον αρχικό. 0
- 33) Πρόβλημα είναι μια οποιαδήποτε κατάσταση που πρέπει να αντιμετωπίσουμε. 1
- 34) Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων. 1.
- 35) Ο αλγόριθμος είναι απαραίτητος μόνο για την επίλυση προβλημάτων Πληροφορικής. 0
- 36) Ο αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο εντολών. 1
- 37) Ο αλγόριθμος μπορεί να περιλαμβάνει και εντολές που δεν είναι σαφείς. 0

- 38) Η Πληροφορική μελετά τους αλγορίθμους μόνο από το πρίσμα των γλωσσών προγραμματισμού. 0.
- 39) Η αναπαράσταση των αλγορίθμων μπορεί να γίνει μόνο με χρήση ελεύθερου κειμένου και φυσικής γλώσσας. 0.
- 40) Τα κυριότερα σύμβολα των διαγραμμάτων ροής είναι η έλλειψη, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο. 1
- 41) Τα κυριότερα σύμβολα των διαγραμμάτων ροής είναι ο κύκλος, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο. 0
- 42) Η δομή της ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση πολύπλοκων προβλημάτων. 0
- 43) Η συνθήκη που ελέγχεται σε μια δομή επιλογής μπορεί να πάρει περισσότερες από δύο διαφορετικές τιμές. 0
- 44) Η ομάδα εντολών που περιέχεται σε μια δομή επιλογής μπορεί να μην εκτελεστεί. 1
- 45) Η ομάδα εντολών που περιέχεται σε μια δομή επιλογής, μπορεί και να εκτελεστεί. 1
- 46) Εμφωλευμένα ΑΝ ονομάζονται δύο ή περισσότερες εντολές της μορφής ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ που περιέχονται η μία μέσα στην άλλη. 1
- 47) Πολλαπλές επιλογές μπορούν να γίνουν και με μία εμφωλευμένη δομή. 1
- 48) Ο τελεστής ΚΑΙ αντιστοιχεί στη λογική πράξη της σύζευξης. 1
- 49) Ο βρόχος Για Κ από 5 μέχρι 1 εκτελείται 5 φορές. 0
- 50) Στην εντολή ΓΙΑ ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό επαναλήψεων. 1
- 51) Στην εντολή Όσο ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό επαναλήψεων. 0
- 52) Ο βρόχος Για Κ από 5 μέχρι 5 δεν εκτελείται καμία φορά. 0
- 53) Ο βρόχος Για κ από -4 μέχρι -3 εκτελείται ακριβώς δύο φορές. 1
- 54) Ο βρόχος Για κ από -4 μέχρι -3 εκτελείται τουλάχιστον 2 φορές. 0

- 55) Στην επαναληπτική δομή Όσο ... Επανάλαβε δεν γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων. 1
- 56) Στην επαναληπτική δομή Όσο ... Επανάλαβε γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων. 0
- 57) Στην επαναληπτική δομή Για, δεν γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων. 0
- 58) Η εντολή επανάληψης ΜΕΧΡΙΣ\_ΟΤΟΥ εκτελείται υποχρεωτικά τουλάχιστον μία φορά. 1.
