

ΑΕΠΠ

Ερωτήσεις τύπου Ενωτά-Πάθος

Κεφάλαιο 1

1. Πρόβλημα είναι μια μαθηματική κατάσταση που πρέπει να αντιμετωπίσουμε
2. Αν υποβάλλουμε τα δεδομένα σε επεξεργασία παίρνουμε πληροφορίες
3. Ο υπολογιστής και το πρόβλημα είναι έννοιες που εξαρτώνται άμεσα η μια από την άλλη
4. Για την επίλυση ενός προβλήματος απαιτείται η σωστή διατύπωσή του
5. Ένα πρόβλημα μπορεί να αναλυθεί σε πολλά επιμέρους προβλήματα
6. Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων
7. Ο ταχύτερος μηχανισμός επεξεργασίας δεδομένων είναι ο υπολογιστής
8. Η κατανόηση ενός προβλήματος ακολουθεί την ανάλυσή του
9. Ο Η/Υ δεν μπορεί να επιτελέσει όλες τις λειτουργίες του ανθρώπινου εγκεφάλου
10. Η χρήση Η/Υ για την επίλυση προβλημάτων ενδείκνυται στις περιπτώσεις που χρειάζεται διαχείριση μεγάλου όγκου δεδομένων
11. Αν ένα πρόβλημα απαιτεί απλούς υπολογισμούς σε μικρό όγκο δεδομένων δεν μπορεί να ανατεθεί σε έναν Η/Υ
12. Ο Η/Υ μπορεί να επιλύσει με άνεση οποιοδήποτε πολύπλοκο πρόβλημα χωρίς τη βοήθεια του ανθρώπου
13. Για κάθε πρόβλημα υπάρχει και μοναδικός αλγόριθμος επίλυσής του
14. Με τη χρήση Η/Υ μπορούμε να επιλύσουμε οποιοδήποτε πρόβλημα
15. Για την επίλυση ενός προβλήματος πρέπει να έχουν καθοριστεί τα δεδομένα και τα ζητούμενα
16. Τα προβλήματα για τα οποία δεν μπορούμε να απαντήσουμε ακόμη, εάν είναι δυνατόν να επιλυθούν ονομάζονται μη επιλύσιμα
17. Η επίλυση της πρωτοβάθμιας εξίσωσης αποτελεί αδόμητο πρόβλημα
18. Άλυτα ονομάζουμε τα προβλήματα των οποίων η λύση δεν έχει βρεθεί
19. Δομή ενός προβλήματος είναι η εύρεση του συνόλου των μερών που το απαρτίζουν
20. Τα δεδομένα ενός προβλήματος είναι πάντοτε κάποιοι αριθμοί
21. Για την παραγωγή πληροφοριών απαιτούνται δεδομένα ή άλλες πληροφορίες
22. Ένα δομημένο πρόβλημα είναι πάντοτε επιλύσιμο
23. Ένα επιλύσιμο πρόβλημα είναι πάντοτε δομημένο
24. Με τη χρήση αλγορίθμων επιλύονται όλα τα προβλήματα
25. Ένα άλυτο πρόβλημα είναι και αδόμητο
26. Η πρόσθεση είναι μια από τις βασικές λειτουργίες που μπορεί να επιτελέσει έναν Η/Υ

27. Πληροφορία είναι το αποτέλεσμα από την επεξεργασία των δεδομένων
28. Πριν από την επίλυση ενός προβλήματος πρέπει αυτό να έχει διατυπωθεί με ακρίβεια και σαφήνεια
29. Με την επεξεργασία πληροφοριών μπορούν να εξαχθούν και άλλες πληροφορίες
30. Η κατανόηση ενός προβλήματος εξαρτάται μόνο από την διατύπωσή του
31. Στη δομή ενός προβλήματος περιλαμβάνονται τα συστατικά του μέρη
32. Το ότι το ύψος ενός ατόμου είναι 1,90 αποτελεί δεδομένο, ενώ είναι πληροφορία ότι το άτομο αυτό είναι ψηλό
33. Ανοικτά είναι τα προβλήματα που δεν είναι άλυτα ούτε επιλύσιμα
34. Η κακή διατύπωση ενός προβλήματος μπορεί να οδηγήσει στην μη επίλυσή του
35. Ο υπολογισμός του εμβαδού ενός τριγώνου είναι ανοικτό πρόβλημα
36. Ο υπολογισμός του εμβαδού ενός τριγώνου είναι δομημένο πρόβλημα
37. Με κριτήριο την δυνατότητα επίλυσης ενός προβλήματος οι κατηγορίες είναι: επιλύσιμα, υπολογιστικά και άλυτα

