

Επιλέξτε Σωστό ή Λάθος για καθένα από τα παρακάτω:**Κεφ. 1 σχολικού βιβλίου**

1. Επιλύσιμο είναι ένα πρόβλημα για το οποίο ξέρουμε ότι έχει λύση, αλλά αυτή δεν έχει βρεθεί ακόμη.
2. Για να αναπαραστήσουμε τα δεδομένα και τα αποτελέσματα σε έναν αλγόριθμο, χρησιμοποιούμε μόνο σταθερές.
3. Ο Η/Υ δεν μπορεί να επιτελέσει όλες τις λειτουργίες του ανθρώπινου εγκεφάλου.
4. Αν ένα πρόβλημα απαιτεί απλούς υπολογισμούς σε μικρό όγκο δεδομένων δεν μπορεί να ανατεθεί σε έναν Η/Υ.
5. Για κάθε πρόβλημα υπάρχει και μοναδικός αλγόριθμος επίλυσής του.
6. Τα δεδομένα ενός προβλήματος είναι πάντοτε κάποιοι αριθμοί.
7. Ένα επιλύσιμο πρόβλημα είναι πάντοτε δομημένο.
8. Ένα άλυτο πρόβλημα είναι και αδόμητο.
9. Το ότι ο βαθμός πρόσβασης ενός μαθητή είναι 18,4 αποτελεί δεδομένο, ενώ είναι πληροφορία ότι ο μαθητής αυτός είναι αριστούχος.
10. Οι έννοιες δεδομένα και πληροφορία δεν είναι ταυτόσημες. Τα δεδομένα είναι ακατέργαστα γνωστικά στοιχεία, ενώ η πληροφορία είναι γνωστικό στοιχείο που προέρχεται από επεξεργασία.
11. Το πρόβλημα δίνεται ένας αριθμός και ζητείται να διαπιστωθεί αν είναι ατελής είναι πρόβλημα απόφασης.
12. Σύμφωνα με τη σχηματική παρουσίαση ενός προβλήματος, το αρχικό πρόβλημα αναπαρίσταται από μία έλλειψη.
13. Ένα επιλύσιμο πρόβλημα είναι πάντα δομημένο.
14. Ένα ανοικτό πρόβλημα μπορεί στο μέλλον να γίνει επιλύσιμο ή άλυτο ή να παραμείνει ανοικτό.
15. Δομημένα χαρακτηρίζονται εκείνα τα προβλήματα, των οποίων η επίλυση προέρχεται από μία αυτοματοποιημένη διαδικασία.
16. Η μεταφορά δεδομένων είναι μία από τις βασικές λειτουργίες που εκτελεί ο υπολογιστής.
17. Η έννοια του αλγορίθμου συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής.
18. Η καταγραφή της δομής ενός προβλήματος σημαίνει αυτόματα ότι έχει αρχίσει η διαδικασία ανάλυσης του προβλήματος σε άλλα απλούστερα.
19. Μία από τις πράξεις που μπορεί να εκτελέσει απευθείας ο υπολογιστής είναι ο πολλαπλασιασμός.
20. Η μετακίνηση από μία πόλη Α σε μία πόλη Β με ποδήλατο, όταν υπάρχει μόνο ένας δρόμος είναι ένα αδόμητο πρόβλημα.
21. Ο υπολογισμός της δευτεροβάθμιας εξίσωσης, αποτελεί πρόβλημα απόφασης.
22. Ένα δομημένο πρόβλημα είναι επιλύσιμο.
23. Ο υπολογισμός του εμβαδού ενός κύκλου αποτελεί πρόβλημα βελτιστοποίησης.
24. Η λογική πράξη Η είναι αληθής όταν έστω μία από τις δύο προτάσεις που συνοδεύει είναι αληθής.
25. Άλυτα χαρακτηρίζονται εκείνα τα προβλήματα για τα οποία έχουμε φτάσει στην παραδοχή ότι δεν επιδέχονται λύση.
26. Με τον όρο δεδομένο αναφέρεται οποιοδήποτε γνωστικό στοιχείο προέρχεται από επεξεργασία δεδομένων.
27. Ένα από τα στάδια αντιμετώπισης ενός προβλήματος είναι η ανάλυση.
28. Μια κατάσταση που χρήζει αντιμετώπισης αλλά έχει προφανή λύση δεν αποτελεί πρόβλημα.
29. Ένα επιλύσιμο πρόβλημα μπορεί να ανήκει και στην κατηγορία των προβλημάτων βελτιστοποίησης.
30. Τα δεδομένα ενός προβλήματος μπορούν να γίνουν αντιληπτά μόνο με την όραση ή την ακοή.
31. Τα δεδομένα ενός προβλήματος μπορούν να γίνουν αντιληπτά με μία από τις 5 ανθρώπινες αισθήσεις.
32. Η παραγοντοποίηση ενός θετικού ακεραίου αριθμού Ν, σε γινόμενο πρώτων παραγόντων, αποτελεί πρόβλημα βελτιστοποίησης.
33. Για την κατανόηση ενός προβλήματος, δεν αρκεί η σωστή διατύπωση από τον δημιουργό του.
34. Ένα πρόβλημα λέγεται άλυτο, όταν δεν έχουμε βρει λύση ακόμα.
35. Μια βασική λειτουργία του υπολογιστή είναι ο πολλαπλασιασμός αλλά ρωσικά
36. Τα στάδια αντιμετώπισης ενός προβλήματος είναι η διερεύνηση, η κατανόηση και η επίλυση.
37. Η λογική πράξη Η ονομάζεται και σύζευξη.
38. Αν είναι δεδομένο ότι ένα αυτοκίνητο τρέχει με 150 km/h, τότε η πληροφορία είναι ότι τρέχει γρήγορα.