- 59) Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με χρήση της εντολής ΓΙΑ ΑΠΟ ΜΕΧΡΙ. 0
- 60) Ένα υποπρόγραμμα μπορεί να καλείται από ένα άλλο υποπρόγραμμα ή από το κύριο πρόγραμμα. 1
- 61) Ένα υποπρόγραμμα μπορεί να καλείται από μόνο από το κύριο πρόγραμμα. 0
- 62) Μια διαδικασία και μια συνάρτηση μπορούν να εκτελούν ακριβώς τις ίδιες λειτουργίες. 0
- 63) Κάθε υποπρόγραμμα πρέπει να έχει μόνο μία είσοδο και μία έξοδο. 1
- 64) Μια συνάρτηση υπολογίζει και επιστρέφει παραπάνω από μία τιμές με το όνομά της. 0
- 65) Ο τρόπος κλήσης των διαδικασιών και των συναρτήσεων είναι ίδιος, ενώ ο τρόπος σύνταξής τους είναι διαφορετικός. 0
- 66) Μία συνάρτηση είναι δυνατό να επιστρέφει μόνον ακέριαιες ή πραγματικές τιμές. 0
- 67) Οι συναρτήσεις μπορούν να επιστρέφουν μόνο μία τιμή. 1
- 68) Στη διαδικασία η λίστα παραμέτρων είναι υποχρεωτική. 0
- 69) Η κλήση των διαδικασιών γίνεται με απλή αναφορά του ονόματος τους. 0
- 70) Το πλήθος των τυπικών και των πραγματικών παραμέτρων πρέπει να είναι ίδιο. 1
- 71) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για εισαγωγή 3 δεδομένων. 0
- 72) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για εισαγωγή 1 δεδομένου. 0
- 73) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για υπολογισμό του μικρότερου από 7 αριθμούς. 1
- 74) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για υπολογισμό 2 μικρότερων από 7 αριθμούς. 0

- 75) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για να ελέγξουμε αν 2 αριθμοί είναι ίσοι. 1
- 76) Μπορούμε να φτιάξουμε υποπρόγραμμα διαδικασία για να ταξινομήσουμε έναν πίνακα. 1
- 77) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για να ταξινομήσουμε 2 πίνακες. 0
- 78) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για να ελέγχει αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο. 1
- 79) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για να διαβάσει ένα αριθμό και να επιστρέφει το τετράγωνο του. 0
- 80) Μπορούμε να φτιάξουμε υποπρόγραμμα συνάρτηση για να ελέγχει αν ένας αριθμός είναι άρτιος. 1
- 81) Δυναμικές είναι οι δομές που αποθηκεύονται σε συνεχόμενες θέσεις μνήμης. 0
- 82) Ένας πίνακας έχει σταθερό μέγεθος αλλά μεταβαλλόμενο περιεχόμενο. 1
- 83) Ένας πίνακας μπορεί να αποθηκεύσει και ακεραίους αλλά και πραγματικούς αριθμούς. 0
- 84) Οι δυναμικές δομές έχουν σταθερό μέγεθος. 0
- 85) Σε μία δυναμική δομή δεδομένων τα δεδομένα αποθηκεύονται υποχρεωτικά σε συνεχόμενες θέσεις μνήμης. 0
- 86) Η δυναμική παραχώρηση μνήμης χρησιμοποιείται στις στατικές δομές δεδομένων. 0
- 87) Η δυναμική παραχώρηση μνήμης χρησιμοποιείται στις δομές των πινάκων. 0
- 88) Η ουρά και η στοίβα μπορούν να υλοποιηθούν με δομή πίνακα. 1
- 89) Η απόθεση (pop) στοιχείου γίνεται από το πίσω άκρο της στοίβας. 0
- 90) Κατά τη διαδικασία της ώθησης πρέπει να ελέγχεται αν η στοίβα είναι γεμάτη. 1
- 91) Οι λειτουργίες ώθηση και απόθεση είναι οι κύριες λειτουργίες σε μία στοίβα. 1
- 92) Η στοίβα χρησιμοποιεί δύο δείκτες. 0
- 93) Η λειτουργία της ώθησης σχετίζεται με τη δομή της στοίβας. 1
- 94) Η εξαγωγή (dequeue) στοιχείου γίνεται από το εμπρός άκρο της ουράς. 1
- 95) Η ώθηση (push) στοιχείου είναι μία από τις λειτουργίες της ουράς. 0