Κεφάλαιο 2

1. Ο αλγόριθμος είναι απαραίτητος μόνο για την επίλυση προβλημάτων πληροφορικής
2. Ο αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο εντολών
3. Αλγόριθμος είναι μια "συνταγή" που ορίζει τι πρέπει να γίνει ώστε να φτάσουμε στον επιθυμητό σκοπό
4. Ο αλγόριθμος μπορεί να περιλαμβάνει και εντολές που δεν είναι σαφείς
5. Η Πληροφορική μελετά τους αλγορίθμους μόνο από την σκοπιά των γλωσσών προγραμματισμού
6. Η αναπαράσταση των αλγορίθμων μπορεί να γίνει μόνο με χρήση ελεύθερου κειμένου και φυσικής γλώσσας
7. Τα σύμβολα των διαγραμμάτων ροής είναι η έλλειψη, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο
8. Η δομή της ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση πολύπλοκων προβλημάτων
9. Μια σταθερά μπορεί να αλλάξει τιμή κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
10. Μια μεταβλητή μπορεί να αλλάζει τιμή και όνομα κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
11. Μια μεταβλητή μπορεί να αλλάζει τύπο δεδομένων κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
12. Μια μεταβλητή μπορεί να αποθηκεύσει και αλφαριθμητικά δεδομένα
13. Δεξιά μιας εντολής εκχώρησης τιμής δεν μπορεί να βρίσκεται η ίδια μεταβλητή που βρίσκεται αριστερά
14. Όλα τα προβλήματα λύνονται και αλγοριθμικά
15. Τα σχόλια διευκολύνουν την κατανόηση ενός αλγορίθμου
16. Σε μια εντολή εκχώρησης δεν επιτρέπεται η χρήση σταθερών
17. Ένας αλγόριθμος επιλύει μόνο υπολογιστικά προβλήματα

18. Για την αναπαράσταση των δεδομένων εισόδου ενός αλγορίθμου χρησιμοποιούμε τις σταθερές
19. Η σειρά εκτέλεσης των εντολών στη δομή ακολουθίας είναι προκαθορισμένη
20. Στη δομή ακολουθίας εκτελούνται όλες οι εντολές
21. Στο δεξί τμήμα μιας εντολής εκχώρησης πρέπει να υπάρχει υποχρεωτικά πράξη
22. Τα είδη των μεταβλητών που χρησιμοποιούμε είναι οι αριθμητικές, οι αλφαριθμητικές και οι σταθερές
23. Για να αναπαραστήσουμε τα δεδομένα και τα αποτελέσματα σ' έναν αλγόριθμο, χρησιμοποιούμε σταθερές
24. Η είσοδος σε ένα αλγοριθμικό πρόβλημα είναι ένα σύνολο μεταβλητών που σχετίζονται με τα δεδομένα του
25. Στο διάγραμμα ροής το σχήμα του ρόμβου δηλώνει το τέλος ενός αλγορίθμου
26. Η εντολή εκχώρησης τιμής αποδίδει το αποτέλεσμα μιας έκφρασης (παράστασης) σε μια μεταβλητή
27. Σε μια εντολή εκχώρησης είναι δυνατόν μια παράσταση στο δεξιό μέλος να περιέχει τη μεταβλητή που βρίσκεται στο αριστερό
28. Το αποτέλεσμα μια πράξης μπορεί να εκχωρηθεί σε μια σταθερά
29. Το αποτέλεσμα μια πράξης μπορεί να εκχωρηθεί σε μια μεταβλητή
30. Δεσμευμένες λέξεις ονομάζονται αυτές που ορίζει ο προγραμματιστής ως ονομασίες των μεταβλητών που χρησιμοποιεί
31. Η εντολή $X \leftarrow X * X$ είναι έγκυρη
32. Στη δομή ακολουθίας μια συγκεκριμένη εντολή μπορεί να εκτελεστεί πολλές φορές
33. Χρησιμοποιούμε τη δομή επιλογής όταν θέλουμε μια ομάδα εντολών να εκτελεστεί πολλές φορές
34. Η σύζευξη δύο λογικών συνθηκών είναι ψευδής όταν μόνο μία από τις δύο λογικές συνθήκες είναι αληθής
35. Μια δομή επιλογής μπορεί να εκτελεστεί πολλές φορές
36. Η δομή της επιλογής χρησιμοποιείται στις περιπτώσεις όπου υπάρχει μία συγκεκριμένη σειρά βημάτων για την επίλυση ενός προβλήματος
37. Όταν χρειάζεται να υπάρξει απόφαση με βάση κάποιο κριτήριο, τότε χρησιμοποιείται η δομή της επιλογής
38. Η δομή της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής)
39. Οι διαδικασίες των πολλαπλών επιλογών εφαρμόζονται στα προβλήματα όπου εκτελούνται κάποιες εντολές ανάλογα με την τιμή που παίρνει μία μεταβλητή
40. Μία εντολή «Αν...τότε» δεν μπορεί να περιληφθεί στα όρια κάποιας άλλης εντολής "Αν...τότε"
41. Με την εντολή "**Av** $X \bmod 2 = 0$ " ελέγχουμε αν ο X είναι άρτιος
42. Η λογική πρόταση " $X^2 \geq 0$ " είναι πάντοτε αληθής
43. Στη δομή απλής επιλογής η ομάδα εντολών εντός της δομής εκτελείται όταν η συνθήκη είναι αληθής
44. Στην πολλαπλή επιλογή κάθε περίπτωση αντιστοιχεί σε διαφορετική τιμή της συνθήκης
45. Για τον υπολογισμό του μέσου όρου αριθμών πρέπει να χρησιμοποιηθεί η δομή επιλογής
46. Στη δομή επιλογής υπάρχει περίπτωση κάποιες εντολές να μην εκτελεστούν ποτέ
47. Κάθε εντολή Αν περιέχει Αλλιώς

