

2019

ΚΕΦΑΛΑΙΟ 2

Algorithmos Computer
Science Laboratory

Αλγόριθμος



[ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ, ΔΟΜΗ ΕΠΙΛΟΓΗΣ, ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ]

[Επιλεγμένα θέματα θεωρίας σύμφωνα με την ύλη 2018-2019
στην ενότητα Έννοιες και Βασικά στοιχεία Αλγορίθμων - Προγραμμάτων
(Κεφάλαιο 2-7-8 του Σχολικού βιβλίου). Συγκεκριμένα αναλύονται:
Δομή Ακολουθίας, Δομή Επιλογής, Δομή Επανάληψης.]

Πίνακας περιεχομένων

1. Να δώσετε τον ορισμό του αλγορίθμου.....	4
2. Ποια είναι τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος	4
3. Να αναφερθούν και να περιγραφούν οι τρόποι αναπαράστασης ενός αλγορίθμου	5
4. Τι είναι οι μεταβλητές; Ποιοι είναι οι τύποι μεταβλητών	5
5. Τι είναι σταθερές;.....	5
6. Να αναφερθούν οι κανόνες ονοματολογίας των μεταβλητών.	6
7. Ποιο είναι το αλφάβητο της ΓΛΩΣΣΑΣ	6
8. Ποιοι οι τύποι δεδομένων που υποστηρίζει η ΓΛΩΣΣΑ	6
9. Τι χώρο καταλαμβάνουν στη μνήμη οι μεταβλητές διαφορετικών τύπων	7
10. Να γραφεί και να περιγραφεί η σύνταξη της εντολής εκχώρησης.....	7
11. Να περιγράψετε τη λειτουργία των εντολών ΔΙΑΒΑΣΕ και ΓΡΑΨΕ.....	7
12. Με ποιες εντολές ξεκινά και τερματίζεται η κωδικοποίηση στην ψευδογλώσσα	8
13. Ποιες είναι οι δηλωτικές και οι εκτελεστές εντολές; Τι είναι δεσμευμένες λέξεις	8
14. Πώς εισάγονται σχόλια σε έναν αλγόριθμο και τι ρόλο εξυπηρετούν;.....	8
15. Τι περιλαμβάνει μια αριθμητική έκφραση στη ΓΛΩΣΣΑ;.....	8
16. Να αναφέρετε τους αριθμητικούς τελεστές.....	9
17. Τι είναι οι τελεστές και ποιες είναι οι κατηγορίες των τελεστών;.....	9
18. Πως χρησιμοποιούνται οι αριθμητικοί τελεστές mod, div	9
19. Να αναφέρετε τις βασικές αλγοριθμικές δομές.	9
20. Να δώσετε τον ορισμό της δομής ακολουθίας.....	10
21. Ποιες είναι οι γνωστές συναρτήσεις της ΓΛΩΣΣΑΣ;	10
22. Να αναφέρετε τους τελεστές σύγκρισης.	10
23. Πως συντάσσεται μία απλή λογική έκφραση	10
24. Πως ορίζονται οι λογικές πράξεις	11
25. Τι ονομάζουμε σύνθετες λογικές εκφράσεις – συνθήκες.....	11
26. Να αναφέρετε: λογικές σταθερές, λογικούς τελεστές.....	11
3. Να αναφέρετε: συγκριτικό τελεστή, λογικό τελεστή, λογική σταθερά.....	12
27. Ποια είναι η ιεραρχία των τελεστών στις εκφράσεις	12
28. Πώς πραγματοποιείται η σύγκριση αλφαριθμητικών και λογικών τιμών;	12
29. Ποια είναι τα βασικά χαρακτηριστικά της απλής επιλογής;	13
30. Ποια είναι τα βασικά χαρακτηριστικά της σύνθετης επιλογής;	13
31. Ποια είναι τα βασικά χαρακτηριστικά της πολλαπλής επιλογής;	14
32. Τι είναι τα εμφωλευμένα Αν	14
33. Πότε χρησιμοποιούνται οι δομές επανάληψης σε έναν αλγόριθμο	14

34. Να περιγράψετε την επαναληπτική δομή ΟΣΟ	15
35. Να περιγράψετε την επαναληπτική δομή ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ ΟΤΟΥ.....	15
36. Να περιγράψετε την επαναληπτική δομή ΓΙΑ.	16
37. Πως υπολογίζω το πλήθος των επαναλήψεων στην ΓΙΑ;	16
38. Ποια είναι η βασική διαφορά ΟΣΟ και ΜΕΧΡΙΣ ΟΤΟΥ;.....	16
39. Ποια είναι η βασική διαφορά της Για από τις άλλες δυο δομές;	17
40. Περιγράψτε τον αλγόριθμο του πολλαπλασιασμού αλλά ρωσικά.	17
41. Τι είναι η ολίσθηση(Shift);	19
42. Ποιά η πρακτική σημασία του πολλαπλασιασμού αλλά ρώσικα;.....	18
43. Να γράψετε τους κανόνες των εμφωλευμένων βρόχων.	18
44. Τι γνωρίζετε για την χρήση τιμών στον τερματισμό μιας επαναληπτικής διαδικασίας...	18

1. Να δώσετε τον ορισμό του αλγορίθμου.