- 96) Η μέθοδος επεξεργασίας FIFO εφαρμόζεται στη λειτουργία της ουράς. 1
- 97) Η σειριακή αναζήτηση χρησιμοποιείται αποκλειστικά στους ταξινομημένους πίνακες. 0
- 98) Η δυαδική αναζήτηση χρησιμοποιείται αποκλειστικά στους ταξινομημένους πίνακες. 1
- 99) Η δυαδική αναζήτηση χρησιμοποιείται στους μη ταξινομημένους πίνακες. 0
- 100) Όταν γίνεται σειριακή αναζήτηση κάποιου στοιχείου σε έναν μη ταξινομημένο πίνακα και το στοιχείο δεν υπάρχει στον πίνακα, τότε υποχρεωτικά προσπελούνται όλα τα στοιχεία του πίνακα. 1
- 101) Η μέθοδος της σειριακής αναζήτησης δικαιολογείται στην περίπτωση που ο πίνακας είναι μη ταξινομημένος και μικρού μεγέθους. 1
- 102) Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος αλγόριθμος ταξινόμησης. 0
- 103) Η ταξινόμηση φυσαλίδας είναι ο πιο απλός αλγόριθμος ταξινόμησης. 1
- 104) Η ταξινόμηση φυσαλίδας είναι ο πιο γρήγορος αλγόριθμος ταξινόμησης. 0
- 105) Η εγγραφή είναι δομή δεδομένων η οποία αποτελείται από πεδία που αποθηκεύουν χαρακτηριστικά. 1
- 106) Η ταξινόμηση είναι χρήσιμη διαδικασία γιατί έτσι εκτελείται γρηγορότερα η αναζήτηση. 1
- 107) Η ταξινόμηση είναι χρήσιμη διαδικασία γιατί έτσι εκτελείται γρηγορότερα η προσπέλαση. 1
- 108) Η ταξινόμηση ευθείας ανταλλαγής είναι πολύ αποτελεσματική σε πίνακες που είναι ταξινομημένοι κατά την αντίστροφη φορά σε σχέση με την επιθυμητή. 0
- 109) Ένα πρόγραμμα σε γλώσσα μηχανής είναι μια ακολουθία δυαδικών ψηφίων. 1
- 110) Ένα πρόγραμμα σε γλώσσα μηχανής χρειάζεται μεταγλώττιση. 0
- 111) Για την εκτέλεση μιας εντολής συμβολικής γλώσσας απαιτείται η μετάφρασή της σε γλώσσα μηχανής. 1
- 112) Οι εντολές στις συμβολικές γλώσσες αποτελούνται από ακολουθίες 0 και 1. 0
- 113) Τα προγράμματα σε γλώσσες υψηλού επιπέδου είναι ανεξάρτητα του υπολογιστή που αναπτύχθηκαν. 1

- 114) Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος είναι απαραίτητη στο δομημένο προγραμματισμό. 0
- 115) Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος καλό είναι να αποφεύγεται η χρήση της. 1
- 116) Ο δομημένος προγραμματισμός εξασφαλίζει τη δημιουργία σωστών προγραμμάτων. 0
- 117) Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα. 1
- 118) Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα. 0
- 119) Ο συντάκτης μας επιτρέπει να συντάσσουμε ένα πρόγραμμα. 1
- 120) Ο μεταγλωττιστής δέχεται στην είσοδό του ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. 1
- 121) Το πηγαίο πρόγραμμα εκτελείται από τον υπολογιστή χωρίς μεταγλώττιση. 0
- 122) Ο διερμηνευτής διαβάζει μία προς μία τις εντολές του πηγαίου προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής. 1
- 123) Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. 1
- 124) Τα συντακτικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. 0
- 125) Τα συντακτικά λάθη στον πηγαίο κώδικα εμφανίζονται κατά το στάδιο της μεταγλώττισής του. 1
- 126) Ένα πρόγραμμα δεν μπορεί να μεταγλωττιστεί αν έχει συντακτικά λάθη. 1
- 127) Ένα πρόγραμμα δεν μπορεί να μεταγλωττιστεί αν έχει λογικά λάθη. 0
- 128) Το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται εκτελέσιμο. 0
- 129) Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση. 0