1. Οι μεταβλητές αντιστοιχίζονται από τον μεταγλωττιστή κάθε γλώσσας σε συγκεκριμένες θέσεις μνήμης του υπολογιστή.
2. Μπορούμε να αρχικοποιήσουμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ.
3. Η τιμή της μεταβλητής είναι το περιεχόμενο της αντίστοιχης θέσης μνήμης και δεν μεταβάλλεται στη διάρκεια εκτέλεσης του προγράμματος.
4. Ο τύπος της μεταβλητής αλλάζει κατά την εκτέλεση του προγράμματος.
5. Η περατότητα ενός αλγορίθμου αναφέρεται στο γεγονός ότι καταλήγει στη λύση του προβλήματος μετά από πεπερασμένο αριθμό βημάτων.
6. Η εντολή Αλγόριθμος είναι δηλωτική εντολή ενώ η εντολή Διάβασε είναι εκτελεστέα εντολή.
7. Σε ένα διάγραμμα ροής η έλλειψη δηλώνει την είσοδο ή την έξοδο στοιχείων.
8. Ένας αλγόριθμος μπορεί να μην έχει έξοδο.
9. Η σύγκριση 'ΑΛΗΘΗΣ' > 'ΑΛΗΘΕΣ' δίνει τιμή ΨΕΥΔΗΣ.
10. Ο ακέραιος τύπος δεν έχει συγκεκριμένο εύρος τιμών.
11. Οι τελεστές div, * έχουν ίδια ιεραρχία.
12. Η δήλωση σταθερών είναι υποχρεωτική σε ένα πρόγραμμα.
13. Ένας αλγόριθμος είναι μία πεπερασμένη σειρά ενεργειών.
14. Οι ενέργειες που ορίζει ένας αλγόριθμος είναι αυστηρά καθορισμένες.
15. Η έννοια του αλγορίθμου συνδέεται αποκλειστικά με την Πληροφορική.
16. Ο αλγόριθμος τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης εντολών.
17. Ο πιο δομημένος τρόπος παρουσίασης αλγορίθμων είναι με ελεύθερο κείμενο.
18. Ένας αλγόριθμος στοχεύει στην επίλυση ενός προβλήματος.
19. Η διαίρεση με το 0 παραβιάζει το κριτήριο περατότητας ενός αλγορίθμου.
20. Η αποτελεσματικότητα είναι ένα από τα κριτήρια που πρέπει να ικανοποιεί ένας Αλγόριθμος.
21. Δεν υπάρχει αλγόριθμος για τη σχεδίαση αλγορίθμων.
22. Δεσμευμένες λέξεις καλούνται οι λέξεις που χρησιμοποιούνται από την ίδια τη ΓΛΩΣΣΑ για συγκεκριμένους λόγους και δεν μπορούν να χρησιμοποιηθούν ως ονόματα.
23. Ο τελεστής MOD χρησιμοποιείται για τον υπολογισμό του ηλίκου μίας διαίρεσης ακεραίων αριθμών.
24. Κάθε εντολή ενός αλγορίθμου πρέπει να καθορίζεται χωρίς αμφιβολία για τον τρόπο εκτέλεσής της.
25. Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μία έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες.
26. Ο πολλαπλασιασμός σε ένα πρόγραμμα γραμμένο σε ΓΛΩΣΣΑ έχει μεγαλύτερη προτεραιότητα σε σχέση με την ύψωση σε δύναμη.
27. Η έξοδος ενός αλγορίθμου μπορεί να είναι είσοδος σε άλλον αλγόριθμο.
28. Μια αριθμητική έκφραση δεν μπορεί να περιέχει μεταβλητές, σταθερές, τελεστές και παρενθέσεις.
29. Μια μεταβλητή μπορεί να πάρει τιμή μόνο με την εντολή ΔΙΑΒΑΣΕ.
30. Το αποτέλεσμα της παρακάτω έκφρασης $2^3 + 3 * (27 \bmod (25 \bmod 7))$ είναι 11.
31. Για την δήλωση μιας λογικής σταθεράς, μπορούμε να γράψουμε ΛΟΓΙΚΕΣ: flag = ΑΛΗΘΗΣ.
32. Σε μια μεταβλητή πραγματικού τύπου μπορούμε να εκχωρήσουμε και το αποτέλεσμα μιας ακέραιας αριθμητικής παράστασης.
33. Ένας από τους τρόπους αναπαράστασης αλγορίθμων είναι και η κωδικοποίηση.
34. Ο τελεστής + είναι συγκριτικός.
35. Το γεωμετρικό σχήμα του πλάγιου παραλληλογράμμου αναπαριστά την είσοδο δεδομένων ή την έξοδο των αποτελεσμάτων σε ένα λογικό διάγραμμα.
36. Το διάγραμμα ροής είναι ένας τρόπος αναπαράστασης αλγορίθμου που μπορεί να παραβιάσει το κριτήριο της καθοριστικότητας.
37. Κατά την εκτέλεση ενός προγράμματος η τιμή μιας μεταβλητής μπορεί να αλλάζει, ο τύπος της όμως μένει υποχρεωτικά αναλλοίωτος.
38. Μπορούμε να αρχικοποιήσουμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ.
39. Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρις ότου βρεθεί κάποια διαφορά.
40. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ.
41. Η σχέση (ΟΧΙ συνθήκη1 ΚΑΙ ΟΧΙ συνθήκη2) δίνει πάντα τα ίδια αποτελέσματα με τη σχέση ΟΧΙ (συνθήκη1 ΚΑΙ συνθήκη2).
42. Μια μεταβλητή που δεν έχει πάρει τιμή μέσω μιας εντολής εκχώρησης ή ΔΙΑΒΑΣΕ, έχει απροσδιόριστο περιεχόμενο.
43. Κάθε αλγόριθμος τελειώνει με τη δεσμευμένη λέξη Τέλος και το όνομα του αλγορίθμου.
44. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή ΔΙΑΒΑΣΕ.
45. Ο αλγόριθμος είναι ένα μέσο παραγωγής δεδομένων.
46. Το όνομα μιας συμβολικής σταθεράς πρέπει να ξεκινάει με ψηφίο.
47. Ο αλγόριθμος είναι ένα μέσο παραγωγής πληροφοριών.
48. Το όνομα μιας συμβολικής σταθεράς πρέπει να ξεκινάει με γράμμα.

49. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή εκχώρησης ακριβώς μία φορά.
50. Σε έναν αλγόριθμο στον οποίο υπάρχει μόνο η δομή ακολουθίας, κάθε εντολή εκτελείται.
51. Όταν μία μεταβλητή είναι αλφαριθμητικού τύπου, μπορούμε να της εκχωρήσουμε τον αριθμό τηλεφώνου κάποιου.
52. Η λέξη διάβαζε μπορεί να αποτελέσει το όνομα μιας μεταβλητής.
53. Αριστερά του τελεστή εκχώρησης (\leftarrow), μπορεί να υπάρξει μόνο μεταβλητή.
54. Σε ένα διάγραμμα ροής, η εντολή Διάβασε συμβολίζεται με ένα ορθογώνιο παραλληλόγραμμο.
55. Κάθε αλγόριθμος πρέπει να πληρεί το κριτήριο της πληρότητας.
56. Η ακολουθιακή δομή εντολών χρησιμοποιείται για την αντιμετώπιση προβλημάτων στα οποία είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών.
57. Για τον υπολογισμό μιας έκφρασης, όλες οι μεταβλητές που εμφανίζονται σ' αυτή πρέπει να έχουν οριστεί προηγουμένως, δηλαδή να έχουν κάποια τιμή.
58. Ενώ η τιμή μίας μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της.
59. Σε μία εντολή εκχώρησης του αποτελέσματος μίας έκφρασης σε μία μεταβλητή, η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου.
60. Η λογική πράξη ΚΑΙ μεταξύ δύο προτάσεων είναι ψευδής όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.
61. Η λογική έκφραση $X \text{ Η } (OXI \ X)$ είναι πάντα αληθής για κάθε τιμή της λογικής μεταβλητής X .
62. Στην αριθμητική έκφραση $A+B*Γ$ εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός.
63. Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή και ο τύπος μιας μεταβλητής.