48. Κάθε εντολή πολλαπλής επιλογής μπορεί να αναπαρασταθεί από πολλά απλά Αν
49. Όταν πρέπει να εκτελεστούν κάποιες εντολές υπό κάποια συνθήκη χρησιμοποιείται η δομή ακολουθίας
50. Μια δομή επιλογής μπορεί να περιλαμβάνει μόνο εντολές εκχώρησης τιμής
51. Σε μια έκφραση εκτελούνται πρώτα οι συγκριτικοί τελεστές και στη συνέχεια οι αριθμητικοί
52. Στη δομή επανάληψης Για δεν είναι δυνατόν η αρχική τιμή να είναι να είναι μεγαλύτερη από την τελική
53. Δεν μπορούμε να έχουμε μια δομή επανάληψης μέσα σε μια άλλη δομή επανάληψης
54. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό
55. Με χρήση της εντολής "Όσο...επανάλαβε" επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη
56. Με την εντολή "Αρχή_επανάληψης...Μέχρις_ότου..." υπάρχει ένας βρόχος που εκτελείται τουλάχιστον μία φορά
57. Η εντολή "Για i από .. μέχρι .. βήμα .." πρέπει να περιλαμβάνει για βήμα πάντοτε ένα θετικό αριθμό
58. Στη δομή επανάληψης Για το βήμα δεν μπορεί να είναι μηδέν
59. Όταν το πλήθος των επαναλήψεων είναι γνωστό δεν μπορεί να χρησιμοποιηθεί η δομή επανάληψης "Όσο...επανάλαβε"
60. Οι εντολές του βρόχου "Για" εκτελούνται τουλάχιστον μια φορά
61. Στην δομή «Όσο», η ομάδα εντολών εκτελείται μέχρι η συνθήκη να γίνει ψευδής
62. Στην δομή «Μέχρις_ότου», υπάρχει περίπτωση η ομάδα εντολών του βρόχου να μην εκτελεστεί καμία φορά
63. Στις δομές "Όσο" και "Μέχρις_ότου", οι συνθήκες είναι μεταξύ τους αντίθετες
64. Η δομή "Όσο...Επανάλαβε" χρησιμοποιείται μόνο όταν γνωρίζουμε το πλήθος των επαναλήψεων
65. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Για"
66. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Όσο"
67. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Μέχρις_ότου"
68. Η δομή "Μέχρις_ότου" τερματίζεται όταν η συνθήκη είναι αληθής
69. Κάθε βρόχος "Για" μπορεί να μετατραπεί σε "Όσο"
70. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Για"
71. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Μέχρις_ότου"
72. Εντός μιας δομής επιλογής δεν μπορεί να περιέχεται δομή επανάληψης
73. Εντός μιας δομής επανάληψης δεν μπορεί να περιέχεται δομή επιλογής
74. Στην δομή επανάληψης "Μέχρις_ότου" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να αρχικοποιούνται πριν το βρόχο
75. Στην δομή επανάληψης "Όσο" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να πάρουν τιμή πριν το βρόχο
76. Οι επαναληπτικές δομές χρησιμοποιούνται στην περίπτωση που μια ομάδα εντολών πρέπει να εκτελεστεί πολλές φορές

77. Μια δομή επανάληψης πρέπει να φροντίζει για μεταβολή της τιμής της συνθήκης ώστε κάποτε να τερματίζεται
78. Στη δομή επανάληψης Για πρέπει η τιμή του μετρητή να μεταβάλλεται εντός του βρόχου
79. Όταν σε μια δομή "Για" παραλείπεται το βήμα, τότε εννοείται ως βήμα το 1
80. Η δομή "Όσο" τερματίζεται όταν η συνθήκη γίνεται ψευδής
81. Εντός της δομής "Για" δεν επιτρέπεται η τροποποίηση της τιμής του μετρητή