Αλγόριθμος είναι μια πεπερασμένη (δηλαδή συγκεκριμένη) σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο (δηλαδή όχι άπειρο) χρόνο που στόχο έχουν την επίλυση ενός προβλήματος.

2. Ποια είναι τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος ;

Κάθε αλγόριθμος πρέπει να ικανοποιεί τα κριτήρια : (Εξετάσεις 2004,2005 & 2011)

Είσοδος: Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον Αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών. Κάθε αλγόριθμος πρέπει απαραίτητα να ικανοποιεί το κριτήριο της εισόδου.

Για την είσοδο δεδομένων στην ψευδογλώσσα μπορούν να χρησιμοποιηθούν οι εντολές:

Διάβασε, για είσοδο από τον χρήστη (πληκτρολόγιο). Η εντολή αυτή μπορεί να εντοπιστεί σε οποιοδήποτε σημείο του κώδικα και όχι απαραίτητα στην πρώτη θέση. Κατά την εκτέλεση της εντολής, παγώνει η ροή εκτέλεσης τους αλγορίθμου και αναμένεται είσοδος τιμής από το πληκτρολόγιο.

Δεδομένα, για είσοδο με άλλον τρόπο (όταν κάτι θεωρείται γνωστό στα όρια του αλγορίθμου, χωρίς να διαβαστεί από τον χρήστη, θα μπορούσε να αποτελεί είσοδο από άλλον αλγόριθμο). Η εντολή αυτή, αν υπάρχει, στον κώδικα θα τοποθετηθεί μόνο ως πρώτη εντολή.

Έξοδος: Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.

Για την έξοδο δεδομένων σε κάποιον αλγόριθμο εκφρασμένο σε ψευδογλώσσα μπορούν να χρησιμοποιηθούν οι εντολές:

Εκτύπωσε, για έξοδο στον εκτυπωτή. Η εντολή αυτή, αν υπάρχει, μπορεί να τοποθετηθεί σε οποιοδήποτε σημείο του κώδικα. **Εμφάνισε**, για έξοδο στην οθόνη. Η εντολή αυτή, αν υπάρχει, μπορεί να τοποθετηθεί σε οποιοδήποτε σημείο του κώδικα.

Η εντολή **Γράψε** αποτελεί την εντολή εξόδου της ΓΛΩΣΣΑΣ και θεωρείται αποδεκτή εντολή εξόδου και στο περιβάλλον της ψευδογλώσσας .

Αποτελέσματα, για έξοδο με άλλον τρόπο (όταν κάτι πρέπει να αποτελεί έξοδο χωρίς ωστόσο να το ζητά η εκφώνηση. Η εντολή αυτή, αν υπάρχει, στον κώδικα θα τοποθετηθεί μόνο ως τελευταία εντολή.

Καθοριστικότητα: Οι εντολές ενός αλγορίθμου θα πρέπει να είναι επακριβώς και αυστηρώς καθορισμένες, έτσι πώς να μην υπάρχει καμία αμφιβολία ως προς την εκτέλεση τους. *Λόγου χάρη η διαίρεση με το μηδέν θα αποτελούσε πρόβλημα καθοριστικότητας.*

Περατότητα: Ο αλγόριθμος θα πρέπει να τερματίζει έπειτα από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μια διαδικασία η οποία δεν ολοκληρώνεται έπειτα από συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά απλώς μία υπολογιστική διαδικασία. Παράδειγμα:

$k \leftarrow 2$
Όσο $k > 0$ επανάλαβε
Εμφάνισε k
 $k \leftarrow k+1$
Τέλος_επανάληψης

Αρχή_επανάληψης
 $k \leftarrow 5$
εμφάνισε k
Μέχρις_ότου $k = 1$

Για i από 1 μέχρι 2 με_βήμα 0
Εμφάνισε i
Τέλος_επανάληψης

Αποτελεσματικότητα: Κάθε μεμονωμένη εντολή ενός αλγορίθμου πρέπει να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει οριστεί, αλλά πρέπει να είναι και εκτελέσιμη.

Αν επιχειρηθεί η διαίρεση 2 πραγματικών αριθμών που ο καθένας αναπαρίσταται από άπειρα δεκαδικά ψηφία, τότε ο αλγόριθμος δε διαθέτει το χαρακτηριστικό της αποτελεσματικότητας, αφού η διαίρεση αυτή ούτε μπορεί να αναπαρασταθεί πλήρως ούτε να εκτελεστεί ακριβώς.

