

## Α. Σωστό - Λάθος

## I. Κεφάλαια 2-7-8

1. Ο αλγόριθμος είναι απαραίτητος μόνο για την επίλυση προβλημάτων πληροφορικής
2. Ο αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο εντολών
3. Ένας αλγόριθμος στοχεύει στην επίλυση ενός προβλήματος.
4. Ο αλγόριθμος είναι μια "συνταγή" που ορίζει τι πρέπει να γίνει ώστε να φτάσουμε στον επιθυμητό σκοπό.
5. Ο αλγόριθμος μπορεί να περιλαμβάνει και εντολές που δεν είναι σαφείς
6. Το πλάγιο παραλληλόγραμμο χρησιμοποιείται για αρχή και τέλος αλγορίθμου.
7. Ένας αλγόριθμος στοχεύει στην επίλυση ενός προβλήματος.
8. Η αναπαράσταση των αλγορίθμων μπορεί να γίνει μόνο με χρήση ελεύθερου κειμένου και φυσικής γλώσσας
9. Τα σύμβολα των διαγραμμάτων ροής είναι η έλλειψη, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο
10. Η δομή της ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση πολύπλοκων προβλημάτων
11. Μια σταθερά μπορεί να αλλάξει τιμή κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
12. Μια μεταβλητή μπορεί να αλλάζει τιμή και όνομα κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
13. Μια μεταβλητή μπορεί να αλλάζει τύπο δεδομένων κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
14. Μια μεταβλητή μπορεί να αποθηκεύσει και αλφαριθμητικά δεδομένα
15. Δεξιά μιας εντολής εκχώρησης τιμής δεν μπορεί να βρίσκεται η ίδια μεταβλητή που βρίσκεται αριστερά
16. Όλα τα προβλήματα λύνονται και αλγοριθμικά
17. Τα σχόλια διευκολύνουν την κατανόηση ενός αλγορίθμου
18. Σε μια εντολή εκχώρησης δεν επιτρέπεται η χρήση σταθερών
19. Ένας αλγόριθμος επιλύει μόνο υπολογιστικά προβλήματα
20. Για την αναπαράσταση των δεδομένων εισόδου ενός αλγορίθμου χρησιμοποιούμε τις σταθερές.
21. Η σειρά εκτέλεσης των εντολών στη δομή ακολουθίας είναι προκαθορισμένη
22. Στη δομή ακολουθίας εκτελούνται όλες οι εντολές
23. Στο δεξί τμήμα μιας εντολής εκχώρησης πρέπει να υπάρχει υποχρεωτικά πράξη
24. Τα είδη των μεταβλητών που χρησιμοποιούμε είναι οι αριθμητικές, οι αλφαριθμητικές και οι σταθερές
25. Για να αναπαραστήσουμε τα δεδομένα και τα αποτελέσματα σ' έναν αλγόριθμο, χρησιμοποιούμε σταθερές
26. Η είσοδος σε ένα αλγοριθμικό πρόβλημα είναι ένα σύνολο μεταβλητών που σχετίζονται με τα δεδομένα του
27. Στο διάγραμμα ροής το σχήμα του ρόμβου δηλώνει το τέλος ενός αλγορίθμου
28. Η εντολή εκχώρησης τιμής αποδίδει το αποτέλεσμα μιας έκφρασης (παράστασης) σε μια μεταβλητή
29. Σε μια εντολή εκχώρησης είναι δυνατόν μια παράσταση στο δεξιό μέλος να περιέχει τη μεταβλητή που βρίσκεται στο αριστερό
30. Το αποτέλεσμα μια πράξης μπορεί να εκχωρηθεί σε μια σταθερά
31. Το αποτέλεσμα μια πράξης μπορεί να εκχωρηθεί σε μια μεταβλητή

32. Δεσμευμένες λέξεις ονομάζονται αυτές που ορίζει ο προγραμματιστής ως ονομασίες των μεταβλητών που χρησιμοποιεί
33. Η εντολή  $X \leftarrow X * X$  είναι έγκυρη
34. Στη δομή ακολουθίας μια συγκεκριμένη εντολή μπορεί να εκτελεστεί πολλές φορές
35. Χρησιμοποιούμε τη δομή επιλογής όταν θέλουμε μια ομάδα εντολών να εκτελεστεί πολλές φορές
36. Η σύζευξη δύο λογικών συνθηκών είναι ψευδής όταν μόνο μία από τις δύο λογικές συνθήκες είναι αληθής.
37. Μια δομή επιλογής μπορεί να εκτελεστεί πολλές φορές.
38. Η δομή της επιλογής χρησιμοποιείται στις περιπτώσεις όπου υπάρχει μία συγκεκριμένη σειρά βημάτων για την επίλυση ενός προβλήματος.
39. Όταν χρειάζεται να υπάρξει απόφαση με βάση κάποιο κριτήριο, τότε χρησιμοποιείται η δομή της επιλογής
40. Η δομή της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής)
41. Οι διαδικασίες των πολλαπλών επιλογών εφαρμόζονται στα προβλήματα όπου εκτελούνται κάποιες εντολές ανάλογα με την τιμή που παίρνει μία μεταβλητή
42. Μία εντολή «Αν...τότε» δεν μπορεί να περιληφθεί στα όρια κάποιας άλλης εντολής "Αν...τότε"
43. Με την εντολή "Αν  $X \text{ div } 2 = 0$ " ελέγχουμε αν ο  $X$  είναι άρτιος
44. Η λογική πρόταση " $X^2 \geq 0$ " είναι πάντοτε αληθής
45. Στη δομή απλής επιλογής η ομάδα εντολών εντός της δομής εκτελείται όταν η συνθήκη είναι αληθής
46. Στην πολλαπλή επιλογή κάθε περίπτωση αντιστοιχεί σε διαφορετική τιμή της συνθήκης
47. Για τον υπολογισμό του μέσου όρου αριθμών πρέπει να χρησιμοποιηθεί η δομή επιλογής
48. Στη δομή επιλογής υπάρχει περίπτωση κάποιες εντολές να μην εκτελεστούν ποτέ
49. Κάθε εντολή Αν περιέχει Αλλιώς
50. Κάθε εντολή πολλαπλής επιλογής μπορεί να αναπαρασταθεί από πολλά απλά Αν
51. Όταν πρέπει να εκτελεστούν κάποιες εντολές υπό κάποια συνθήκη χρησιμοποιείται η δομή ακολουθίας
52. Μια δομή επιλογής μπορεί να περιλαμβάνει μόνο εντολές εκχώρησης τιμής
53. Σε μια έκφραση εκτελούνται πρώτα οι συγκριτικοί τελεστές και στη συνέχεια οι αριθμητικοί
54. Στη δομή επανάληψης Για δεν είναι δυνατόν η αρχική τιμή να είναι να είναι μεγαλύτερη από την τελική
55. Δεν μπορούμε να έχουμε μια δομή επανάληψης μέσα σε μια άλλη δομή επανάληψης
56. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό
57. Με χρήση της εντολής "Όσο...επανάλαβε" επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη
58. Με την εντολή "Αρχή\_επανάληψης...Μέχρις\_ότου..." υπάρχει ένας βρόχος που εκτελείται τουλάχιστον μία φορά
59. Η εντολή "Για  $i$  από .. μέχρι .. βήμα .." πρέπει να περιλαμβάνει για βήμα πάντοτε ένα θετικό αριθμό