Κεφάλαιο 3

1. Οι δομές δεδομένων διακρίνονται σε στατιστικές και δυναμικές
2. Κάθε δομή μπορεί να χρησιμοποιηθεί σε οποιοδήποτε πρόβλημα ή εφαρμογή
3. Δυναμικές είναι οι δομές που αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
4. Ένας πίνακας έχει σταθερό μέγεθος αλλά μεταβαλλόμενο περιεχόμενο
5. Ένας πίνακας μπορεί να υποθηκεύσει ακεραίους αριθμούς και ονόματα
6. Μία ουρά διατηρεί τα δεδομένα ταξινομημένα ως προς τη σειρά άφιξής τους
7. Η υλοποίηση της ουράς χρησιμοποιεί μία μόνο μεταβλητή-δείκτη για τη διαχείριση των εισαγωγών/διαγραφών, όπως και η περίπτωση της στοίβας
8. Όταν ψάχνουμε σε ένα τηλεφωνικό κατάλογο χρησιμοποιούμε τη σειριακή μέθοδο αναζήτησης
9. Η δυναμική παραχώρηση μνήμης είναι η τεχνική που χρησιμοποιείται στους πίνακες
10. Υπερχείλιση συμβαίνει όταν συμβεί απώθηση σε γεμάτη στοίβα
11. Υποχείλιση συμβαίνει σε μια ουρά όταν ζητήσουμε διαγραφή και ο δείκτης εμπρός είναι ίσος με τον δείκτη πίσω
12. Η ταξινόμηση είναι χρήσιμη διαδικασία γιατί έτσι εκτελείται γρηγορότερα η αναζήτηση
13. Δομή δεδομένων είναι ένα σύνολο δεδομένων που μπορούμε να εφαρμόσουμε μια σειρά λειτουργιών
14. Αλγόριθμοι + Δεδομένα = Προγράμματα
15. Η ουρά και η στοίβα είναι οι μόνες δομές δεδομένων στις οποίες εφαρμόζονται και οι 8 λειτουργίες
16. Η ταξινόμηση ευθείας ανταλλαγής είναι πολύ αποτελεσματική αν ο πίνακας έχει λίγα στοιχεία
17. Για να εφαρμοστεί η μέθοδος της σειριακής αναζήτησης είναι απαραίτητο τα στοιχεία να είναι ταξινομημένα
18. Τα στοιχεία ενός πίνακα είναι απαραίτητο να είναι όλα του ίδιου τύπου
19. Η σειριακή αναζήτηση μπορεί να οδηγήσει στην προσπέλαση ακόμη και ολόκληρου του πίνακα
20. Η ταξινόμηση έχει ως στόχο να διατάξει τα στοιχεία ενός μονοδιάστατου πίνακα με αύξουσα ή φθίνουσα διάταξη
21. Η σειριακή αναζήτηση χρησιμοποιείται κυρίως για μικρούς ή μη ταξινομημένους πίνακες
22. Στην υλοποίηση της στοίβας με τη χρήση πίνακα χρησιμοποιούνται 2 δείκτες για να δείχνουν την είσοδο και την έξοδο των δεδομένων
23. Στη στοίβα το στοιχείο που ωθείται τελευταίο απωθείται πρώτο
24. Η σειριακή αναζήτηση μπορεί να εκτελεστεί μόνο σε μη ταξινομημένους πίνακες

25. Στην ουρά το στοιχείο που εισάγεται πρώτο εξάγεται και πρώτο
26. Στη στοίβα το στοιχείο που εισάγεται τελευταίο εξάγεται και τελευταίο
27. Σε μια ουρά μπορούμε να προσθέσουμε στοιχεία στο μέσο της
28. Ο πίνακας είναι μια δυναμική δομή δεδομένων
29. Η ταξινόμηση της φουσαλίδας ταξινομεί τα στοιχεία ενός μονοδιάστατου πίνακα μόνο σε αύξουσα σειρά
30. Η θέση ενός στοιχείου σ' έναν δισδιάστατο πίνακα καθορίζεται από δυο αριθμούς
31. Οι διαστάσεις ενός πίνακα μπορούν να μεταβληθούν κατά την διάρκεια εκτέλεσης ενός αλγορίθμου
32. Η χρήση πινάκων έχει το μειονέκτημα της υπερβολικής χρήσης μνήμης
33. Η ταξινόμηση εφαρμόζεται και σε δισδιάστατους πίνακες
34. Στο ΠΙΝΑΚΑΣ[α, β] το α αντιστοιχεί στη γραμμή του πίνακα και το β στη στήλη
35. Προσπέλαση είναι η εύρεση ενός κόμβου με κάποιο κριτήριο
36. Για την υλοποίηση της ουράς χρησιμοποιούνται δυο δείκτες εμπρός και πίσω
37. Υποχείλιση συμβαίνει όταν εισαχθεί τιμή σε μια γεμάτη στοίβα
38. Για να προσπελάσουμε τα στοιχεία ενός πίνακα χρησιμοποιούμε επαναληπτική δομή
39. Για τον υπολογισμό μέσου όρου 120 αριθμών πρέπει να χρησιμοποιηθεί πίνακας
40. Στην ουρά όποιο στοιχείο μπαίνει πρώτο, βγαίνει τελευταίο
41. Δεν μπορεί να χρησιμοποιηθεί ο αλγόριθμος της φουσαλίδας σε πίνακα χαρακτήρων
42. Η ταξινόμηση της φουσαλίδας χρησιμοποιείται μόνο σε ταξινομημένους πίνακες
43. Οι δισδιάστατοι πίνακες μπορούν να θεωρηθούν ως μονοδιάστατοι πίνακες όπου κάθε θέση τους θεωρούνται άλλοι μονοδιάστατοι πίνακες
44. Οι διαστάσεις ενός πίνακα μπορούν να τροποποιηθούν αν χρειάζεται κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου

Κεφάλαιο 6

1. Ο προγραμματισμός αφορά την διατύπωση ενός αλγορίθμου σε κατανοητή από τον υπολογιστή μορφή
2. Βασικό στοιχείο του προγράμματος πέραν της κωδικοποίησης, είναι τα δεδομένα και οι δομές δεδομένων
3. Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία ανθρώπου - μηχανής
4. Ένα πρόγραμμα γραμμένο σε γλώσσα μηχανής είναι εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες
5. Ένα πρόγραμμα σε γλώσσα μηχανής χρειάζεται μετατροπή σε ακολουθία δυαδικών ψηφίων ώστε να εκτελεστεί από τον υπολογιστή
6. Μια ακολουθία εντολών 0, 1 είναι κατανοητή από τον επεξεργαστή και μπορεί να εκτελεστεί χωρίς μετατροπή
7. Ο προγραμματισμός σε γλώσσα μηχανής ήταν μια εξαιρετικά δύσκολη δουλειά που ελάχιστοι μπορούσαν να πραγματοποιήσουν
8. Ένα πρόγραμμα σε συμβολική γλώσσα ή γλώσσα χαμηλού επιπέδου τελικά μετατρέπεται σε γλώσσα μηχανής

9. Η συμβολική γλώσσα είναι μια ακολουθία 0 και 1
10. Μια εντολή ενός προγράμματος γλώσσας χαμηλού επιπέδου μεταφράζεται σε γλώσσα μηχανής
11. Με τις γλώσσες χαμηλού επιπέδου το κέντρο βάρους μετακινήθηκε από τον υπολογιστή προς τον άνθρωπο
12. Οι εντολές σε συμβολική γλώσσα αποτελούνται από συμβολικά ονόματα που αντιστοιχούν σε εντολές της γλώσσας μηχανής
13. Οι συμβολικές γλώσσες έφεραν την ανεξαρτησία από την αρχιτεκτονική κάθε υπολογιστή
14. Τα προγράμματα σε γλώσσες χαμηλού επιπέδου είναι εύκολο να γραφτούν και να συντηρηθούν
15. Τα προγράμματα σε γλώσσες υψηλού επιπέδου είναι ανεξάρτητα του υπολογιστή που αναπτύχθηκαν
16. Οι γλώσσες υψηλού επιπέδου ήρθαν να επιλύσουν τις αδυναμίες των συμβολικών γλωσσών για καλύτερη επικοινωνία ανθρώπου - μηχανής
17. Η Fortran είναι γλώσσα χαμηλού επιπέδου
18. Η γλώσσα προγραμματισμού Fortran είναι κατάλληλη για την επίλυση όλων των ειδικών προβλημάτων
19. Η Fortran είναι κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων
20. Ένα πρόγραμμα γλώσσας υψηλού επιπέδου μπορεί να εκτελεστεί σε οποιονδήποτε υπολογιστή ανεξάρτητα της αρχιτεκτονικής
21. Δεν έχει αναπτυχθεί γλώσσα υψηλού επιπέδου που να επιλύει όλα τα είδη προβλημάτων
22. Η COBOL είναι γλώσσα προσανατολισμένη στην ανάπτυξη εμπορικών εφαρμογών
23. Η COBOL δεν μπορεί να επιλύσει μαθηματικά προβλήματα
24. Η Algol είναι μια γλώσσα γενικού σκοπού αλλά με ελάχιστη πρακτική εφαρμογή
25. Η Algol αναπτύχθηκε για την επίλυση μαθηματικών προβλημάτων
26. Η PL/1 είναι γλώσσα υψηλού επιπέδου γενικής χρήσης με ελάχιστη επιτυχία
27. Η PL/1 προσπάθησε να συνθέσει Fortran και COBOL
28. Η Lisp και η Prolog είναι γλώσσες προγραμματισμού που χρησιμοποιούνται στον τομέα της Τεχνητής Νοημοσύνης
29. Η Basic είναι γλώσσα γενικού σκοπού με έμφαση στην εκπαίδευση αρχαρίων στον προγραμματισμό
30. Η Basic κατόρθωσε με συνεχείς ανανεώσεις να καταστεί από τις πλέον δημοφιλείς γλώσσες προγραμματισμού
31. Η γλώσσα Pascal είναι γλώσσα γενικού σκοπού και κατάλληλη τόσο για την εκπαίδευση όσο και για τη δημιουργία ισχυρών προγραμμάτων
32. Η Pascal αποτέλεσε τη βάση για την παρουσίαση γλωσσών όπως η ADA και η Modula-2
33. Η C χρησιμοποιήθηκε για την ανάπτυξη συστημάτων και έχει πολλές δυνατότητες χαμηλού επιπέδου
34. Η C++ είναι γλώσσα τέταρτης γενιάς
35. Η C++ και η Java είναι αντικειμενοστραφείς γλώσσες
36. Η Java είναι κατάλληλη για την ανάπτυξη εφαρμογών που εκτελούνται δικτυακά
37. Ο οδηγούμενος από τα γεγονότα προγραμματισμός και οι γλώσσες οπτικού προγραμματισμού αξιοποιούν τα γραφικά χαρακτηριστικά του υπολογιστή (ποντίκι, μενού κ.λ.π.)