3. Να αναφερθούν και να περιγραφούν οι τρόποι αναπαράστασης ενός αλγορίθμου . (Εξετάσεις 2000,2006,2011 & 2012)

➤ Ελεύθερο κείμενο

Είναι μία περιγραφή του αλγορίθμου με απλή καθημερινή γλώσσα. Αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί να παραβιάσει το κριτήριο της αποτελεσματικότητας καθώς μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση.

➤ Φυσική γλώσσα κατά βήματα

Αποτελείται και αυτός ο τρόπος από περιγραφή του αλγορίθμου, δομημένη όμως κατά βήματα. Χρειάζεται προσοχή, γιατί μπορεί να παραβιαστεί το κριτήριο του καθορισμού.

➤ Διαγραμματικές Τεχνικές (Εξετάσεις 2008 & 2014)

Είναι γραφικοί τρόποι παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο γνωστή, είναι το διάγραμμα ροής.

Σε ένα διάγραμμα ροής χρησιμοποιείται η έλλειψη, που δηλώνει την αρχή και το τέλος του αλγορίθμου το πλάγιο παραλληλόγραμμο, που δηλώνει την είσοδο και έξοδο τιμών, το ορθογώνιο παραλληλόγραμμο, που δηλώνει την εκτέλεση πράξεων, και ο ρόμβος, που εκφράζει τις συνθήκες οι οποίες διατυπώνονται σε έναν αλγόριθμο.

➤ Κωδικοποίηση

Ο αλγόριθμος αναπαρίσταται είτε με ένα πρόγραμμα γραμμένο σε κάποια γλώσσα προγραμματισμού είτε με μία ψευδογλώσσα που όταν εκτελεστεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

4. Τι είναι οι μεταβλητές; Ποιοι είναι οι τύποι μεταβλητών; (Εξετάσεις 2005)

Μεταβλητή είναι ένα γλωσσικό αντικείμενο που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου. Είναι μια ποσότητα που η τιμή της μπορεί να μεταβάλλεται. Σε μία μεταβλητή εκχωρείται (αποθηκεύεται) μόνο ένα δεδομένο, το οποίο μπορεί να αλλάζει κατά την διάρκεια εκτέλεσης του προγράμματος. Τα είδη των μεταβλητών είναι:

- Αριθμητικές (ακέραιες ή πραγματικές)
- Αλφαριθμητικές (κείμενο ή χαρακτήρες)
- Λογικές (τιμές αληθής/ψευδής).

5. Τι είναι σταθερές; (Εξετάσεις 2005)

Οι σταθερές (constants) αποτελούν προκαθορισμένες τιμές που δεν μεταβάλλονται κατά την διάρκεια εκτέλεσης του προγράμματος. Οι σταθερές μπορεί να είναι ακέραιες, πραγματικές, αλφαριθμητικές ή λογικές.

Προσοχή: Σταθερά \neq Συμβολική σταθερά

Στο τμήμα δηλώσεων ενός προγράμματος, αν γράψουμε για παράδειγμα $\pi=3$, αντιστοιχίζουμε το 3, μία σταθερά, με ένα σύμβολο, το π . Στη ΓΛΩΣΣΑ επιτρέπεται η αντιστοίχιση σταθερών τιμών με ονόματα, που ονομάζονται συμβολικές σταθερές. Τότε το γράμμα π είναι μία **συμβολική σταθερά** και ο υπολογιστής θα το βλέπει ως 3, (ενώ το 3 από μόνο του λέγεται σταθερά).

Να δώσετε από ένα παράδειγμα δήλωσης σταθεράς και δήλωσης μεταβλητής στη ΓΛΩΣΣΑ.

ΣΤΑΘΕΡΕΣ

$\pi = 3.14$! Το π είναι *συμβολική σταθερά* και το 3.14 *σταθερά*

ΦΠΑ = 0.24

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α

ΠΡΑΓΜΑΤΙΚΕΣ: β

6. Να αναφερθούν οι κανόνες ονοματολογίας των μεταβλητών.

Μια μεταβλητή μπορεί να έχει οποιοδήποτε όνομα αρκεί να ικανοποιεί τις ακόλουθες προϋποθέσεις:

1. Το όνομα μίας μεταβλητής προκύπτει μόνο από συνδυασμό μόνο :

- ✓ των γραμμάτων του ελληνικού αλφαβήτου (πεζά ή κεφαλαία)
- ✓ των γραμμάτων του λατινικού αλφαβήτου (πεζά ή κεφαλαία)
- ✓ των αριθμητικών ψηφίων από 0-9
- ✓ καθώς και της κάτω παύλας ($_$)

2. Ο πρώτος χαρακτήρας μίας μεταβλητής πρέπει υποχρεωτικά να είναι κάποιο γράμμα.

3. Το όνομα μίας μεταβλητής δεν μπορεί να είναι μία δεσμευμένη λέξη (γράψε, τέλος, διάβασε, mod κλπ).