Δομή Επιλογής (κεφ. 2 και 8 σχολ. Βιβλίου)

1. Η λογική πράξη "ή" μεταξύ δύο προτάσεων είναι ψευδής, όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.
2. Στη δομή επιλογής μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν.
3. Η δομή επιλογής επιτρέπει τη διενέργεια ελέγχων κατά τη διάρκεια εκτέλεσης των αλγορίθμων.
4. Αν η Μ είναι μια λογική μεταβλητή η εντολή εκχώρησης $M \leftarrow \text{'Αληθής'}$ είναι επιτρεπτή.
5. Αν έχουμε την εντολή $A \leftarrow (X < 5)$, τότε η μεταβλητή Α είναι λογικού τύπου.
6. Το διάγραμμα ροής ενός αλγορίθμου που έχει μία εντολή $AN \dots \Lambda\Lambda\Lambda\Omega\Sigma$ θα περιέχει δύο ρόμβους.
7. Η χρήση εμφωλευμένων AN είναι καλή προγραμματιστική τακτική.
8. Στην εντολή $\alpha \leftarrow \text{"Φλεβάρης"} > \text{"Μάρτης"}$ το α παίρνει την τιμή $\Lambda\Lambda\text{Η}\Theta\text{Η}\Sigma$.
9. Η λογική συνθήκη " α " < " β " είναι αληθής.
10. Η λογική παράσταση $x \text{ ΚΑΙ } \text{ΟΧΙ}(x)$ είναι πάντα ψευδής ανεξάρτητα από την τιμή του x .

Δομή Επανάληψης (κεφ. 2 και 8 σχολ. Βιβλίου)

1. Ο ρόμβος έχει 2 εξόδους.
2. Η Μέχρις_ότου προτιμάται στον έλεγχο εγκυρότητας.
3. Κάθε εντολή ΟΣΟ μπορεί να μετατραπεί σε μια ισοδύναμη εντολή ΓΙΑ.
4. Όταν το βήμα σε μια εντολή ΓΙΑ μειώνεται κατά ένα τότε αυτό μπορεί να παραληφθεί.
5. Αν η συνθήκη της ΟΣΟ είναι αληθής τότε ο βρόχος θα εκτελεστεί οπωσδήποτε πάνω από μια φορά.
6. Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του.
7. Η ολίσθηση ενός αριθμού προς τα αριστερά στο δυαδικό σύστημα, ισούται με τον υποδιπλασιασμό του.
8. Το βήμα σε μια εντολή επανάληψης Για πρέπει να είναι πάντα ακέραιος αριθμός.
9. Ο βρόχος Για x από 5 μέχρι 6 με_βήμα -1 εκτελείται μία μόνο φορά.
10. Η μορφή επανάληψης Για μπορεί να χρησιμοποιηθεί και όταν δεν γνωρίζουμε στην αρχή τον αριθμό των επαναλήψεων.
11. Ο βρόχος Για i από 0 μέχρι 3 με_βήμα 3 εκτελείται τρεις φορές.
12. Με χρήση της εντολής Όσο...επανάλαβε επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη.
13. Οι εντολές της επαναληπτικής Όσο θα εκτελεστούν τουλάχιστον μια φορά.
14. Όταν το πλήθος των επαναλήψεων είναι γνωστό, δεν μπορεί να χρησιμοποιηθεί η εντολή επανάληψης Όσο.
15. Η εντολή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ εκτελείται υποχρεωτικά τουλάχιστον μία φορά.
16. Σε ένα διάγραμμα ροής ο ρόμβος δηλώνει την αρχή και το Τέλος του αλγόριθμου.
17. Ο βρόχος Για K από 5 μέχρι 5 δεν εκτελείται καμία φορά.
18. Ο βρόχος Για K από 5 μέχρι 1 εκτελείται 5 φορές.
19. Στην επαναληπτική δομή Για ... από ... μέχρι ...με_βήμα οι τιμές από, μέχρι και με_βήμα δεν είναι απαραίτητο να είναι ακέραιες.
20. Η εντολή επανάληψης Για ... ΑΠΟ ... ΜΕΧΡΙ ...ΜΕ_ΒΗΜΑ μπορεί να χρησιμοποιηθεί, όταν έχουμε άγνωστο αριθμό επαναλήψεων.
21. Για k από - 4 μέχρι - 3 εκτελείται ακριβώς δύο φορές.
22. Η εντολή ΓΙΑ μπορεί να χρησιμοποιηθεί Για έλεγχο εγκυρότητας τιμής.
23. Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.
24. Η ΓΛΩΣΣΑ υποστηρίζει τρεις εντολές επανάληψης, την εντολή ΟΣΟ, την εντολή ΜΕΧΡΙΣ_ΟΤΟΥ και την εντολή ΓΙΑ.
25. Όταν ένας βρόχος είναι εμφωλευμένος σε άλλο, ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
26. Όταν το πλήθος των επαναλήψεων είναι γνωστό, δεν μπορεί να χρησιμοποιηθεί η εντολή επανάληψης Όσο ...Επανάλαβε.
27. Μία εντολή ΓΙΑ μπορεί να μετατραπεί σε ΟΣΟ ακόμα κι Αν το βήμα της είναι αρνητικό.
28. Η επανάληψη που δεν τελειώνει ποτέ παραβιάζει το κριτήριο της περατότητας.
29. Όταν μια ομάδα εντολών δεν ολοκληρώνεται μετά από ένα πεπερασμένο αριθμό βημάτων, αποκαλείται υπολογιστική διαδικασία.
30. Η εντολή Για i ΑΠΟ 0 ΜΕΧΡΙ 20 θα εκτελεστεί 20 φορές.