60. Στη δομή επανάληψης Για το βήμα δεν μπορεί να είναι μηδέν
61. Όταν το πλήθος των επαναλήψεων είναι γνωστό δεν μπορεί να χρησιμοποιηθεί η δομή επανάληψης "Όσο...επανάλαβε"
62. Οι εντολές του βρόχου "Για" εκτελούνται τουλάχιστον μια φορά
63. Στην δομή «Όσο», η ομάδα εντολών εκτελείται μέχρι η συνθήκη να γίνει ψευδής
64. Στην δομή «Μέχρις\_ότου», υπάρχει περίπτωση η ομάδα εντολών του βρόχου να μην εκτελεστεί καμία φορά
65. Στις δομές "Όσο" και "Μέχρις\_ότου", οι συνθήκες είναι μεταξύ τους αντίθετες
66. Η δομή "Όσο...Επανάλαβε" χρησιμοποιείται μόνο όταν γνωρίζουμε το πλήθος των επαναλήψεων .
67. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Για" .
68. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Όσο" .
69. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Μέχρις\_ότου" .
70. Η δομή "Μέχρις\_ότου" τερματίζεται όταν η συνθήκη είναι αληθής.
71. Κάθε βρόχος "Για" μπορεί να μετατραπεί σε "Όσο" .
72. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Για".
73. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Μέχρις\_ότου" .
74. Εντός μιας δομής επιλογής δεν μπορεί να περιέχεται δομή επανάληψης.
75. Εντός μιας δομής επανάληψης δεν μπορεί να περιέχεται δομή επιλογής.
76. Στην δομή επανάληψης "Μέχρις\_ότου" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να αρχικοποιούνται πριν το βρόχο.
77. Στην δομή επανάληψης "Όσο" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να πάρουν τιμή πριν το βρόχο.
78. Οι επαναληπτικές δομές χρησιμοποιούνται στην περίπτωση που μια ομάδα εντολών πρέπει να εκτελεστεί πολλές φορές.
79. Μια δομή επανάληψης πρέπει να φροντίζει για μεταβολή της τιμής της συνθήκης ώστε κάποτε να τερματίζεται.
80. Στη δομή επανάληψης Για πρέπει η τιμή του μετρητή να μεταβάλλεται εντός του βρόχου.
81. Όταν σε μια δομή "Για" παραλείπεται το βήμα, τότε εννοείται ως βήμα το 1.
82. Η δομή "Όσο" τερματίζεται όταν η συνθήκη γίνεται ψευδής.
83. Εντός της δομής "Για" δεν επιτρέπεται η τροποποίηση της τιμής του μετρητή .
84. Η δομή της επιλογής χρησιμοποιείται στις περιπτώσεις όπου υπάρχει μια συγκεκριμένη σειρά βημάτων για την επίλυση ενός προβλήματος .
85. Η εντολή " Για i από ... μέχρι ... βήμα ..." πρέπει να περιλαμβάνει για βήμα ένα ακέραιο αριθμό
86. Οι επαναληπτικές διαδικασίες εφαρμόζονται όταν μια ακολουθία εντολών πρέπει να εφαρμοστεί σε δύο περιπτώσεις με βάση κάποια συνθήκη

87. Ο πολλαπλασιασμός αλά Ρωσικά βασίζεται σε δύο απλές για τον Η/Υ πράξεις, την πρόσθεση και την αφαίρεση
88. Όταν γνωρίζουμε το πλήθος των επαναλήψεων δεν μπορούμε να χρησιμοποιήσουμε την επαναληπτική δομή Όσο...επανάλαβε.
89. Οποιαδήποτε εντολή επανάληψης «Όσο...» μπορεί να γραφεί με την χρήση των άλλων δύο εντολών επανάληψης.
90. Η πρόταση Π ή (όχι Π) είναι πάντα αληθής, όπου Π λογική μεταβλητή.
91. Αν  $a=5$  και  $b=8$  η έκφραση  $a^2 \bmod 2 > b$  είναι αληθής, όπου  $a$  και  $b$  ακέραιες μεταβλητές.
92. Όταν το βήμα είναι 0 στην «Για ... από ... μέχρι ... με\_βήμα ..» παραβιάζεται το κριτήριο της καθοριστικότητας.
93. Ο τύπος μιας μεταβλητής μεταβάλλεται κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
94. Η ομάδα εντολών μέσα στην «Αρχή\_επανάληψης ..μέχρις\_ότου» εκτελείται τουλάχιστον μία φορά.
95. Ολίσθηση αριστερά στον «πολλαπλασιασμό αλά ρωσικά» σημαίνει ακέραια διαίρεση με το 2.
96. Δύο είναι οι κύριες λειτουργίες που εκτελούνται σε μια ουρά: η εισαγωγή και η διαγραφή
97. Κάθε αλγόριθμος πρέπει να πληρεί το κριτήριο της πληρότητας.
98. Αλγόριθμος ο οποίος θα εμφανίζει τα αποτελέσματά του στην οθόνη δεν πληρεί το κριτήριο της εξόδου.
99. Σε ένα διάγραμμα ροής, η εντολή διάβασε συμβολίζεται με ένα ορθογώνιο παραλληλόγραμμο.
100. Αριστερά του τελεστή εκχώρησης ( $\leftarrow$ ), μπορεί να υπάρξει μόνο μεταβλητή.
101. Η λέξη διάβαζε μπορεί να αποτελέσει το όνομα μιας μεταβλητής.
102. Ο πολλαπλασιασμός σε ένα πρόγραμμα γραμμένο σε ΓΛΩΣΣΑ έχει μεγαλύτερη προτεραιότητα σε σχέση με την ύψωση σε δύναμη.
103. Το διάγραμμα ροής ενός αλγορίθμου που έχει μία εντολή  $AN...ΑΛΛΙΩΣ$  θα περιέχει δύο ρόμβους.
104. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ
105. Αν είναι δεδομένο ότι ένα αυτοκίνητο τρέχει με 180km/h τότε η πληροφορία είναι ότι τρέχει γρήγορα.
106. Οι δηλώσεις των σταθερών προηγούνται πάντοτε των δηλώσεων των μεταβλητών
107. Τα σχόλια τοποθετούνται πάντα στην αρχή του προγράμματος
108. Κάθε εντολή  $AN$  περιλαμβάνει υποχρεωτικά το τμήμα  $ΑΛΛΙΩΣ$
109. Κάθε τμήμα προγράμματος που χρησιμοποιεί την εντολή  $ΕΠΙΛΕΞΕ$  μπορεί να γραφεί και με εντολές  $AN$
110. Η χρήση εμφωλευμένων  $AN$  είναι καλή προγραμματιστική τακτική
111. Όλες οι δομές επιλογής κλείνουν με την εντολή τέλος\_αν.
112. Στη δομή επιλογής οι εντολές που βρίσκονται μεταξύ του 'τότε' και του 'αλλιώς' μπορεί να μην εκτελεστούν.
113. 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
114. Οι εντολές που βρίσκονται σε μία επανάληψη  $ΓΙΑ$  εκτελούνται τουλάχιστο μία φορά
115. Κάθε επανάληψη μπορεί να γραφεί με την εντολή  $ΟΣΟ - ΕΠΑΝΑΛΑΒΕ$
116. Σε περίπτωση εμφωλευμένων βρόχων, ο εσωτερικός πρέπει να περικλείεται ολόκληρος στον εξωτερικό