Έγκυρα ονόματα:

$x1$, $\gamma 21$, μέσος_όρος, Δευτέρα, sum, SUM, Min_Max, όνομα κλπ.

Μη έγκυρα ονόματα:

μέσος όρος (περιέχει το σύμβολο «κενό»), διάβασε (δεσμευμένη λέξη)

4κ (ξεκίνημα με αριθμό), $\alpha + \beta$ (το μόνο σύμβολο που επιτρέπεται είναι « $_$ »)

7. Ποιο είναι το αλφάβητο της ΓΛΩΣΣΑΣ;

Οι χαρακτήρες του ελληνικού και λατινικού αλφαβήτου (πεζοί - κεφαλαίοι), τα ψηφία 0-9, κάποιοι ειδικοί χαρακτήρες: +, -, *, /, (,), [,], \!, & και ο χαρακτήρας του κενού.

8. Ποιοι οι τύποι δεδομένων που υποστηρίζει η ΓΛΩΣΣΑ;

Οι τύποι δεδομένων που υποστηρίζει η γλώσσα είναι οι ακόλουθοι: (Εξετάσεις 2006 & 2010)

Ακέραιος τύπος: Ο τύπος αυτός περιλαμβάνει τους ακέραιους αριθμούς όπως αυτοί είναι γνωστοί από τα μαθηματικά. Οι ακέραιοι μπορεί να είναι θετικοί αρνητικοί ή μηδέν.

Πραγματικός τύπος: Ο τύπος αυτός περιλαμβάνει τους πραγματικούς αριθμούς που γνωρίζουμε από τα μαθηματικά. Οι αριθμοί 3.543, -4.546, 0.02 είναι πραγματικοί αριθμοί. Οι πραγματικοί, όπως και οι ακέραιοι, μπορεί να είναι θετικοί αρνητικοί ή μηδέν.

Σημείωση! Οι ακέραιες και οι πραγματικές αποτελούν τον αριθμητικό τύπο.

Χαρακτήρας: Ο τύπος αυτός αναφέρεται τόσο σε ένα χαρακτήρα όσο και σε μία σειρά χαρακτήρων . Τα δεδομένα αυτού του τύπου μπορούν να περιέχουν οποιονδήποτε χαρακτήρα παράγεται από το πληκτρολόγιο και πρέπει να βρίσκονται μέσα σε απλά εισαγωγικά. Παραδείγματα χαρακτήρων αποτελούν : 'Ε' , '3ύ@' , 'Η ζωή είναι ωραία' , '48'

Λογικός: Αυτός ο τύπος δέχεται μόνο 2 τιμές , Αληθής ή Ψευδής .

Τύπος	Εντολή εκχώρησης
Ακέραιες	$K \leftarrow 10$
Πραγματικές	$\Phi\text{Π}\text{Α} \leftarrow 0.24$
Χαρακτήρες	$\text{Επίθετο} \leftarrow \text{'Ράσκος'}$
Λογικές	$\text{Flag} \leftarrow \text{Αληθής}$

9. Τι χώρο καταλαμβάνουν στη μνήμη του υπολογιστή οι μεταβλητές διαφορετικών τύπων;

Για κάθε μεταβλητή στον κώδικα δεσμεύεται συγκεκριμένο μέγεθος στη μνήμη του υπολογιστή. Στην πραγματικότητα τα δεδομένα καταχωρούνται στη μνήμη του υπολογιστή καταλαμβάνοντας συγκεκριμένο αριθμό θέσεων (bytes). Ανάλογα με τον τύπο του δεδομένου ποικίλλει το μέγεθος της μνήμης, αλλά και οι τιμές που μπορεί να λάβει. Έτσι, στον υπολογιστή διαθέτουμε ένα υποσύνολο ακέραιων ή πραγματικών αριθμών. Συνήθεις τύποι δεδομένων στις διάφορες γλώσσες προγραμματισμού είναι ο ακέραιος (integer) σε 1, 2 ή 4 bytes και ο πραγματικός (real) σε ή 8 bytes.

10. Να γραφεί και να περιγραφεί η σύνταξη της εντολής εκχώρησης.

Η **σύνταξη** της εντολής εκχώρησης είναι: [\(Εξετάσεις 2007\)](#)

Μεταβλητή \leftarrow έκφραση

Ως όνομα μεταβλητής επιτρέπεται οποιοδήποτε έγκυρο όνομα μεταβλητής σύμφωνα με τους κανόνες ονοματολογίας. Η έκφραση μπορεί να είναι:

- i. σταθερά
- ii. μεταβλητή
- iii. αριθμητική παράσταση
- iv. λογική παράσταση.