Δομές Δεδομένων-Πίνακες (κεφ. 3 & 9 σχολ. Βιβλίου)

1. Οι δείκτες ενός πίνακα μπορούν να είναι χαρακτηρισ.
2. Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής (φυσάλιδα) βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία.
3. Για την εισαγωγή ενός νέου στοιχείου στην ουρά ο δείκτης rear αυξάνεται κατά 1 και ο δείκτης front μειώνεται κατά 1.
4. Στη δομή FIFO απαιτείται η ύπαρξη δύο δεικτών
5. Ένας 3-διάστατος πίνακας είναι δυναμική δομή δεδομένων.
6. Το όνομα ενός πίνακα καθορίζει μια ομάδα διαδοχικών θέσεων στη μνήμη.
7. Ο πίνακας δεν δηλώνεται ως σταθερά.
8. Η παραβίαση ορίων πινάκα αναιρεί την καθοριστικότητα.
9. Το πρόβλημα ταξινόμησης πίνακα με τον πιο γρήγορο τρόπο είναι πρόβλημα βελτιστοποίησης.
10. Σκοπός της ταξινόμησης των στοιχείων ενός πίνακα είναι και να διευκολυνθεί στη συνέχεια η αναζήτηση.
11. Στην πράξη, είναι σύνηθες να χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή.
12. Οι πίνακες επεκτείνουν τις δυνατότητες του προγράμματος.
13. Η δυαδική αναζήτηση εφαρμόζεται σε ταξινομημένο πίνακα.
14. Η σειριακή αναζήτηση θα πρέπει να αποφεύγεται όταν ο πίνακας είναι μεγάλος.
15. Οι πίνακες είναι δυναμικές δομές δεδομένων.
16. Ένας πίνακας δεν έχει το ίδιο μέγεθος σε όλη τη διάρκεια εκτέλεσης του αλγόριθμου.
17. Τα δεδομένα που αποθηκεύονται σε έναν πίνακα δεν μπορούν να είναι διαφορετικού τύπου.
18. Για να προσπελάσουμε έναν πίνακα δεν μπορούμε να χρησιμοποιήσουμε τη δομή επανάληψης Όσο ... Επανάλαβε
19. Η σειριακή αναζήτηση είναι η πιο αποδοτική μέθοδος αναζήτησης.
20. Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα.
21. Η ουρά και η στοίβα μπορούν να υλοποιηθούν με δομή πίνακα.
22. Η εξαγωγή (dequeue) στοιχείου γίνεται από το εμπρός άκρο της ουράς.
23. Η απόθεση (pop) στοιχείου γίνεται από το πίσω άκρο της στοίβας.
24. Κατά τη διαδικασία της ώθησης πρέπει να ελέγχεται αν η στοίβα είναι γεμάτη.
25. Η ώθηση (push) στοιχείου είναι μία από τις λειτουργίες της ουράς.
26. Ο Αλγόριθμος της σειριακής αναζήτησης χρησιμοποιείται αποκλειστικά σε ταξινομημένους πίνακες.
27. Ένας πίνακας έχει σταθερό περιεχόμενο αλλά μεταβλητό μέγεθος.
28. Οι δυναμικές δομές δεδομένων αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης.
29. Η μέθοδος επεξεργασίας «πρώτο μέσα πρώτο έξω» (FIFO) εφαρμόζεται στη δομή δεδομένων ΟΥΡΑ.
30. Η ταξινόμηση είναι μια από τις βασικές λειτουργίες των δομών δεδομένων.
31. Η μέθοδος της σειριακής αναζήτησης δικαιολογείται στην περίπτωση που ο πίνακας είναι μη ταξινομημένος και μικρού μεγέθους.
32. Σε μια στατική δομή το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά την εκτέλεση του προγράμματος.
33. Όταν γίνεται σειριακή αναζήτηση κάποιου στοιχείου σε έναν μη ταξινομημένο πίνακα και το στοιχείο δεν υπάρχει στον πίνακα, τότε υποχρεωτικά σαρώνονται όλα τα στοιχεία του πίνακα.
34. Ένα από τα πλεονεκτήματα των πινάκων είναι ότι χρειάζονται λίγο χώρο στην μνήμη.
35. Οι πίνακες είναι μία δομή δεδομένων που στηρίζεται στην τεχνική δυναμικής παραχώρησης μνήμης.
36. Η χρήση πινάκων αυξάνει την απαιτούμενη μνήμη για την εκτέλεση του προγράμματος.
37. Οι δομές δεδομένων διακρίνονται σε τρεις μεγάλες κατηγορίες: τις στατικές, τις δυναμικές και τις ημιδομημένες.
38. Ο πίνακας που χρησιμοποιεί ένα μόνο δείκτη για την αναφορά των στοιχείων του ονομάζεται μονοδιάστατος.
39. Η ταξινόμηση φυσάλιδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος Αλγόριθμος ταξινόμησης.
40. Οι λειτουργίες ώθηση και απόθεση είναι οι κύριες λειτουργίες σε μια ουρά.
41. Η προσπέλαση είναι μια από τις βασικές πράξεις των δομών δεδομένων.
42. Με τη λειτουργία της συγχώνευσης, δύο ή περισσότερες δομές δεδομένων συνενώνονται σε μία ενιαία δομή.
43. Για να προσδιορίσουμε την θέση σε έναν διδιάστατο πίνακα χρειαζόμαστε μία μεταβλητή.
44. Όλες οι δομές δεδομένων μπορούν να υλοποιήσουν όλες τις βασικές πράξεις.
45. Σε κάθε πίνακα η δυαδική αναζήτηση είναι ο καταλληλότερος τρόπος αναζήτησης
46. Ο μικρότερος μονοδιάστατος πίνακας έχει 2 στοιχεία.
47. Ο μικρότερος διδιάστατος πίνακας έχει 4 στοιχεία.
48. Η εγγραφή δεν είναι απαραίτητο να έχει πεδία ίδιου τύπου.
49. Η εισαγωγή και η διαγραφή είναι δύο λειτουργίες που δεν γίνονται στους πίνακες.
50. Το πλήθος των εγγραφών ενός αρχείου μπορεί να αυξομειώνεται ανάλογα με τις ανάγκες που παρουσιάζονται.
51. Σε μια ουρά στην οποία ο δείκτης FRONT έχει την τιμή 5 και ο δείκτης REAR έχει την τιμή 7, μπορεί να γίνει εξαγωγή το πολύ 3 φορές αν στο μεταξύ δε γίνει καμία εισαγωγή.
52. Ένας πίνακας καταλαμβάνει μέρος της κύριας μνήμης του υπολογιστή.
53. Οι διδιάστατοι πίνακες δεν μπορούν να δοθούν ως παράμετροι σε μια διαδικασία.
54. Οι κόμβοι μιας στατικής δομής καταλαμβάνουν συνεχόμενες θέσεις στη μνήμη του υπολογιστή