117. Η τιμή του βήματος αναφέρεται υποχρεωτικά σε κάθε εντολή ΓΙΑ
118. Τα ονόματα και το πλήθος των εισιτηρίων 10 θεάτρων μπορούν να αποθηκευτούν σε ένα διδιάστατο πίνακα
119. Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του Η/Υ
120. Δεσμευμένες λέξεις καλούνται οι λέξεις που έχουν δεσμεύσει για τα ονόματα των μεταβλητών
121. Το αλφάβητο της γλώσσας αποτελείται μόνο από γράμματα ελληνικά - λατινικά και αριθμούς
122. Σε μια εντολή εκχώρησης η μεταβλητή αριστερά και η έκφραση δεξιά του βέλους πρέπει να είναι του ίδιου τύπου
123. Η δομή ενός προγράμματος σε ΓΛΩΣΣΑ είναι αυστηρά καθορισμένη
124. Το τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ πρέπει απαραίτητα να προηγείται του τμήματος εντολών
125. Μια έκφραση μπορεί να περιέχει μεταβλητές, σταθερές, τελεστές και παρενθέσεις
126. Όταν μια δομή επανάληψης είναι εμφωλευμένη σε μια άλλη, τότε για κάθε εξωτερικό βρόχο πρέπει να ολοκληρώνονται όλες οι επαναλήψεις του εσωτερικού
127. 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
128. Μπορούμε να αρχικοποιούμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ
129. Σε ένα πρόγραμμα σε ΓΛΩΣΣΑ δεν μπορούμε να χρησιμοποιήσουμε λογική σταθερά
130. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση των παρενθέσεων.
131. Η λογική έκφραση "ΜΕΓΑΛΟΣ" > "ΜΙΚΡΟΣ" είναι αληθής.
132. Αφού ισχύει η συνθήκη:  $(A + B) / 2 = A / 2 + B / 2$ , τότε ισχύει και η συνθήκη:  $(A + B) \text{ DIV } 2 = A \text{ DIV } 2 + B \text{ DIV } 2$ .
133. Μόνο οι υπολογιστές μπορούν να επιλύσουν προβλήματα με πολύπλοκες πράξεις.
134. Οι τελεστές div, mod δε μπορούν να χρησιμοποιηθούν σε πραγματικούς αριθμούς.
135. Μια λογική έκφραση περιέχει υποχρεωτικά συγκριτικό τελεστή.
136. Μια λογική έκφραση περιέχει υποχρεωτικά λογικό τελεστή.
137. Μια αλφαριθμητική μεταβλητή δε μπορεί να λάβει τις τιμές "Αληθής" ή "Ψευδής" καθώς σε αυτήν την περίπτωση θα έπρεπε να αποτελεί λογική μεταβλητή.
138. Η εντολή «Για I από 1 μέχρι 1 με βήμα -1» θα κάνει ακριβώς μία επανάληψη
139. Η εντολή «Για I από 1 μέχρι -1 με βήμα 1» θα κάνει ακριβώς μία επανάληψη
140. Η εντολή «Για I από -1 μέχρι 1 με βήμα 1» θα κάνει ακριβώς μία επανάληψη
141. Η εντολή «Για I από 1 μέχρι 1 με βήμα 0» θα κάνει ακριβώς μία επανάληψη
142. Η εντολή «Για I από 1 μέχρι 0 με βήμα -1» θα κάνει ακριβώς μία επανάληψη
143. Η εντολή «Για I από 0 μέχρι 1 με βήμα -1» θα κάνει ακριβώς μία επανάληψη
144. Η ΓΛΩΣΣΑ υποστηρίζει τρεις τύπους δεδομένων: τους ακέραιους, πραγματικούς και χαρακτήρες.
145. Η λογική πράξη Η είναι αληθής όταν έστω μία από τις δύο προτάσεις που συνοδεύει είναι αληθής.
146. Η λογική συνθήκη "α" < "β" είναι αληθής.