- 130) Οι τύποι των μεταβλητών που υποστηρίζει η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ. 0
- 131) Στη δομή ενός προγράμματος το τμήμα δήλωσης των σταθερών ακολουθεί το τμήμα δήλωσης των μεταβλητών. 1
- 132) Οι δεσμευμένες λέξεις της ΓΛΩΣΣΑΣ δεν μπορούν να χρησιμοποιηθούν ως ονόματα δεδομένων σε ένα πρόγραμμα. 1



- 133) Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή και ο τύπος μιας μεταβλητής. 0
- 134) Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή μίας μεταβλητής. 1
- 135) Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή μίας σταθεράς. 0
- 136) Ενώ η τιμή μίας μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της. 1
- 137) Σε μία εντολή εκχώρησης του αποτελέσματος μίας έκφρασης σε μία μεταβλητή, η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου. 1
- 138) Οι μεταβλητές που χρησιμοποιούνται σ' ένα πρόγραμμα αντιστοιχίζονται από το μεταγλωττιστή σε συγκεκριμένες θέσεις της μνήμης του υπολογιστή. 1
- 139) Η τιμή της μεταβλητής είναι το περιεχόμενο της αντίστοιχης θέσης μνήμης και δεν μεταβάλλεται στη διάρκεια εκτέλεσης του προγράμματος. 0
- 140) Ο τύπος της μεταβλητής αλλάζει κατά την εκτέλεση του προγράμματος. 0
- 141) Η δήλωση των μεταβλητών που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική. 1
- 142) Οι λογικές μεταβλητές δέχονται μόνο δύο τιμές. 1
- 143) Ο τελεστής MOD χρησιμοποιείται για τον υπολογισμό του ηλίκου μίας διαίρεσης ακεραίων αριθμών. 0
- 144)  $A\_M(X)$  είναι η συνάρτηση της ΓΛΩΣΣΑΣ που υπολογίζει την απόλυτη τιμή του  $X$ . 0
- 145)  $A\_M(X)$  είναι η συνάρτηση της ΓΛΩΣΣΑΣ που υπολογίζει το ακέραιο μέρος του  $X$ . 1
- 146) Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μία έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. 1
- 147) Στην αριθμητική έκφραση  $A+B*\Gamma$  εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός. 0
- 148) Το σύμβολο της εντολής εκχώρησης είναι το ίσον =. 0
- 149) Κατά τον υπολογισμό μιας αριθμητικής παράστασης, πρώτα εκτελείται ο πολλαπλασιασμός και στη συνέχεια η πρόσθεση. 1
- 150) Κατά την εκτέλεση του προγράμματος η εντολή ΔΙΑΒΑΣΕ διακόπτει την εκτέλεσή του και περιμένει την εισαγωγή τιμών από το πληκτρολόγιο. 1

- 151) Η ιεραρχία των λογικών τελεστών είναι μικρότερη των αριθμητικών. 1
- 152) Η σύγκριση λογικών δεδομένων έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>). 1
- 153) Κάθε εντολή AN περιλαμβάνει υποχρεωτικά το τμήμα ΑΛΛΙΩΣ. 0
- 154) Κάθε τμήμα προγράμματος που χρησιμοποιεί την εντολή ΕΠΙΛΕΞΕ μπορεί να γραφεί και με εντολές AN. 1
- 155) Η χρήση εμφωλευμένων AN είναι καλή προγραμματιστική τακτική. 0
- 156) Κάθε εντολή AN πρέπει να έχει την αντίστοιχη εντολή ΤΕΛΟΣ\_ΑΝ. 1
- 157) Αν το A έχει την τιμή 5 και το B την τιμή 6 τότε η λογική έκφραση  $A > 5 \text{ 'H } A < 3 \text{ ΚΑΙ } B > 5$  είναι ψευδής. 1
- 158) Αν το A έχει την τιμή 10 και το B την τιμή 20 τότε η έκφραση  $(A > 8 \text{ ΚΑΙ } B < 20) \text{ 'H } (A > 10 \text{ 'H } B = 10)$  είναι αληθής. 0
- 159) Η τιμή του βήματος στην εντολή ΓΙΑ είναι υποχρεωτική να αναγράφεται. 0
- 160) Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με χρήση της εντολής ΓΙΑ. 0