55. Γνωρίζοντας ότι η μέγιστη τιμή ενός πίνακα A $N \times M$ βρίσκεται σε μία μοναδική θέση του πίνακα, ο εντοπισμός της θέσης αυτής μπορεί να υλοποιηθεί μέσω συνάρτησης.
56. Σε μία ουρά θα πρέπει πάντα να ισχύει $rear \geq front$, διαφορετικά θα σημαίνει ότι συνέβη υποχείλιση.
57. Η αναφορά $A[3, 2, 4, 5]$ αφορά στοιχείο τετραδιάστατου πίνακα.
58. Όλες οι πράξεις των δομών δεδομένων μπορούν να εφαρμοστούν και σε πίνακες.
59. Τα δεδομένα στους πίνακες χάνονται όταν τελειώσει το πρόγραμμα.
60. Η φυσαλίδα είναι η μόνη μέθοδος ταξινόμησης ενός μονοδιάστατου πίνακα.
61. Ένας λογικός πίνακας δεν μπορεί να ταξινομηθεί με φυσαλίδα
62. Αναζήτηση μιας τιμής μπορεί να γίνει μόνο αν ο πίνακας είναι ταξινομημένος

Εισαγωγή στον Προγραμματισμό (κεφ. 6 σχολ. Βιβλίου)