38. Στον οπτικό προγραμματισμό ο προγραμματιστής δεν γράφει κώδικα αλλά σχεδιάζει τα οπτικά αντικείμενα της εφαρμογής του
39. Οι πιο διαδεδομένες οπτικές γλώσσες είναι η Visual Basic, η Visual C++ και η Java
40. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα
41. Χάρη στο δομημένο προγραμματισμό δημιουργούνται προγράμματα απλούστερα, κατανοήσιμα και εύκολα στη διόρθωση
42. Στο δομημένο προγραμματισμό ακολουθούνται οι αρχές του ιεραρχικού και του τμηματικού προγραμματισμού
43. Στην ιεραρχική σχεδίαση, η ανάλυση του αλγορίθμου πραγματοποιείται με την τεχνική «από πάνω προς τα κάτω»
44. Ένα πλεονέκτημα των γλωσσών προγραμματισμού χαμηλού επιπέδου είναι η μεταφερσιμότητα των προγραμμάτων
45. Παρά τη μεταφερσιμότητά τους, τα προγράμματα υψηλού επιπέδου είναι δυσκολότερο να διορθωθούν και να συντηρηθούν
46. Στα πλεονεκτήματα των γλωσσών προγραμματισμού υψηλού επιπέδου σε σχέση με τις συμβολικές γλώσσες είναι η ανεξαρτησία από τον τύπο του υπολογιστή
47. Οι γλώσσες υψηλού επιπέδου προτιμούνται για τον κοντύτερα στον άνθρωπο τρόπο έκφρασης
48. Είναι ευκολότερη η εκμάθηση μιας γλώσσας χαμηλού επιπέδου
49. Ο παράλληλος προγραμματισμός προϋποθέτει την ύπαρξη περισσότερων από έναν επεξεργαστών
50. Οι γλώσσες 4ης γενιάς χρησιμοποιούνται σε εφαρμογές που χρησιμοποιούν βάσεις δεδομένων
51. Η καλύτερη γλώσσα προγραμματισμού είναι η Pascal
52. Η επιλογή της καλύτερης γλώσσας προγραμματισμού εξαρτάται από το είδος της εφαρμογής
53. Μια γλώσσα προγραμματισμού προσδιορίζεται από: το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τη σημασιολογία της
54. Λεξιλόγιο μιας γλώσσας είναι όλες οι ακολουθίες που δημιουργούνται από τα στοιχεία του αλφαβήτου της γλώσσας, τις λέξεις
55. Η γραμματική είναι το συντακτικό μιας γλώσσας
56. Δεν υπάρχουν διαφορές μεταξύ φυσικών και τεχνητών γλωσσών
57. Η ιεραρχική σχεδίαση διασπά ένα πρόβλημα σε υποπρόβληματα και τα επιλύει, λύνοντας το αρχικό πρόβλημα
58. Ο τμηματικός προγραμματισμός υλοποιεί την φιλοσοφία της ιεραρχικής σχεδίασης
59. Κάθε υποπρόβλημα της ιεραρχικής σχεδίασης αποτελεί μια ανεξάρτητη ενότητα στον τμηματικό προγραμματισμό
60. Ο δομημένος προγραμματισμός περιέχει την ιεραρχική σχεδίαση και τον τμηματικό προγραμματισμό
61. Ο δομημένος προγραμματισμός προϋποθέτει τη χρήση εξειδικευμένων αλγοριθμικών δομών
62. Η αντικειμενοστραφής προσέγγιση εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τις τρεις αλγοριθμικές δομές: δομή ακολουθίας, δομή επιλογής και δομή επανάληψης
63. Για την ανάπτυξη προγραμμάτων με τη φιλοσοφία του παράλληλου προγραμματισμού απαιτείται η χρήση εξειδικευμένων γλωσσών προγραμματισμού