- 161) Η ΓΛΩΣΣΑ υποστηρίζει τρεις εντολές επανάληψης, την εντολή ΟΣΟ, την εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ και την εντολή ΓΙΑ. 1
- 162) Όταν ένας βρόχος είναι εμφωλευμένος σε άλλο, ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος. 1
- 163) Η είσοδος σε κάθε βρόχο επανάληψης υποχρεωτικά γίνεται από την αρχή του. 1
- 164) Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου. 1
- 165) Οι εντολές που βρίσκονται σε μία επανάληψη ΓΙΑ εκτελούνται τουλάχιστο μία φορά. 0
- 166) Κάθε επανάληψη μπορεί να γραφεί με την εντολή ΟΣΟ-ΕΠΑΝΑΛΑΒΕ. 1
- 167) Σε περίπτωση εμφωλευμένων βρόχων, ο εσωτερικός πρέπει να περικλείεται ολόκληρος στον εξωτερικό. 1
- 168) Η τιμή του βήματος αναφέρεται υποχρεωτικά σε κάθε εντολή ΓΙΑ. 0
- 169) Η δήλωση των πινάκων που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική. 1
- 170) Τα στοιχεία ενός πίνακα πρέπει να είναι του ιδίου τύπου. 1

- 171) Όλοι οι πίνακες δηλώνονται στο τμήμα δήλωσης μεταβλητών του προγράμματος. 1
- 172) Ο δείκτης ενός μονοδιάστατου πίνακα πρέπει να είναι πάντα I. 0
- 173) Σε ένα μεγάλο και σύνθετο πρόγραμμα, η άσκοπη χρήση μεγάλων πινάκων μπορεί να οδηγήσει ακόμη και σε αδυναμία εκτέλεσης του προγράμματος. 1
- 174) Οι πίνακες πρέπει να χρησιμοποιούνται πάντα όταν αυτό είναι δυνατό. 0
- 175) Η χρήση των πινάκων σε ένα πρόγραμμα αυξάνει την απαιτούμενη μνήμη. 1
- 176) Σκοπός της συγχώνευσης δύο ταξινομημένων πινάκων είναι η δημιουργία ενός τρίτου ταξινομημένου πίνακα, που περιέχει τα στοιχεία των δύο πινάκων. 1
- 177) Για την ταξινόμηση ενός πίνακα 100 στοιχείων μπορεί να χρησιμοποιηθεί μόνο μία μέθοδος. 0
- 178) Η ταξινόμηση των στοιχείων ενός πίνακα πρέπει να γίνεται πάντα πριν από την αναζήτηση. 0
- 179) Η δυαδική αναζήτηση είναι πάντα ο καλύτερος τρόπος αναζήτησης 0
- 180) Τα ονόματα και τα εισιτήρια 10 θεάτρων μπορούν να αποθηκευτούν σε ένα δισδιάστατο πίνακα. 0
- 181) Οι δισδιάστατοι πίνακες είναι οι πίνακες με τη μεγαλύτερη διάσταση που χειρίζονται οι γλώσσες προγραμματισμού. 0
- 182) Η ταξινόμηση αποτελεί τυπική επεξεργασία σε έναν πίνακα. 1
- 183) Η προσθήκη στοιχείων αποτελεί τυπική επεξεργασία σε έναν πίνακα. 1
- 184) Ο πολλαπλασιασμός στοιχείων αποτελεί τυπική επεξεργασία σε έναν πίνακα. 0
- 185) Η συγχώνευση αποτελεί τυπική επεξεργασία σε έναν πίνακα. 1
- 186) Η αναζήτηση αποτελεί τυπική επεξεργασία σε έναν πίνακα. 1
- 187) Η επιλογή του καλύτερου αλγόριθμου ταξινόμησης εξαρτάται από τη διάσταση του πίνακα 0
- 188) Η επιλογή του καλύτερου αλγόριθμου ταξινόμησης εξαρτάται από το πλήθος των στοιχείων του πίνακα . 1
- 189) Η επιλογή του καλύτερου αλγόριθμου ταξινόμησης εξαρτάται από την αρχική διάταξη των στοιχείων. 1

- 190) Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. 1
- 191) Η εντολή Αρχή\_επανάληψης... Μέχρις\_ότου Εκτελείται οπωσδήποτε μια φορά 1.
- 192) Το τμήμα του αλγορίθμου που επαναλαμβάνεται, δηλαδή από την εντολή Όσο μέχρι το τέλος\_επανάληψης αποκαλείται βρόχος.1
- 193) Η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί δύο, ενώ η ολίσθηση προς τα δεξιά ισοδυναμεί με την ακέραια διαίρεση διά δύο.1
- 194) Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών. 1
- 195) Αλγόριθμοι+ Δομές δεδομένων = ΠΡΟΓΡΑΜΜΑ 1.