Αναφορικά με τη **λειτουργία**, αρχικά αποτιμάται η τιμή της έκφρασης στο δεξί σκέλος και αποδίδεται η τιμή αυτή στη μεταβλητή που υπάρχει αριστερά. Η έκφραση και η μεταβλητή πρέπει να είναι του ίδιου τύπου. Για τη σύνταξη της εντολής εκχώρησης δεν χρησιμοποιείται το = (που αποτελεί τελεστή σύγκρισης) αλλά το βέλος, που υποδηλώνει ιεράρχηση ενεργειών.

Εντολή	Η εντολή σε ελεύθερο κείμενο	Αποτέλεσμα εκτέλεσης
$x \leftarrow 10 + 2$	Πρόσθεσε το 10 με το 2 και αποθήκευσε το αποτέλεσμα στη μεταβλητή με όνομα x.	Το κελί κ περιέχει το 12.

11. Να περιγράψετε τη λειτουργία των εντολών ΔΙΑΒΑΣΕ και ΓΡΑΨΕ.

[\(Εξετάσεις 2005\)](#)

Οι εντολές εισόδου και εξόδου αντίστοιχα χρησιμοποιούνται για την αλληλεπίδραση με τον χρήστη και την ανταλλαγή δεδομένων και αποτελεσμάτων επεξεργασίας.

Δομή Ακολουθίας, Δομή Επιλογής, Δομή Επανάληψης

Την εντολή **ΔΙΑΒΑΣΕ** ακολουθεί λίστα μεταβλητών, χωρισμένων με κόμμα, με τιμές που ζητείται να εισαχθούν στο πρόγραμμα. Κατά την εκτέλεση της εντολής, παγώνει η ροή εκτέλεσης τους αλγορίθμου και αναμένεται είσοδος τιμής από το πληκτρολόγιο.

Μια εντολή **ΔΙΑΒΑΣΕ** μπορεί να διαβάσει μία ή περισσότερες μεταβλητές σε μία ή περισσότερες εντολές-γραμμές. Είναι ισοδύναμα τα παρακάτω:

ΔΙΑΒΑΣΕ α, β, γ	ΔΙΑΒΑΣΕ α
	ΔΙΑΒΑΣΕ β
	ΔΙΑΒΑΣΕ γ

Την εντολή **ΓΡΑΨΕ** ακολουθεί λίστα εκφράσεων (μεταβλητών, σταθερών και συμβολικών σταθερών, αριθμητικών ή λογικών πράξεων), χωρισμένων με κόμμα που θα προωθηθούν στη συσκευή εξόδου.

Μια εντολή **ΓΡΑΨΕ** μπορεί να εμφανίσει ένα ή περισσότερα στοιχεία και να εκτείνεται σε περισσότερες από μία γραμμές. Είναι ισοδύναμα τα παρακάτω:

ΓΡΑΨΕ α, β, γ, 'μήνυμα', 3.14	ΓΡΑΨΕ α, β, γ
	ΓΡΑΨΕ 'μήνυμα'
	ΓΡΑΨΕ 3.14

12. Με ποιες εντολές ξεκινά και τερματίζεται η κωδικοποίηση στην ψευδογλώσσα;

Για την έναρξη ενός αλγορίθμου χρησιμοποιείται η δεσμευμένη λέξη Αλγόριθμος και ακολουθεί το όνομά του. Ο τερματισμός δηλώνεται με τη δεσμευμένη λέξη Τέλος, που ακολουθείται από το ίδιο όνομα. Για την είσοδο τιμών από το πληκτρολόγιο χρησιμοποιείται η εντολή Διάβασε και ακολουθεί το όνομα της μεταβλητής.

13. Ποιες είναι και τι εκφράζουν οι δηλωτικές και οι εκτελεστές εντολές σε έναν αλγόριθμο; Τι είναι δεσμευμένες λέξεις;

Οι **εκτελεστές εντολές** πραγματοποιούν ενέργειες στον κώδικα (π.χ. εντολή Διάβασε κ.ά.), ενώ οι **δηλωτικές εντολές** νοηματοδοτούν συγκεκριμένες περιοχές στον κώδικα (π.χ. εντολή Αλγόριθμος, Τέλος κ.ά.).

Οι δεσμευμένες λέξεις είναι μια ακολουθία χαρακτήρων που τις μεταφράζει ο υπολογιστής ως λέξεις ειδικής σκοπιμότητας (π.χ. Διάβασε, Εμφάνισε κ.ά.) και δεν μπορούν να χρησιμοποιηθούν ως ονόματα μεταβλητών.

14. Πώς εισάγονται σχόλια σε έναν αλγόριθμο και τι ρόλο εξυπηρετούν;

Προκειμένου να διαχωρίζονται οι επεξηγηματικές φράσεις από τις εντολές ενός αλγορίθμου, χρησιμοποιείται το θαυμαστικό «!», και ότι ακολουθεί αγνοείται από τον αλγόριθμο. Τα σχόλια δεν παίζουν ρόλο στην εκτέλεση του αλγορίθμου.