147. Η λογική παράσταση  $x$  ΚΑΙ ΟΧΙ  $x$  είναι πάντα ψευδής ανεξάρτητα από την τιμή του  $x$ .
148. Μία εντολή ΑΝ μπορεί προαιρετικά να συνοδεύεται από μία εντολή ΑΛΛΙΩΣ.
149. Κάθε εντολή εμφωλευμένων ΑΝ μπορεί να μετατραπεί σε ισοδύναμες ΑΝ...ΑΛΛΙΩΣ\_ΑΝ...ΑΛΛΙΩΣ.
150. Η εντολή ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ\_ΟΤΟΥ εκτελείται μέχρι η συνθήκη να γίνει αληθής.
151. Μία εντολή ΓΙΑ μπορεί να μετατραπεί σε ΟΣΟ ακόμα κι αν το βήμα της είναι αρνητικό.
152. Κάθε εντολή ΟΣΟ μπορεί να μετατραπεί σε μια ισοδύναμη εντολή ΓΙΑ.
153. Η εντολή ΓΙΑ μπορεί να χρησιμοποιηθεί για έλεγχο εγκυρότητας τιμής.
154. Όταν το βήμα σε μια εντολή ΓΙΑ μειώνεται κατά ένα τότε αυτό μπορεί να παραληφθεί.
155. Οι συναρτήσεις ΗΜ(), ΣΥΝ() και ΕΦ() δέχονται παράμετρο σε ακτίνια (rad).
156. Η ιεραρχία των λογικών πράξεων είναι: 1. όχι , 2. και 3. ή.
157. Μια υπολογιστική διαδικασία που δεν τελειώνει μετά από συγκεκριμένο αριθμό βημάτων αποτελεί αλγόριθμο.
158. Η  $A\_M(X)$  είναι η συνάρτηση της ΓΛΩΣΣΑΣ που υπολογίζει την απόλυτη τιμή του  $X$ .
159. Στην αριθμητική έκφραση  $A+B*Γ$  εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός.
160. Ο πιο αδόμετος τρόπος αναπαράστασης ενός αλγορίθμου είναι η κωδικοποίηση.
161. Στη δομή ακολουθίας όλες οι εντολές εκτελούνται υποχρεωτικά.
162. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή ΔΙΑΒΑΣΕ.
163. Σε μία εντολή εκχώρησης δεν μπορεί να υπάρχει αναφορά σε περισσότερες από μία συναρτήσεις.
164. Η λογική πράξη  $\text{Ή}$  μεταξύ δύο προτάσεων είναι ψευδής όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.
165. Η σύγκριση λογικών δεδομένων έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>).
166. Σε μια λογική έκφραση, οι συγκριτικοί τελεστές έχουν χαμηλότερη ιεραρχία από τους λογικούς τελεστές.
167. Η σύγκριση  $\text{'ΑΛΗΘΗΣ'}$  >  $\text{'ΑΛΗΘΕΣ'}$  δίνει τιμή ΨΕΥΔΗΣ.
168. Οι λογικές τιμές είναι οι εξής: ΟΧΙ, ΚΑΙ, Ή.
169. Στη δομή επανάληψης Για αν το βήμα είναι μηδέν, σε κάθε περίπτωση, ο βρόχος εκτελείται άπειρες φορές.
170. Ο βρόχος Για Κ από 5 μέχρι 1 εκτελείται 5 φορές.
171. Στην επαναληπτική δομή Για ... από ... μέχρι ... με\_βήμα οι τιμές από, μέχρι και με\_βήμα δεν είναι απαραίτητο να είναι ακέραιες.
172. Κάθε βρόχος που υλοποιείται με την εντολή Όσο ... επανάλαβε μπορεί να γραφεί και με χρήση της εντολής Για ... από ... μέχρι.
173. Η συνθήκη στην εντολή Όσο...επανάλαβε ελέγχεται τουλάχιστον μια φορά.
174. Μπορεί να υπάρχουν περισσότεροι από έναν αλγόριθμοι για την επίλυση ενός προβλήματος.

## II. Κεφάλαια 3-9

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. Όταν ψάχνουμε σε ένα τηλεφωνικό κατάλογο χρησιμοποιούμε τη σειριακή μέθοδο αναζήτησης
7. Η δυναμική παραχώρηση μνήμης είναι η τεχνική που χρησιμοποιείται στους πίνακες
8. Η ταξινόμηση είναι χρήσιμη διαδικασία γιατί έτσι εκτελείται γρηγορότερα η αναζήτηση
9. Δομή δεδομένων είναι ένα σύνολο δεδομένων που μπορούμε να εφαρμόσουμε μια σειρά λειτουργιών
10. Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα
11. Η ταξινόμηση ευθείας ανταλλαγής είναι πολύ αποτελεσματική αν ο πίνακας έχει λίγα στοιχεία
12. Για να εφαρμοστεί η μέθοδος της σειριακής αναζήτησης είναι απαραίτητο τα στοιχεία να είναι ταξινομημένα
13. Τα στοιχεία ενός πίνακα είναι απαραίτητο να είναι όλα του ίδιου τύπου
14. Η σειριακή αναζήτηση μπορεί να οδηγήσει στην προσπέλαση ακόμη και ολόκληρου του πίνακα
15. Η ταξινόμηση έχει ως στόχο να διατάζει τα στοιχεία ενός μονοδιάστατου πίνακα με αύξουσα ή φθίνουσα διάταξη
16. Η σειριακή αναζήτηση χρησιμοποιείται κυρίως για μικρούς ή μη ταξινομημένους πίνακες
17. Η σειριακή αναζήτηση μπορεί να εκτελεστεί μόνο σε μη ταξινομημένους πίνακες
18. Ο πίνακας είναι μια δυναμική δομή δεδομένων
19. Η ταξινόμηση της φυσαλίδας ταξινομεί τα στοιχεία ενός μονοδιάστατου πίνακα μόνο σε αύξουσα σειρά
20. Η θέση ενός στοιχείου σ' έναν διδιάστατο πίνακα καθορίζεται από δυο αριθμούς
21. Οι διαστάσεις ενός πίνακα μπορούν να μεταβληθούν κατά την διάρκεια εκτέλεσης ενός αλγορίθμου
22. Η χρήση πινάκων έχει το μειονέκτημα της υπερβολικής χρήσης μνήμης
23. Η ταξινόμηση εφαρμόζεται και σε διδιάστατους πίνακες
24. Στο ΠΙΝΑΚΑΣ[α, β] το α αντιστοιχεί στη γραμμή του πίνακα και το β στη στήλη
25. Προσπέλαση είναι η εύρεση ενός κόμβου με κάποιο κριτήριο
26. Για να προσπελάσουμε τα στοιχεία ενός πίνακα χρησιμοποιούμε επαναληπτική δομή
27. Για τον υπολογισμό μέσου όρου 120 αριθμών πρέπει να χρησιμοποιηθεί πίνακας
28. Δεν μπορεί να χρησιμοποιηθεί ο αλγόριθμος της φυσαλίδας σε πίνακα χαρακτήρων
29. Η ταξινόμηση της φυσαλίδας χρησιμοποιείται μόνο σε ταξινομημένους πίνακες
30. Οι διδιάστατοι πίνακες μπορούν να θεωρηθούν ως μονοδιάστατοι πίνακες όπου κάθε θέση τους θεωρούνται άλλοι μονοδιάστατοι πίνακες
31. Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν (ταξινομηθούν) όλα τα στοιχεία.
32. Οι διαστάσεις ενός πίνακα μπορούν να τροποποιηθούν αν χρειάζεται κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
33. Μία από τις λειτουργίες επί των δομών δεδομένων είναι η εισαγωγή ενός στοιχείου στην δομή.
34. Ένα από τα πλεονεκτήματα των πινάκων είναι ότι χρειάζονται λίγο χώρο στην μνήμη.
35. Για να προσδιορίσουμε την θέση σε έναν διδιάστατο πίνακα χρειαζόμαστε μία μεταβλητή.