- 196) Στη Συγχώνευση (merging), δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή. 1
- 197) Οι δυναμικές δομές δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης (dynamic memory allocation).1
- 198) . Όλες οι σύγχρονες γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης. 1
- 199) Όμως, γενικά η αναφορά στα στοιχεία ενός πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα ακολουθούμενου από την τιμή ενός ή περισσότερων δεικτών (indexes) σε παρένθεση ή αγκύλη. 1
- 200) Δοθέντων των στοιχείων  $a_1, a_2, \dots, a_n$  η ταξινόμηση συνίσταται στη μετάθεση (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά  $a_{k_1}, a_{k_2}, \dots, a_{k_n}$  έτσι ώστε, δοθείσης μίας συνάρτησης διάταξης (ordering function),  $f$ , να ισχύει:
- $$f(a_{k_1}) \leq f(a_{k_2}) \leq \dots \leq f(a_{k_n}) . 1$$
- 201) Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερη αποτελεσματική μέθοδος αναζήτησης. 1
- 202) Τα στοιχεία ενός αρχείου ονομάζονται εγγραφές 1.
- 203) κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία (fields), που ταυτοποιούν την εγγραφή . 1
- 204) Αλφάβητο μίας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα. 1

- 205) Τυπικό : Είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή. 1
- 206) : Η σημασιολογία (Semantics) είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατά επέκταση των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μία γλώσσα. 1
- 207) Το Λεξιλόγιο αποτελείται από ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου, τις λέξεις που είναι δεκτές από τη γλώσσα. 1
- 208) Ένα λογικό λάθος, δεν έχει να κάνει με την λανθασμένη σύνταξη μιας εντολής. 1
- 209) Ο διερμηνευτής διαβάσει μία προς μία τις εντολές του αρχικού προγράμματος και για καθεμία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής. 1
- 210) Το αρχικό πρόγραμμα λέγεται πηγαίο πρόγραμμα .1
- 211) το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται αντικείμενο πρόγραμμα . 1
- 212) Το αποτέλεσμα του συνδέτη είναι η παραγωγή του εκτελέσιμου προγράμματος.1
- 213) Το αποτέλεσμα του συνδέτη είναι η παραγωγή του εκτελέσιμου προγράμματος (executable), το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή. 1
- 214) .Η δημιουργία του εκτελέσιμου προγράμματος γίνεται μόνο στην περίπτωση που το αρχικό πρόγραμμα δεν περιέχει συντακτικά λάθη . 1
- 215) .Η δημιουργία του εκτελέσιμου προγράμματος γίνεται μόνο στην περίπτωση που το αρχικό πρόγραμμα δεν περιέχει λογικά λάθη. 0
- 216) . Τα λάθη του προγράμματος είναι γενικά δύο ειδών, λογικά και συντακτικά . 1
- 217) Κάθε γλώσσα προσδιορίζεται από το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τη σημασιολογία της . 1
- 218) Η ιεραρχική σχεδίαση ή ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα. 1
- 219) Τα σύγχρονα ολοκληρωμένα προγραμματιστικά περιβάλλοντα δεν παρέχουν απλώς ένα μεταφραστή μιας γλώσσας προγραμματισμού. Περιέχουν όλα τα προγράμματα και τα εργαλεία που απαιτούνται και βοηθούν τη συγγραφή, την εκτέλεση και κύρια τη διόρθωση των προγραμμάτων.1

- 220) Οι ακέραιοι μπορούν να είναι θετικοί, αρνητικοί ή μηδέν. 1.