15. Τι περιλαμβάνει μια αριθμητική έκφραση στη ΓΛΩΣΣΑ;

Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις. Κάθε έκφραση παριστάνει μια συγκεκριμένη αριθμητική τιμή, η οποία βρίσκεται μετά την εκτέλεση των πράξεων. Γι' αυτό είναι απαραίτητο όλες οι μεταβλητές που εμφανίζονται σε μία έκφραση να έχουν οριστεί προηγουμένως, δηλαδή να έχουν κάποια τιμή.

Σε μια αριθμητική έκφραση περιλαμβάνονται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί τελεστές και παρενθέσεις. Για την ιεραρχία εκτέλεσης των πράξεων και την αποτίμηση των εκφράσεων ακολουθούνται οι γνωστοί κανόνες των μαθηματικών.

16. Να αναφέρετε τους αριθμητικούς τελεστές και να δώσετε τη σειρά προτεραιότητας (ιεραρχία) των τελεστών αυτών στις αριθμητικές εκφράσεις. (Εξετάσεις 2001)

Οι αριθμητικοί τελεστές είναι οι: +, −, *, /, Λ, div, mod. Η ιεραρχία είναι η ίδια που ισχύει και στα μαθηματικά: Λ, * /, +, −. Όσον αφορά την ιεραρχία στις πράξεις, οι τελεστές div και mod ιεραρχούνται στο ίδιο επίπεδο με τον πολλαπλασιασμό και τη διαίρεση.

Ακόμη, η χρήση παρενθέσεων αλλάζει την ιεραρχία, ενώ, όταν υπάρχουν πράξεις ίδιας προτεραιότητας σε μια έκφραση, αυτές εκτελούνται από αριστερά προς τα δεξιά.

Αριθμητικός τελεστής	Πράξη	Παραδείγματα
+	Πρόσθεση	$13 + 5 = 18$
-	Αφαίρεση	$22 - 2 = 20$
*	Πολλαπλασιασμός	$2 * 7 = 14$
/	Διαίρεση	$11 / 2 = 5.5$
Λ	Ύψωση σε δύναμη	$3 \wedge 2 = 9$
Div	Πηλίκo Ακέραιας Διαίρεσης	$9 \text{ div } 2 = 4$
Mod	Υπόλοιπο Ακέραιας Διαίρεσης	$9 \text{ mod } 2 = 1$

17. Τι είναι οι τελεστές και ποιες είναι οι κατηγορίες των τελεστών; Να αναφέρετε τους τελεστές ανά κατηγορία. (Εξετάσεις 2007)

Τελεστές είναι ειδικά σύμβολα που χρησιμοποιούνται στις διάφορες πράξεις.

- Υπάρχουν οι αριθμητικοί τελεστές (+, −, *, /, Λ, div, mod),
- οι συγκριτικοί (>, <, >=, <=, =, <>) και
- οι λογικοί τελεστές (και, ή, όχι).

Τελεστές είναι οι αριθμοί ή οι μεταβλητές στους οποίους εφαρμόζονται οι τελεστές. Επιπλέον, οι εκφράσεις διαμορφώνονται από τελεστούς (μεταβλητές, σταθερές), τελεστές και παρενθέσεις.

18. Πως χρησιμοποιούνται οι αριθμητικοί τελεστές mod, div και σε ποιους τελεστούς μπορούν να εφαρμοστούν ;

Οι αριθμητικοί τελεστές div και mod υπολογίζουν το πηλίκo και το υπόλοιπο αντίστοιχα της ακέραιας διαίρεσης δυο θετικών ακέραιων αριθμών.

Πράξη	Διαιρετέος	Διαιρέτης
	Υπόλοιπο	Πηλίκo
$A \text{ mod } 10$		Ψηφίο μονάδων
$A \text{ mod } 100$		Τελευταία 2 ψηφία αριθμού
$A \text{ div } 10$ (όπου A διψήφιος)		Ψηφίο δεκάδων

$$\text{Υπόλοιπο} = \text{Διαιρετέος} \text{ mod } \text{Διαιρέτης} \quad | \quad \text{Πηλίκo} = \text{Διαιρετέος} \text{ div } \text{Διαιρέτης}$$

19. Να αναφέρετε τις βασικές αλγοριθμικές δομές. (Εξετάσεις 2001)

Δομή ακολουθίας, δομή επιλογής και δομή επανάληψης.

20. Να δώσετε τον ορισμό της δομής ακολουθίας.

Σύμφωνα με τη δομή ακολουθίας, μετά την εκτέλεση μιας εντολής ακολουθεί η εκτέλεση της επόμενης. Έτσι οι εντολές εκτελούνται όλες, διαδοχικά η μία μετά από την άλλη. Η ακολουθιακή δομή εντολών χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών. Πρόκειται για την πιο απλή αλγοριθμική δομή.