36. Οι πίνακες είναι μία δομή δεδομένων που στηρίζεται στην τεχνική δυναμικής παραχώρησης μνήμης.
37. Οι πίνακες καταναλώνουν μεγάλο χώρο στη μνήμη του υπολογιστή.
38. Η σειριακή αναζήτηση θα πρέπει να αποφεύγεται όταν ο πίνακας είναι μεγάλος.
39. Η ταξινόμηση των στοιχείων ενός πίνακα πρέπει να γίνεται πάντα πριν από την αναζήτηση
40. Όλοι οι πίνακες δηλώνονται στο τμήμα δήλωσης μεταβλητών του προγράμματος
41. Τα στοιχεία ενός πίνακα πρέπει να είναι του ίδιου τύπου
42. Η χρήση πινάκων αυξάνει την απαιτούμενη μνήμη για την εκτέλεση του προγράμματος
43. Ο δείκτης ενός μονοδιάστατου πίνακα πρέπει να είναι πάντα I
44. Ένα πρόγραμμα σε ΓΛΩΣΣΑ δεσμεύει τόσες συνεχόμενες θέσεις μνήμης για έναν πίνακα όσες και οι θέσεις που εμφανίζονται στο τμήμα δηλώσεων
45. Σε οποιοδήποτε σημείο ενός προγράμματος σε ΓΛΩΣΣΑ μπορούν να τοποθετηθούν σχόλια
46. Στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ δηλώνουμε τα ονόματα των πινάκων αλλά όχι και το μέγεθός τους
47. Ένα πρόγραμμα σε ΓΛΩΣΣΑ έχει τη δυνατότητα να τροποποιεί το μέγεθος ενός πίνακα στο τμήμα εντολών
48. Για την εκτύπωση όλων των περιεχομένων ενός πίνακα απαιτείται η χρήση δομών επανάληψης
49. Για την αναζήτηση σε ταξινομημένους πίνακες προτιμάται η δυαδική αναζήτηση
50. Η συγχώνευση δυο πινάκων έχει ως στόχο να συνενώσει δυο πίνακες
51. Κάθε δομή μπορεί να χρησιμοποιηθεί σε οποιοδήποτε πρόβλημα ή εφαρμογή
52. Δυναμικές είναι οι δομές που αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
53. Ένας πίνακας έχει σταθερό μέγεθος αλλά μεταβαλλόμενο περιεχόμενο
54. Ένας πίνακας μπορεί να αποθηκεύσει ακεραίους αριθμούς και ονόματα
55. Για να προσπελάσουμε τα στοιχεία ενός πίνακα χρησιμοποιούμε επαναληπτική δομή
56. Για τον υπολογισμό μέσου όρου 120 αριθμών πρέπει να χρησιμοποιηθεί πίνακας
57. Ένας πίνακας που χρησιμοποιεί δύο δείκτες για τον πλήρη προσδιορισμό της θέσης του κάθε στοιχείου του είναι πάντα α) γραμμικός β) διδιάστατος γ) μονοδιάστατος δ) τετραγωνικός
58. Η σειριακή αναζήτηση και η δυαδική αναζήτηση μπορούν να χρησιμοποιηθούν σε όλους τους μονοδιάστατους πίνακες
59. Οι διαστάσεις ενός πίνακα μπορούν να τροποποιηθούν αν χρειάζεται κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου
60. Οι στατικές δομές δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
61. Οι στατικές δομές έχουν σταθερό μέγεθος, το οποίο καθορίζεται κατά τη στιγμή του προγραμματισμού τους (στην αρχή του προγράμματος).
62. Οι στατικές δομές υλοποιούνται με πίνακες.
63. Οι δυναμικές δομές αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
64. Οι δυναμικές δομές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους αυξάνεται και μειώνεται, όταν στη δομή αντίστοιχα εισάγονται ή διαγράφονται δεδομένα.
65. Στις δυναμικές δομές το μέγεθος της μνήμης καθορίζεται κατά την στιγμή του προγραμματισμού.



66. Με δυναμικές δομές υλοποιούνται οι λίστες, τα δένδρα και οι γράφοι.
67. Η γλώσσα προγραμματισμού ΓΛΩΣΣΑ, που χρησιμοποιείται στο βιβλίο, υποστηρίζει μόνο στατικές δομές.
68. 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.
69. Στον πίνακα, κατά την πράξη της ταξινόμησης, αναδιατάσσονται οι κόμβοι του.
70. Η λογικής μεταβλητής done (ή εύρηκα ή OK) στον αλγόριθμος της σειριακής αναζήτησης λέγεται «σημαία» και χρησιμοποιείται προκειμένου να αποφευχθούν περιττές επαναλήψεις
71. Οι δυναμικές δομές είναι πιο ευέλικτες από τις στατικές δομές, γιατί επεκτείνονται και συρρικνώνονται κατά την διάρκεια εκτέλεσης του προγράμματος.
72. Η σειριακή αναζήτηση χρησιμοποιείται αποκλειστικά στους ταξινομημένους πίνακες.
73. Τα στοιχεία ενός πίνακα μπορεί να είναι διαφορετικού τύπου.
74. Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος αλγόριθμος ταξινόμησης.
75. Οι πίνακες δεν μπορούν να έχουν περισσότερες από δύο διαστάσεις.