- 221) Μία μεταβλητή ή σταθερά, μπορεί το όνομά της να ξεκινάει με αριθμό. 0
- 222) Μία μεταβλητή ή σταθερά, μπορεί το όνομά της να ξεκινάει με «». 0
- 223) Το όνομα μεταβλητής μπάμπης\_σουβλακια είναι αποδεκτό από την ΓΛΩΣΣΑ. 1
- 224) Το όνομα μεταβλητής ΠΡΟΑΣΤΙΑΚΟΣ ΣΙΔΗΡΟΔΡΟΜΟΣ είναι αποδεκτό από την ΓΛΩΣΣΑ. 0
- 225) Η συνάρτηση  $A\_T(x)$ , υπολογίζει την απόλυτη τιμή του X. 1
- 226) Η συνάρτηση  $A\_M(x)$ , υπολογίζει την απόλυτη τιμή του X. 0
- 227) Η συνάρτηση  $\Lambda O\Gamma(x)$ , υπολογίζει τον λογάριθμο του X. 1
- 228) Η συνάρτηση  $E(x)$ , υπολογίζει το  $e^x$ . 1
- 229) Η διαίρεση έχει την μεγαλύτερη ιεραρχία. 0
- 230) Ο τελεστής `div` χρησιμοποιείται για τον υπολογισμό του πηλίκου μιας διαίρεσης ακεραίων αριθμών. 1
- 231) Ο τελεστής `mod` χρησιμοποιείται για τον υπολογισμό του υπόλοιπου μιας διαίρεσης ακεραίων αριθμών. 1
- 232) Μπορούμε να γράψουμε σε μία γραμμή πάνω από μία εντολή. 0
- 233) Τα δεδομένα στην πραγματικότητα καταχωρούνται στην μνήμη RAM του υπολογιστή. 1
- 234) Σε μια εντολή εκχώρησης η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου. 1
- 235) Πάντα πρέπει να χρησιμοποιούνται ζεύγη παρενθέσεων. Διαφορετικός αριθμός αριστερών από δεξιές παρενθέσεις στην ίδια έκφραση είναι ένα από τα πιο συνηθισμένα λάθη. 1
- 236) Στην ουρά, ο,τι εισέρχεται τελευταίο (Last in), εξέρχεται πρώτο (First out). 0
- 237) Τοποθέτηση στοιχείου στην κορυφή μιας στοίβας. ονομάζεται ώθηση (push). 1
- 238) Απομάκρυνση στοιχείου από την κορυφή. ονομάζεται απώθηση (pop). 1
- 239) Η διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχει, αν η στοίβα είναι γεμάτη, οπότε λέγεται ότι συμβαίνει υπερχειλίση (overflow) της στοίβας. 1

- 240) Η δομή στοίβα μπορεί να υλοποιηθεί με :  
έναν πίνακα και έναν δείκτη στο στοιχείο της κορυφής της  
στοίβας. (μια βοηθητική μεταβλητή). 1
- 241) Στην ουρά , ο,τι εισέρχεται πρώτο (First in) ,
- 242) εξέρχεται πρώτο (First out). 1
- 243) Στην στοίβα , ο,τι εισέρχεται τελευταίο (Last in) ,
- 244) εξέρχεται πρώτο (First out) . 1
- 245) Τμηματικός προγραμματισμός ονομάζεται η τεχνική  
σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο  
από απλούστερα τμή-ματα προγραμμάτων.1
- 246) Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα  
άλλα. 1
- 247) Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία  
λειτουργία 1.
- 248) Μία παράμετρος είναι μία μεταβλητή που επιτρέπει  
το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε  
ένα άλλο.1
- 249) Η διαδικασία είναι ένας τύπος υποπρογράμματος που  
μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος.1
- 250) Η απεριόριστη εμβέλεια υποχρεώνει όλες τις μεταβλητές  
που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να  
δηλώνονται σε αυτό το τμήμα. 0
- 251) Αναδρομή ονομάζεται η δυνατότητα ενός  
υποπρογράμματος να καλεί τον εαυτό του.