εντολή 1
 εντολή 2
 εντολή 3
 ...
 εντολή ν

21. Ποιες είναι οι γνωστές συναρτήσεις της ΓΛΩΣΣΑΣ;

Συνάρτηση	Περιγραφή	Παράδειγμα
1. T_P(x)	Υπολογισμός τετραγωνικής ρίζας	T_P(9) αντιστοιχεί στην $\sqrt{9}$
2. A_T(x)	Απόλυτη τιμή του x	A_T(-13+ 9) δίνει 10
3. A_M(x)*	Ακέραιο μέρος του x	A_M(3.1415) δίνει 3
4. HM(x)*	Υπολογισμός ημιτόνου	HM(90)
5. ΣΥΝ(x)*	Υπολογισμός συνημίτονου	ΣΥΝ(60)
6. ΕΦ(x)	Υπολογισμός εφαπτομένης	ΕΦ(45)
7. ΛΟΓ(x)	Υπολογισμός φυσικού λογάριθμου	ΛΟΓ(5)
8. E(x)	Υπολογισμός του e^x	E(t+3) αντιστοιχεί στο e^{t+3}

* Η γωνία x εκφράζεται σε μοίρες.

22. Να αναφέρετε τους τελεστές σύγκρισης. (Εξετάσεις 2001)

Οι συγκριτικοί τελεστές στην κωδικοποίηση είναι οι: >, <, >=, <=, =, <>. Στην ψευδογλώσσα και στο διάγραμμα ροής θα μπορούσαν να αξιοποιηθούν και οι τελεστές όπως αποτυπώνονται στα μαθηματικά: ≥, ≤, ≠.

23. Τι Πως συντάσσεται μία απλή λογική έκφραση ή αλλιώς μια απλή λογική συνθήκη;

Απλή έκφραση είναι εκείνη που δεν αξιοποιεί λογικούς τελεστές, αλλά μόνο συγκριτικούς (και πιθανώς αριθμητικούς). Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές, καθώς και παρενθέσεις. Μία απλή λογική συνθήκη σε έναν αλγόριθμο ή ένα πρόγραμμα συντάσσεται ως εξής:

	Έκφραση1	Συγκριτικός τελεστής	Έκφραση2
Παράδειγμα	x	>	5

Στις απλές λογικές συνθήκες γίνεται σύγκριση της τιμής μιας έκφρασης που βρίσκεται αριστερά από τον συγκριτικό τελεστή με την τιμή μιας έκφρασης που βρίσκεται δεξιά. Το αποτέλεσμα της είναι πάντα μία λογική τιμή **Αληθής** ή **Ψευδής**. Οι δύο εκφράσεις μπορεί να είναι: αριθμητικές, αλφαριθμητικές ή λογικές (πρέπει όμως να είναι του ίδιου τύπου).

Ο συγκριτικός τελεστής μπορεί να είναι ένας από τους παρακάτω:

Τελεστής	Ελεγχόμενη σχέση	Παράδειγμα
>	μεγαλύτερο από	πλήθος > 50
<	μικρότερο από	$\Sigma < x$
\geq ή \geq	μεγαλύτερο ή ίσο από	$(A + B) \geq 13$
\leq ή \leq	μικρότερο ή ίσο από	$A * B \leq A + B$
=	ίσο με	όνομα = 'Εύα'
<> ή \neq	διάφορο από	βρέθηκε <> αληθής

Στην ψευδογλώσσα μπορούμε ελεύθερα να χρησιμοποιήσουμε οποιονδήποτε από τους συγκριτικούς τελεστές, ενώ στη ΓΛΩΣΣΑ πρέπει να αποφεύγουμε τη χρήση των \geq , \leq , \neq .

24. Πως ορίζονται οι λογικές πράξεις;

Όπως ο υπολογιστής μπορεί να εκτελέσει πράξεις μεταξύ αριθμητικών τιμών, έτσι μπορεί να εκτελέσει πράξεις και μεταξύ λογικών τιμών. Συμπερασματικά σύμφωνα με τον πίνακα αληθείας, το αποτέλεσμα της σύζευξης (**ΚΑΙ**) δύο λογικών τιμών είναι αληθές μόνο αν είναι και οι δυο αληθείς. Το αποτέλεσμα της διάζευξης (**Η**) δύο λογικών τιμών είναι αληθές όταν αν είναι τουλάχιστον μία από τις δύο αληθείς. Η άρνηση (**ΟΧΙ**) αντιστρέφει μια λογική τιμή.

Συνθήκη1	Συνθήκη2	Σ1 ΚΑΙ Σ2	Σ1 Η Σ2	ΟΧΙ (Σ2)
Αληθής	Αληθής	Αληθής	Αληθής	Ψευδής
Αληθής	Ψευδής	Ψευδής	Αληθής	Αληθής
Ψευδής	Αληθής	Ψευδής	Αληθής	Ψευδής
Ψευδής	Ψευδής	Ψευδής	Ψευδής	Αληθής

25. Τι ονομάζουμε σύνθετες λογικές εκφράσεις – συνθήκες;

Ονομάζονται οι εκφράσεις που προκύπτουν από το συνδυασμό απλών λογικών συνθηκών που ο συνδυασμός τους επιτυγχάνεται με τη βοήθεια των **λογικών** τελεστών : ΟΧΙ, ΚΑΙ, Η. Το αποτέλεσμα των λογικών εκφράσεων προκύπτει από τον πίνακα αληθείας.