### III. 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. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα
5. Χάρη στο δομημένο προγραμματισμό δημιουργούνται προγράμματα απλούστερα, κατανοήσιμα και εύκολα στη διόρθωση
6. Στο δομημένο προγραμματισμό ακολουθούνται οι αρχές του ιεραρχικού και του τμηματικού προγραμματισμού
7. Στην ιεραρχική σχεδίαση, η ανάλυση του αλγορίθμου πραγματοποιείται με την τεχνική «από πάνω προς τα κάτω»
8. Μια γλώσσα προγραμματισμού προσδιορίζεται από: το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τη σημασιολογία της
9. Λεξιλόγιο μιας γλώσσας είναι όλες οι ακολουθίες που δημιουργούνται από τα στοιχεία του αλφαβήτου της γλώσσας, τις λέξεις
10. Η γραμματική είναι το συντακτικό μιας γλώσσας
11. Δεν υπάρχουν διαφορές μεταξύ φυσικών και τεχνητών γλωσσών
12. Η ιεραρχική σχεδίαση διασπά ένα πρόβλημα σε υποπρόβληματα και τα επιλύει, λύνοντας το αρχικό πρόβλημα
13. Ο τμηματικός προγραμματισμός υλοποιεί την φιλοσοφία της ιεραρχικής σχεδίασης
14. Κάθε υποπρόβλημα της ιεραρχικής σχεδίασης αποτελεί μια ανεξάρτητη ενότητα στον τμηματικό προγραμματισμό

15. Ο δομημένος προγραμματισμός περιέχει την ιεραρχική σχεδίαση και τον τμηματικό προγραμματισμό
16. Ο δομημένος προγραμματισμός προϋποθέτει τη χρήση εξειδικευμένων αλγοριθμικών δομών
17. Ο διερμηνευτής (interpreter) μετατρέπει το εκτελέσιμο πρόγραμμα (executable) σε πρόγραμμα αντικείμενο (object)
18. Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα
19. Ο μεταγλωττιστής αναλαμβάνει το ρόλο της μετάφρασης από το πηγαίο πρόγραμμα στο εκτελέσιμο πρόγραμμα
20. Ο διερμηνευτής ελέγχει και μετατρέπει μια-μια τις εντολές του πηγαίου προγράμματος σε γλώσσα μηχανής
21. Ο συνδέτης μετατρέπει το πηγαίο πρόγραμμα σε βιβλιοθήκη
22. Το αντικείμενο πρόγραμμα είναι ουσιαστικά σε γλώσσα μηχανής
23. Ο συνδέτης είναι ένα πρόγραμμα ελέγχου των συντακτικών λαθών του πηγαίου προγράμματος
24. Τα λογικά λάθη εντοπίζονται από έναν μεταγλωττιστή αλλά όχι από έναν διερμηνευτή
25. Τα συντακτικά λάθη επιδιορθώνονται γρήγορα και εύκολα αλλά για τα λογικά λάθη η επιδιόρθωση είναι μια επίπονη διαδικασία
26. 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
27. Ο διερμηνευτής έχει το πλεονέκτημα ότι το πρόγραμμα εκτελείται γρηγορότερα
28. Τα σύγχρονα προγραμματιστικά περιβάλλοντα χρησιμοποιούν μικτές υλοποιήσεις διερμηνευτή και μεταγλωττιστή
29. Σ' ένα σύγχρονο προγραμματιστικό περιβάλλον απαιτείται η παρουσία συντάκτη για την συγγραφή του κειμένου των εντολών
30. Για την επιδιόρθωση των λογικών λαθών πολλές φορές ο προγραμματιστής καλείται να εκτελέσει το πρόγραμμά του επανειλημμένα
31. Ο μεταγλωττιστής σ' ένα σύγχρονο προγραμματιστικό περιβάλλον καθιστά την ύπαρξη του συνδέτη προαιρετική
32. Ο μεταγλωττιστής διορθώνει όλα τα συντακτικά λάθη με τη χρήση βιβλιοθηκών
33. Η παράλειψη μιας εντολής Τέλος\_αν είναι λογικό λάθος
34. Η χρήση της εντολής  $MO \leftarrow a + b + c / 3$  αντί της  $MO \leftarrow (a + b + c) / 3$  είναι λογικό λάθος
35. Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση
36. Ο εντοπισμός των συντακτικών λαθών σε ένα πρόγραμμα γίνεται από τον μεταγλωττιστή (compiler)
37. Κάθε προγραμματιστικό περιβάλλον διαθέτει τον μεταγλωττιστή του
38. Το εκτελέσιμο είναι το πρόγραμμα που εκτελείται από τον υπολογιστή
39. Υπάρχουν δυο μεγάλες κατηγορίες μεταφραστικών προγραμμάτων: οι μεταφραστές και οι μεταγλωττιστές
40. Ο μεταγλωττιστής εξάγει το πηγαίο πρόγραμμα
41. Κατά τη μεταγλώττιση, το εκτελέσιμο πρόγραμμα παράγεται πριν το αντικείμενο.

42. Το αποτέλεσμα του συνδέτη-φορτωτή είναι το πηγαίο πρόγραμμα.
43. Η συγγραφή του πηγαίου προγράμματος και η διόρθωση των λαθών του γίνεται με την βοήθεια του συντάκτη.
44. Τα λογικά λάθη μπορούν να εντοπιστούν από το διερμηνευτή αλλά δεν μπορούν στον μεταγλωττιστή.
45. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα
46. Χάρη στο δομημένο προγραμματισμό δημιουργούνται προγράμματα απλούστερα, κατανοήσιμα και εύκολα στη διόρθωση
47. Η χρήση και η εξέλιξη είναι οι δύο βασικές διαφορές μεταξύ φυσικών και τεχνητών γλωσσών
48. Ο δομημένος προγραμματισμός δεν είναι απλός ένα είδος προγραμματισμού, είναι μια μεθοδολογία σύνταξης προγραμμάτων.
49. Το αρχικό πρόγραμμα το οποίο δημιουργείται στο συντάκτη, λέγεται αντικείμενο πρόγραμμα.
50. Ο ιεραρχικός προγραμματισμός βασίζεται στη μέθοδο «από κάτω προς τα πάνω».
51. Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος.

#### IV. 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
11. Σε μια συνάρτηση μπορούμε να χρησιμοποιήσουμε την εντολή ΓΡΑΨΕ, ενώ σε μια διαδικασία όχι.
12. Ο αριθμός των τυπικών παραμέτρων κατά την κλήση ενός υποπρογράμματος πρέπει να είναι ίδιος με τον αριθμό των πραγματικών παραμέτρων στην δήλωση ενός προγράμματος.
13. Κάθε υποπρόγραμμα θα πρέπει να είναι ανεξάρτητο από άλλα υποπρογράμματα.
14. Μία συνάρτηση μπορεί να δεχτεί μία μόνο παράμετρο.
15. Μία συνάρτηση μπορεί να εμφανίσει στην οθόνη την τιμή που υπολόγισε.
16. Μία διαδικασία επιστρέφει στο πρόγραμμα που την έχει καλέσει τις τιμές όλων των παραμέτρων της.
17. Η λίστα των τυπικών παραμέτρων καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.
18. Η λίστα των πραγματικών παραμέτρων καθορίζει τις παραμέτρους στη κλήση του υποπρογράμματος
19. Σύμφωνα με την αρχή της απειριορίστης εμβέλειας όλες οι μεταβλητές είναι καθολικές.
20. Η απειριορίστη εμβέλεια ενισχύει την αρχή της αυτονομίας των υποπρογραμμάτων.