64. Ο διερμηνευτής (interpreter) μετατρέπει το εκτελέσιμο πρόγραμμα (executable) σε πρόγραμμα αντικείμενο (object)
65. Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα
66. Ο μεταγλωττιστής αναλαμβάνει το ρόλο της μετάφρασης από το πηγαίο πρόγραμμα στο εκτελέσιμο πρόγραμμα
67. Ο διερμηνευτής ελέγχει και μετατρέπει μια-μια τις εντολές του πηγαίου προγράμματος σε γλώσσα μηχανής
68. Ο συνδότης μετατρέπει το πηγαίο πρόγραμμα σε βιβλιοθήκη
69. Το αντικείμενο πρόγραμμα είναι ουσιαστικά γλώσσα μηχανής
70. Ο συνδότης είναι ένα πρόγραμμα ελέγχου των συντακτικών λαθών του πηγαίου προγράμματος
71. Τα λογικά λάθη εντοπίζονται από έναν μεταγλωττιστή αλλά όχι από έναν διερμηνευτή
72. Τα συντακτικά λάθη επιδιορθώνονται γρήγορα και εύκολα αλλά για τα λογικά λάθη η επιδιόρθωση είναι μια επίπονη διαδικασία
73. Ο μεταγλωττιστής έχει το μειονέκτημα ότι ελέγχει όλο το πρόγραμμα και πραγματοποιεί και την διαδικασία της σύνδεσης πολλές φορές μέχρι να επιδιορθωθούν όλα τα λάθη
74. Ο διερμηνευτής έχει το πλεονέκτημα ότι το πρόγραμμα εκτελείται γρηγορότερα
75. Τα σύγχρονα προγραμματιστικά περιβάλλοντα χρησιμοποιούν μικτές υλοποιήσεις διερμηνευτή και μεταγλωττιστή
76. Σ' ένα σύγχρονο προγραμματιστικό περιβάλλον απαιτείται η παρουσία συντάκτη για την συγγραφή του κειμένου των εντολών
77. Για την επιδιόρθωση των λογικών λαθών πολλές φορές ο προγραμματιστής καλείται να εκτελέσει το πρόγραμμά του επανειλημμένα
78. Ο μεταγλωττιστής σ' ένα σύγχρονο προγραμματιστικό περιβάλλον καθιστά την ύπαρξη του συνδότη προαιρετική
79. Ο μεταγλωττιστής διορθώνει όλα τα συντακτικά λάθη με τη χρήση βιβλιοθηκών
80. Η παράλειψη μιας εντολής Τέλος_αν είναι λογικό λάθος
81. Η χρήση της εντολής $MO \hat{Y} \ a + \beta + \gamma / 3$ αντί της $MO \hat{Y} \ (a + \beta + \gamma) / 3$ είναι λογικό λάθος
82. Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση
83. Ο εντοπισμός των συντακτικών λαθών σε ένα πρόγραμμα γίνεται από τον μεταγλωττιστή (compiler)
84. Κάθε προγραμματιστικό περιβάλλον διαθέτει τον μεταγλωττιστή του
85. Το εκτελέσιμο είναι το πρόγραμμα που εκτελείται από τον υπολογιστή
86. Στις γλώσσες υψηλού επιπέδου δεν υπάρχει η ικανότητα της μεταφερσιμότητας
87. Υπάρχουν δυο μεγάλες κατηγορίες μεταφραστικών προγραμμάτων: οι μεταφραστές και οι μεταγλωττιστές
88. Ο μεταγλωττιστής εξάγει το πηγαίο πρόγραμμα

Κεφάλαια 7-8-9

1. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ
2. Οι δηλώσεις των σταθερών προηγούνται πάντοτε των δηλώσεων των μεταβλητών
3. Τα σχόλια τοποθετούνται πάντα στην αρχή του προγράμματος