1. Ένα πρόγραμμα σε γλώσσα μηχανής είναι μια ακολουθία δυαδικών ψηφίων.
2. Ο μεταγλωττιστής δέχεται στην είσοδό του ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.
3. Το πηγαίο πρόγραμμα εκτελείται από τον υπολογιστή χωρίς μεταγλώττιση.
4. Ο διερμηνευτής διαβάζει μία προς μία τις εντολές του πηγαίου προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.
5. Ένα πρόγραμμα σε γλώσσα μηχανής χρειάζεται μεταγλώττιση.
6. Οι συμβολικές γλώσσες είναι ανεξάρτητες από την αρχιτεκτονική των υπολογιστών.
7. Η χρήση διερμηνευτή επιτρέπει την άμεση διόρθωση του προγράμματος
8. Σε μια αριθμητική έκφραση το αριθμός των αριστερών παρενθέσεων ισούται πάντα με τον δεξιών
9. Η χρήση των συμβολικών γλωσσών οδηγεί στη συγγραφή σύντομων προγραμμάτων που είναι εύκολο να συντηρηθούν
10. Η JAVA χρησιμοποιείται ιδιαίτερα για προγραμματισμό στο Διαδίκτυο (Internet).
11. Για την εκτέλεση μιας εντολής συμβολικής γλώσσας απαιτείται η μετάφρασή της σε γλώσσα μηχανής.
12. Ο ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα.
13. Τα συντακτικά λάθη οδηγούν σε λάθη που παράγονται κατά την εκτέλεση του προγράμματος.
14. Οι εντολές σε γλώσσα μηχανής εκτελούνται αμέσως από τον υπολογιστή.
15. Η FORTRAN αναπτύχθηκε ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων.
16. Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος είναι απαραίτητη στο δομημένο προγραμματισμό.
17. Τα συντακτικά λάθη στον πηγαίο κώδικα εμφανίζονται κατά το στάδιο της μεταγλώττισής του.
18. Η JAVA είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών που εκτελούνται σε καταναμημένα περιβάλλοντα, δηλαδή σε διαφορετικούς υπολογιστές οι οποίοι είναι συνδεδεμένοι στο διαδίκτυο.
19. Κατά τη μεταγλώττιση, το εκτελέσιμο πρόγραμμα παράγεται πριν το αντικείμενο.
20. Το αποτέλεσμα του συνδέτη-φορτωτή είναι το πηγαίο πρόγραμμα.
21. Η συγγραφή του πηγαίου προγράμματος και η διόρθωση των λαθών του γίνεται με την βοήθεια του συντάκτη.
22. Στη δομή ενός προγράμματος το τμήμα δήλωσης των σταθερών ακολουθεί το τμήμα δήλωσης των μεταβλητών.
23. Η Fortran είναι μία γλώσσα, που ειδικεύεται στον προγραμματισμό εφαρμογών Τεχνητής Νοημοσύνης.
24. Το αποτέλεσμα του συνδέτη-φορτωτή είναι το πηγαίο πρόγραμμα
25. Το τυπικό μιας γλώσσας είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή.
26. Οι γλώσσες μηχανής είναι στενά συνδεδεμένες με την αρχιτεκτονική του υπολογιστή.
27. Η C είναι μία γλώσσα, που ειδικεύεται στον προγραμματισμό συστημάτων.
28. Η γλώσσα ALGOL είναι κατάλληλη για προγραμματισμό σε καταναμημένα περιβάλλοντα.
29. Ο δομημένος προγραμματισμός επιτρέπει την ευκολότερη συντήρηση και διόρθωση ενός προγράμματος.
30. Ο δομημένος προγραμματισμός δεν περιλαμβάνει εντολές τύπου GOTO.
31. Η γλώσσα JAVA είναι 4ης γενιάς.
32. Ο προγραμματισμός με συμβολική γλώσσα χαρακτηρίζεται από μεταφερσιμότητα.
33. Η ανεξέλεγκτη χρήση της εντολής goto οδηγεί στη δημιουργία δυσνόητων προγραμμάτων.
34. Όταν ένας χρήστης διαπιστώσει την ύπαρξη ενός λογικού λάθους σε ένα εκτελέσιμο πρόγραμμα τότε έχει τη δυνατότητα να το διορθώσει μόνος του, ακόμα κι αν δε διαθέτει το αρχικό πηγαίο πρόγραμμα σε γλώσσα υψηλού επιπέδου.
35. Η έλλειψη της δυνατότητας σύνθετων εκφράσεων και εντολών στις γλώσσες χαμηλού επιπέδου, οδηγεί στη δημιουργία μακροσκελών προγραμμάτων.