- 252) Η χρήση δομημένου προγραμματισμού, επεκτείνει τις  
δυνατότητες των γλωσσών προγραμματισμού. 0
- 253) Η χρήση τμηματικού προγραμματισμού, διευκολύνει την  
ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος. 1
- 254) Ο έλεγχος μαύρου κουτιού (black-box testing) είναι μια  
δημοφιλής τεχνική ελέγχου. 1
- 255) Στον έλεγχο μαύρου κουτιού, οι ελεγκτές εξετάζουν  
ενδεδειγμένα τον κώδικα του προγράμματος. 0
- 256) Ο έλεγχος μαύρου κουτιού ονομάζεται έτσι λόγω της  
«αγνόησης» του κώδικα του προγράμματος κατά τη διαδικασία  
του ελέγχου. 1
- 257) Στον έλεγχο μαύρου κουτιού, τα δεδομένα εισόδου  
προέρχονται απευθείας από τον κώδικα του προγράμματος. 0

- 258) Τα δεδομένα εισόδου για τον έλεγχο μαύρου κουτιού βασίζονται στις προδιαγραφές του προγράμματος. 1
- 259) Η εκσφαλμάτωση είναι μια εύκολη διαδικασία και δεν απαιτεί βαθιά γνώση της γλώσσας προγραμματισμού. 0
- 260) Η εκσφαλμάτωση είναι ένα πρόβλημα λογικής και η καλή κατανόηση του προγράμματος βοηθάει στον εντοπισμό των λαθών. 1
- 261) Τα συντακτικά λάθη συνήθως ανιχνεύονται αυτόματα από το περιβάλλον προγραμματισμού. 1
- 262) Κατά τη φάση της εκσφαλμάτωσης, τα λογικά λάθη και τα λάθη κατά το χρόνο εκτέλεσης είναι τα κυριότερα είδη λαθών που απασχολούν τους προγραμματιστές. 1
- 263) Εργαλεία εκσφαλμάτωσης και ειδικές εντολές ή συναρτήσεις που προσφέρει το περιβάλλον προγραμματισμού μπορούν να βοηθήσουν στην εκσφαλμάτωση λαθών κατά το χρόνο εκτέλεσης. 1
- 264) Η αντικειμενοστραφής σχεδίαση έχει ως πρωτεύον δομικό στοιχείο τους αλγόριθμους και όχι τα δεδομένα. 0
- 265) Ο αντικειμενοστραφής προγραμματισμός βοηθά στη δημιουργία προγραμμάτων που είναι ευέλικτα και επαναχρησιμοποιήσιμα. 1
- 266) Ο αντικειμενοστραφής προγραμματισμός ακολουθεί τις αρχές του δομημένου προγραμματισμού και την ιεραρχική σχεδίαση. 1
- 267) Η δημιουργία κλάσεων στον αντικειμενοστραφή προγραμματισμό επιτρέπει την οργάνωση κοινών χαρακτηριστικών για διαφορετικά αντικείμενα. 1
- 268) Στον αντικειμενοστραφή προγραμματισμό, τα αντικείμενα παίρνουν ως default τα κοινά χαρακτηριστικά από την κλάση που ορίζουμε και στη συνέχεια, προσθέτουμε στο αντικείμενο τα δικά του χαρακτηριστικά. 1



- 269) Στον αντικειμενοστραφή προγραμματισμό, μία κλάση μπορεί να περιγράψει τα κοινά χαρακτηριστικά διαφορετικών ζώων, όπως μύτη, μάτια και αυτιά. 1
- 270) Η δημιουργία αντικειμένων που περιέχουν κοινά χαρακτηριστικά από μία κλάση είναι ένας αποτελεσματικός τρόπος να αποφύγουμε την επανάληψη του ίδιου κώδικα. 1
- 271) Ο αντικειμενοστραφής προγραμματισμός βασίζεται στην έννοια των κλάσεων, οι οποίες περιέχουν μεθόδους και ιδιότητες. 1
- 272) Ο αντικειμενοστραφής προγραμματισμός δεν επιτρέπει την επαναχρησιμοποίηση του κώδικα. 0
- 273) Στην αντικειμενοστραφή σχεδίαση, τα αντικείμενα δημιουργούνται από τα δεδομένα που μορφοποιούνται μέσα στις κλάσεις. 1
- 274) Η ιεραρχική σχεδίαση δεν χρησιμοποιείται στον αντικειμενοστραφή προγραμματισμό. 0
- 275) Η κληρονομικότητα είναι μία βασική αρχή του αντικειμενοστραφούς προγραμματισμού. 1
- 276) Χρησιμοποιώντας την τεχνική "Διαίρει και Βασίλευε" υποδιαιρούμε ένα πρόβλημα σε μικρότερα υποπροβλήματα με την ίδια τυποποίηση αλλά μικρότερα σε μέγεθος. 1
- 277) Η Δυαδική Αναζήτηση δεν ακολουθεί τη φιλοσοφία της μεθόδου "Διαίρει και Βασίλευε". 0