4. Κάθε εντολή AN περιλαμβάνει υποχρεωτικά το τμήμα ΑΛΛΙΩΣ
5. Κάθε τμήμα προγράμματος που χρησιμοποιεί την εντολή ΕΠΙΛΕΞΕ μπορεί να γραφεί και με εντολές AN
6. Η χρήση εμφωλευμένων AN είναι καλή προγραμματιστική τακτική
7. Αν το A έχει την τιμή 10 και το B την τιμή 20 τότε η έκφραση $(A > 8 \text{ ΚΑΙ } B < 20) \text{ Ή } (A > 10 \text{ Ή } B = 10)$ είναι αληθής
8. Οι εντολές που βρίσκονται σε μία επανάληψη ΓΙΑ εκτελούνται τουλάχιστο μία φορά
9. Κάθε επανάληψη μπορεί να γραφεί με την εντολή ΟΣΟ - ΕΠΑΝΑΛΑΒΕ
10. Σε περίπτωση εμφωλευμένων βρόχων, ο εσωτερικός πρέπει να περικλείεται ολόκληρος στον εξωτερικό
11. Η τιμή του βήματος αναφέρεται υποχρεωτικά σε κάθε εντολή ΓΙΑ
12. Τα ονόματα και το πλήθος των εισιτηρίων 10 θεάτρων μπορούν να αποθηκευτούν σε ένα δισδιάστατο πίνακα
13. Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του Η/Υ
14. Η ταξινόμηση των στοιχείων ενός πίνακα πρέπει να γίνεται πάντα πριν από την αναζήτηση
15. Δεσμευμένες λέξεις καλούνται οι λέξεις που έχουν δεσμεύσει για τα ονόματα των μεταβλητών
16. Όλοι οι πίνακες δηλώνονται στο τμήμα δήλωσης μεταβλητών του προγράμματος
17. Τα στοιχεία ενός πίνακα πρέπει να είναι του ίδιου τύπου
18. Η χρήση πινάκων αυξάνει την απαιτούμενη μνήμη για την εκτέλεση του προγράμματος
19. Ο δείκτης ενός μονοδιάστατου πίνακα πρέπει να είναι πάντα I
20. Το αλφάβητο της γλώσσας αποτελείται μόνο από γράμματα ελληνικά - λατινικά και αριθμούς
21. Σε μια εντολή εκχώρησης η μεταβλητή αριστερά και η έκφραση δεξιά του βέλους πρέπει να είναι του ίδιου τύπου
22. Η δομή ενός προγράμματος σε ΓΛΩΣΣΑ είναι αυστηρά καθορισμένη
23. Το τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ πρέπει απαραίτητως να προηγείται του τμήματος εντολών
24. Μια έκφραση μπορεί να περιέχει μεταβλητές, σταθερές, τελεστές και παρενθέσεις
25. Όταν μια δομή επανάληψης είναι εμφωλευμένη σε μια άλλη, τότε για κάθε εξωτερικό βρόχο πρέπει να ολοκληρώνονται όλες οι επαναλήψεις του εσωτερικού
26. Όταν μια δομή "Για" είναι εμφωλευμένη σε μια άλλη δομή "Για", τότε μπορούμε αν το επιθυμούμε για ευκολία να χρησιμοποιήσουμε την ίδια μεταβλητή για μετρητή
27. Ένα πρόγραμμα σε ΓΛΩΣΣΑ δεσμεύει τόσες συνεχόμενες θέσεις μνήμης για έναν πίνακα όσες και οι θέσεις που εμφανίζονται στο τμήμα δηλώσεων
28. Σε οποιοδήποτε σημείο ενός προγράμματος σε ΓΛΩΣΣΑ μπορούν να τοποθετηθούν σχόλια
29. Μπορούμε να αρχικοποιούμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ
30. Σε ένα πρόγραμμα σε ΓΛΩΣΣΑ δεν μπορούμε να χρησιμοποιήσουμε λογική σταθερά
31. Στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ δηλώνουμε τα ονόματα των πινάκων αλλά όχι και το μέγεθός τους
32. Ένα πρόγραμμα σε ΓΛΩΣΣΑ έχει τη δυνατότητα να τροποποιεί το μέγεθος ενός πίνακα στο τμήμα εντολών

- 33.** Για την εκτύπωση όλων των περιεχομένων ενός πίνακα απαιτείται η χρήση δομών επανάληψης
- 34.** Για την αναζήτηση σε ταξινομημένους πίνακες προτιμάται η δυαδική αναζήτηση
- 35.** Η συγχώνευση δυο πινάκων έχει ως στόχο να συνενώσει δυο πίνακες

Κεφάλαιο 10

- 1.** Μια διαδικασία και μια συνάρτηση μπορούν να εκτελούν ακριβώς τις ίδιες λειτουργίες
- 2.** Η ενεργοποίηση μιας συνάρτησης πραγματοποιείται με την εντολή ΚΑΛΕΣΕ
- 3.** Η κλήση των διαδικασιών γίνεται με απλή αναφορά του ονόματός τους
- 4.** Κάθε υποπρόγραμμα πρέπει να έχει μόνο μία είσοδο και μία έξοδο
- 5.** Μια διαδικασία μπορεί να καλέσει το κύριο πρόγραμμα
- 6.** Οι συναρτήσεις μπορούν να υπολογίζουν μόνο μια τιμή και να την επιστρέφουν
- 7.** Μια διαδικασία μπορεί να καλέσει μια συνάρτηση
- 8.** Ο τμηματικός προγραμματισμός έχει ως αποτέλεσμα την ταχύτερη εκτέλεση του προγράμματος
- 9.** Το κυρίως πρόγραμμα πρέπει να είναι πολύ μεγαλύτερο από τα υποπρογράμματα
- 10.** Η διαδικασίες έχουν περιορισμένες λειτουργίες σε σχέση με τις συναρτήσεις