Υποπρογράμματα (κεφ. 10 σχολ. Βιβλίου)

1. Σε μια συνάρτηση μπορούμε να χρησιμοποιήσουμε την εντολή ΓΡΑΨΕ, ενώ σε μια διαδικασία όχι.
2. Κάθε υποπρόγραμμα θα πρέπει να είναι ανεξάρτητο από άλλα υποπρογράμματα.
3. Μία συνάρτηση μπορεί να δεχτεί μία μόνο παράμετρο.
4. Ένα υποπρόγραμμα μπορεί κατά την εκτέλεσή του να καλέσει το κυρίως πρόγραμμα.
5. Τα υποπρογράμματα ενεργοποιούνται από μόνα τους
6. Στον κώδικα υλοποίησης μιας Συνάρτησης, το όνομά της πρέπει οπωσδήποτε να βρίσκεται τουλάχιστον μια φορά στο αριστερό τμήμα εντολής εκχώρησης.
7. Ένα υποπρόγραμμα δεν μπορεί να κληθεί περισσότερες από δυο φορές από το κυρίως πρόγραμμα.
8. Μια συνάρτηση υπολογίζει και επιστρέφει παραπάνω από μία τιμές με το όνομά της.
9. Ένα υποπρόγραμμα μπορεί να καλείται από ένα άλλο υποπρόγραμμα ή από το κύριο πρόγραμμα.
10. Ο τρόπος κλήσης των διαδικασιών και των συναρτήσεων είναι ίδιος, ενώ ο τρόπος σύνταξής τους είναι διαφορετικός.
11. Κατά την κλήση ενός υποπρογράμματος η πραγματική παράμετρος και η αντίστοιχη τυπική της είναι δυνατόν να έχουν το ίδιο όνομα
12. Όταν ένα υποπρόγραμμα καλείται από διαφορετικά σημεία του προγράμματος, οι πραγματικές παράμετροι πρέπει να είναι οι ίδιες.
13. Ο αριθμός των τυπικών παραμέτρων κατά την κλήση ενός υποπρογράμματος πρέπει να είναι ίδιος με τον αριθμό των πραγματικών παραμέτρων στην δήλωση ενός προγράμματος.
14. Η λίστα των πραγματικών παραμέτρων καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.
15. Μία συνάρτηση μπορεί να δεχτεί μία μόνο παράμετρο.
16. Η στοίβα χρόνου εκτέλεσης δημιουργείται από τον μεταγλωττιστή.
17. Κατά την εκτέλεση μιας εντολής εκχώρησης του κύριου προγράμματος σε ΓΛΩΣΣΑ, η στοίβα χρόνου εκτέλεσης είναι κενή.
18. Κατά την κλήση μιας διαδικασίας οι πραγματικές παράμετροι μπορούν να είναι μεταβλητές μόνο πραγματικού τύπου.
19. Μια συνάρτηση μπορεί να ενεργοποιηθεί μέσα από μια εντολή εκχώρησης ή μια εντολή ΓΡΑΨΕ.
20. Κάθε διαδικασία μπορεί να υλοποιηθεί και ως συνάρτηση.
21. Η παράμετρος είναι μια ιδιαίτερη περίπτωση μεταβλητής.
22. Μια συνάρτηση μπορεί να επιτελέσει οποιαδήποτε λειτουργία.
23. Όλα τα υποπρογράμματα ενεργοποιούνται με τον ίδιο τρόπο.
24. Το όνομα μιας διαδικασίας υπόκειται στους κανόνες ονοματολογίας των μεταβλητών και των συμβολικών σταθερών.
25. Πραγματικές ονομάζονται οι παράμετροι που χρησιμοποιούνται κατά την κλήση ενός υποπρογράμματος
26. Όταν σένα υποπρόγραμμα έχουμε παράμετρο πίνακα, τότε η τυπική και η αντίστοιχη πραγματική παράμετρος, πρέπει να είναι ίδιου τύπου και ίδιου μεγέθους
27. Κατά την κλήση ενός υποπρογράμματος η πραγματική παράμετρος και η αντίστοιχη τυπική της είναι δυνατόν να έχουν το ίδιο όνομα.
28. Μία συνάρτηση μπορεί να εμφανίσει στην οθόνη την τιμή που υπολόγισε.
29. Μία διαδικασία επιστρέφει στο πρόγραμμα που την έχει καλέσει τις τιμές όλων των παραμέτρων της.
30. Η συνάρτηση είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελέσει όλες τις λειτουργίες ενός προγράμματος.
31. Κατά την κλήση των υποπρογραμμάτων απαιτείται μία στοίβα, η λεγόμενη στοίβα χρόνου εκτέλεσης του προγράμματος.
32. Την ώρα που εκτελούνται οι εντολές ενός υποπρογράμματος, η στοίβα χρόνου εκτέλεσης είναι κενή.