21. Στην «ΓΛΩΣΣΑ» οι μεταβλητές και οι σταθερές έχουν περιορισμένη εμβέλεια.
22. Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές (και σταθερές) να χρησιμοποιούνται σε ένα τμήμα προγράμματος.
23. Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται εμβέλεια.
24. Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων.
25. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται | μεταβλητές ως τοπικές ή καθολικές,
26. Η στοίβα χρόνου εκτέλεσης είναι απαραίτητη για την κλήση των υποπρογραμμάτων από το κυρίως πρόγραμμα.
27. Οι συναρτήσεις δεν μπορούν να έχουν εντολές εισόδου - εξόδου.
28. Δεν μπορεί να γίνει κλήση μιας διαδικασίας μέσα από μια συνάρτηση.
29. Ένα υποπρόγραμμα μπορεί να καλείται μόνο από το κύριο πρόγραμμα.
30. Κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα.
31. Η λειτουργία των διαδικασιών είναι πιο περιορισμένη από τη λειτουργία των συναρτήσεων.
32. Μεταξύ των υποπρογραμμάτων δεν πρέπει να υπάρχει ανεξαρτησία.
33. Η απεριόριστη εμβέλεια των μεταβλητών καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων.

### B. Συμπλήρωσης κενού

1. Σε μια \_\_\_\_\_ δομή δεδομένων το μέγεθος της μνήμης που χρησιμοποιείται δεν είναι προκαθορισμένο
2. Οι δυναμικές δομές δεδομένων στηρίζονται στην τεχνική \_\_\_\_\_
3. Η λίστα με τις \_\_\_\_\_ παραμέτρους καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.
4. Η λίστα με τις \_\_\_\_\_ παραμέτρους καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.
5. Μερικές γλώσσες προγραμματισμού ονομάζουν ορίσματα τις \_\_\_\_\_ παραμέτρους και απλά παραμέτρους τις \_\_\_\_\_ παραμέτρους.
6. Να συμπληρώσετε τα κενά στο παρακάτω τμήμα αλγορίθμου, έτσι ώστε σε ταξινομημένο σε αύξουσα σειρά πίνακα  $table[n]$ , να υλοποιείται η λειτουργία της δυαδικής αναζήτησης ενός στοιχείου  $key$ .

Δεδομένα //  $n, table, key$  //

$L \leftarrow$  \_\_\_\_\_

$R \leftarrow n$

$D \leftarrow$  \_\_\_\_\_

Όσο  $D \neq \Psi\epsilon\Upsilon\Delta\eta\varsigma$  και \_\_\_\_\_ επανάλαβε

$M \leftarrow (L+R) \text{ DIV } 2$

Αν  $table[M] = key$  τότε

$D \leftarrow \text{ΑΛΗΘΗΣ}$

$\Theta\epsilon\varsigma\eta \leftarrow M$

Αλλιώς\_αν \_\_\_\_\_ τότε

$L \leftarrow M+1$

Αλλιώς

$R \leftarrow$  \_\_\_\_\_

Τέλος\_αν

Τέλος\_επανάληψης

7. Να συμπληρώσετε τα κενά στο παρακάτω τμήμα αλγορίθμου, έτσι ώστε να υλοποιείται η λειτουργία της ταξινόμησης με επιλογή του πίνακα  $table[n]$ .

Δεδομένα //  $table, n$  //

Για  $i$  από 1 μέχρι \_\_\_\_\_

$k \leftarrow$  \_\_\_\_\_

$x \leftarrow table[i]$

Για  $j$  από \_\_\_\_\_ μέχρι  $n$

Αν  $x > table[j]$  τότε

$k \leftarrow j$

$x \leftarrow$  \_\_\_\_\_

Τέλος\_αν

Τέλος\_επανάληψης

$table[k] \leftarrow$  \_\_\_\_\_

$table[i] \leftarrow x$

Τέλος\_επανάληψης

8.

### Γ. Ανάπτυξης

1. Να περιγράψετε τα προβλήματα που είναι δυνατόν να παρουσιαστούν κατά την αναπαράσταση ενός αλγορίθμου, αν χρησιμοποιηθεί ελεύθερο κείμενο και φυσική γλώσσα κατά βήματα.
2. Να γράψετε τις διαφορές μεταξύ στατικών και δυναμικών δομών δεδομένων.
3. Να αναφέρετε δύο βασικές λειτουργίες επί των δομών δεδομένων που δεν μπορούν να χρησιμοποιηθούν στους πίνακες. Να αιτιολογήσετε την απάντησή σας.
4. Ποιες είναι οι διαφορές μεταξύ μεταγλωττιστή και διερμηνευτή.
5. Να αναφέρετε τα πλεονεκτήματα του τμηματικού προγραμματισμού.
6. Να γράψετε τον αλγόριθμο του πολλαπλασιασμού αλλά ρώσικα και να κάνετε σε πίνακα τις τιμές που λαμβάνουν οι μεταβλητές του αλγορίθμου αν για είσοδο δώσουμε τους αριθμούς 17 και 32.
7. Να βρείτε την τιμή της παρακάτω λογικής έκφρασης.  
(OXI(9MOD5 = 20 - 4\*2^2)) Η (5 + 7DIV4 > 4) ΚΑΙ ('B' < 'A')

### Δ. Μετατροπές

1. Να μετατραπεί το παρακάτω τμήμα προγράμματος σε ισοδύναμο με τη χρήση της δομής (αποκλειστικά) **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ ..... ΜΕΧΡΙΣ\_ΟΤΟΥ**.

```

ΔΙΑΒΑΣΕ Χ
ΟΣΟ Χ>0 ΕΠΑΝΑΛΑΒΕ
  ΓΙΑ Υ ΑΠΟ 7 ΜΕΧΡΙ 2 ΜΕ ΒΗΜΑ -2
    Χ←Χ-2
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Χ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

2. Δίνεται το παρακάτω τμήμα αλγορίθμου σε φυσική γλώσσα κατά βήματα. Να κωδικοποιήσετε τον αλγόριθμο σε ψευδογλώσσα σύμφωνα με τις αρχές του δομημένου προγραμματισμού:

```

Βήμα1:  Θέσε  $S = 0$ 
Βήμα2:  Θέσε  $i = 0$ 
Βήμα3:  Αν  $i \leq a$ , τότε πήγαινε στο Βήμα4, αλλιώς πήγαινε στο Βήμα11
Βήμα4:  Θέσε  $i = i + 2$ 
Βήμα5:  Θέσε  $k = 1$ 
Βήμα6:  Θέσε  $S = S + k^2$ 
Βήμα7:  Θέσε  $k = k + 0.5$ 
Βήμα8:  Αν  $k > i$ , τότε πήγαινε στο Βήμα9, αλλιώς πήγαινε στο Βήμα6
Βήμα9:  Τύπωσε τα  $S, k$ 
Βήμα10: Πήγαινε στο Βήμα3
Βήμα11: Τύπωσε το  $S$ 

```

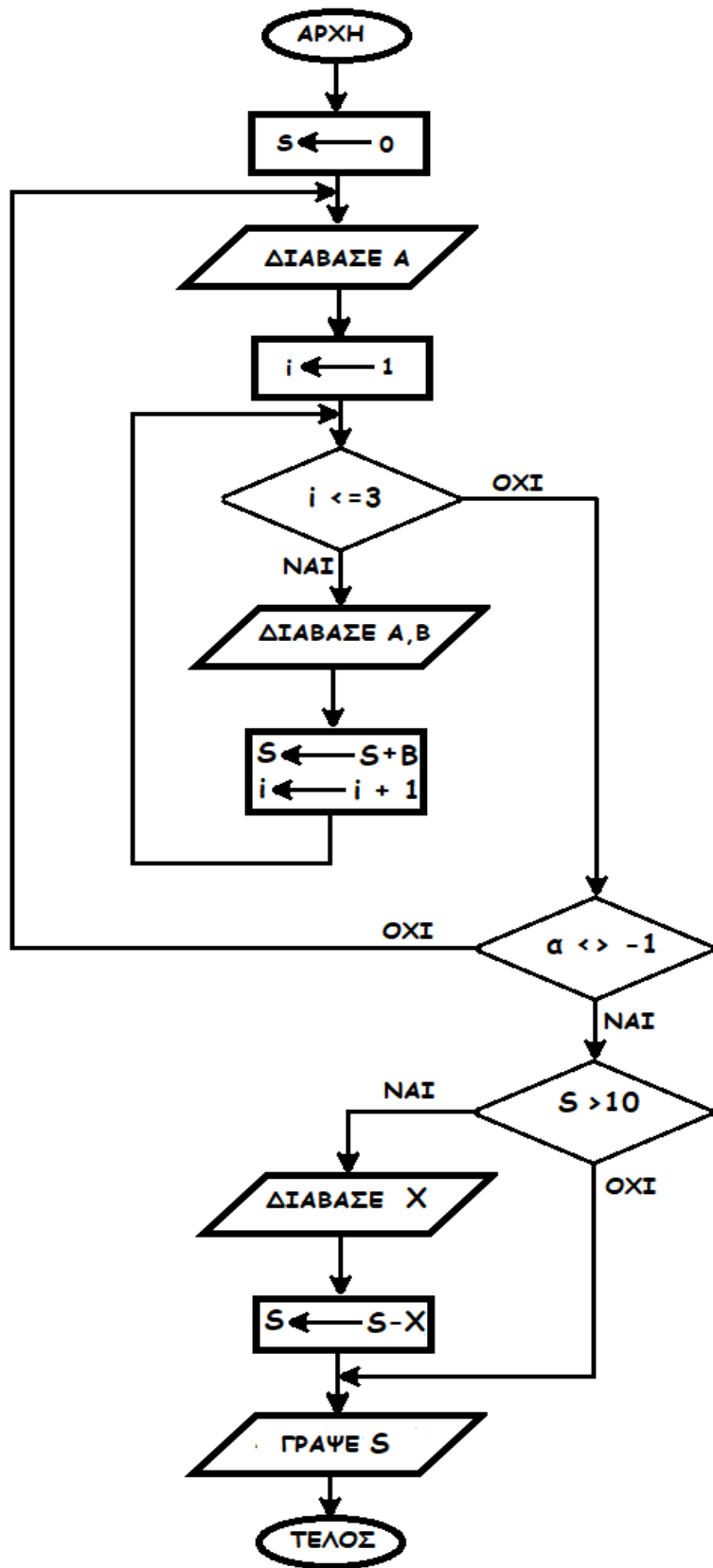
3. Να δημιουργήσετε το διάγραμμα ροής για το παρακάτω τμήμα προγράμματος:

```

Κ←95
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  Ρ←2
  Λ←0
  ΟΣΟ Ρ<=7 ΕΠΑΝΑΛΑΒΕ
    Λ←Λ-Ρ
    Ρ←Ρ+1
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Λ
Κ←Κ+Λ
ΜΕΧΡΙΣ_ΟΤΟΥ Κ<18
ΓΡΑΨΕ Κ

```

4. Να γράψετε σε μορφή κωδικοποιήσετε το παρακάτω διάγραμμα ροής.





## Ε. Εκτέλεση

1. Δίνεται το παρακάτω τμήμα αλγορίθμου στο οποίο έχουν αριθμηθεί όλες οι γραμμές.

1.  $Sum \leftarrow 0$
2. Διάβασε  $x$
3. Όσο  $x \leq 40$  επανάλαβε
4.     Αρχή\_επανάληψης
5.     Διάβασε  $αρ$
6.     Μέχρις\_ότου  $αρ >= 1$
7.      $Sum \leftarrow Sum + x$
8.     Διάβασε  $x$
9.     Τέλος\_επανάληψης
10. Εμφάνισε  $Sum$

Καθώς και το παρακάτω υπόδειγμα πίνακα τιμών, στο οποίο έχει συμπληρωθεί η πρώτη γραμμή.

Αριθμός γραμμής	$x \leq 40$	$αρ >= 1$	$x$	$αρ$	$Sum$	Έξοδος
1					0	
...	...	...	...	...	...	...

Να μεταφέρετε στο τετράδιο σας τον πίνακα και να τον συμπληρώσετε, εκτελώντας τον αλγόριθμο, αν δοθούν ως είσοδος κατά σειρά οι τιμές **20, 2, 40, -1, 1, -20, 3, 50**. Για κάθε εντολή που εκτελείται, να γράψετε σε νέα γραμμή του πίνακα τα εξής:

- Τον αριθμό της γραμμής που εκτελείται (στην πρώτη στήλη).
- Αν η γραμμή περιέχει εντολή εισόδου ή εντολή εκχώρησης, τη νέα τιμή της μεταβλητής στην αντίστοιχη στήλη. Αν η γραμμή περιέχει έλεγχο συνθήκης, την τιμή της συνθήκης (Αληθής, Ψευδής) στην αντίστοιχη στήλη. Αν η γραμμή περιέχει εντολή εξόδου, την τιμή που εμφανίζεται στην στήλη έξοδος.