Μία σύνθετη λογική συνθήκη σε έναν αλγόριθμο ή ένα πρόγραμμα συντάσσεται ως εξής:

	Λογική Έκφραση1	Λογικός τελεστής	Λογική Έκφραση2
Παράδειγμα	$3 = \alpha$	>	$5 > \beta \wedge 2$

26. Να αναφέρετε:

- Τις λογικές σταθερές
- τους λογικούς τελεστές. (Εξετάσεις 2016)
- Πώς χρησιμοποιούνται οι λογικοί τελεστές;
- Ποια η ιεραρχία των λογικών τελεστών;

α.1. ΑΛΗΘΗΣ, ΨΕΥΔΗΣ

β. ΟΧΙ, ΚΑΙ, Η

γ. Οι τελεστές «και», «ή» χρησιμοποιούνται στις λογικές εκφράσεις για τη σύνδεση λογικών συνθηκών. Ο τελεστής «και» αποδίδει την τιμή αληθής μόνο όταν και οι δυο είναι αληθείς. Ο τελεστής «ή» αποδίδει την τιμή αληθής όταν τουλάχιστον μια από τις δύο προτάσεις είναι αληθής. Ο τελεστής «όχι» προσκολλάται στη συνθήκη η οποία ακολουθεί και την αντιστρέφει.

δ. Για την ιεραρχία των λογικών τελεστών ισχύουν τα εξής: πρώτα εκτελείται σε μια έκφραση ο τελεστής «όχι». Στη συνέχεια εκτελείται ο τελεστής «και» και τελευταίος ο τελεστής «ή». Έτσι άλλωστε ισχύει και στην άλγεβρα Boole, που αποτελεί το θεμέλιο για αυτά τα ζητήματα και υλοποιείται στις περισσότερες γλώσσες προγραμματισμού.

3. Να αναφέρετε:

- α. Έναν συγκριτικό τελεστή,
- β. Έναν λογικό τελεστή,
- γ. Μία λογική σταθερά,
- δ. Μια απλή λογική έκφραση,
- δ. Μία σύνθετη λογική έκφραση.

- α. Ένας εκ των τελεστών: $>$, $<$, $>=$, $<=$, $=$, $<>$
- β. Ένας εκ των τελεστών: ΟΧΙ, ΚΑΙ, Η
- γ. Μια εκ των τιμών: ΑΛΗΘΗΣ, ΨΕΥΔΗΣ
- δ. Μια μονή συνθήκη, όπως η: $\alpha > 0$
- ε. Μια σύνθετη διπλή συνθήκη όπως η: $\alpha > 0$ ΚΑΙ $\beta < 5$

27. Ποια είναι η ιεραρχία των τελεστών στις εκφράσεις;

Όταν συνδυάζονται όλες σε μία λογική έκφραση, τότε εκτελούνται πρώτα οι **αριθμητικοί** τελεστές, στη συνέχεια οι **συγκριτικοί** και τελευταίοι οι **λογικοί** τελεστές. Η ιεραρχία όλων των τελεστών κατά κατηγορία αποτυπώνεται στον πίνακα (οι τελεστές στο ίδιο επίπεδο ιεραρχίας εκτελούνται από αριστερά προς τα δεξιά). Φυσικά, η χρήση των παρενθέσεων επιτρέπει αλλαγές στην εκτέλεση των πράξεων, όπως γνωρίζουμε από τα μαθηματικά.

Α. Αριθμητικοί	1. \wedge 2. $*$, $/$, Div, Mod 3. $+$, $-$
Β. Συγκριτικοί	$>$, $<$, $>=$, $<=$, $=$, $<>$
Γ. Λογικοί	1. ΟΧΙ 2. ΚΑΙ 3. Η

*Για παράδειγμα, στην έκφραση $3 = \alpha$ ΚΑΙ $5 > \beta + 2$ θα εκτελεστεί πρώτα η πρόσθεση, στη συνέχεια θα ελεγχθούν οι συγκρίσεις και στο τέλος θα εκτελεστεί το ΚΑΙ.

28. Πώς πραγματοποιείται η σύγκριση αλφαριθμητικών και λογικών τιμών;

Η σύγκριση ατομικών χαρακτήρων βασίζεται στην αλφαβητική σειρά. Η σύγκριση αλφαριθμητικών βασίζεται στη χαρακτήρα προς χαρακτήρα σύγκριση μέχρι να βρεθεί διαφορά, όπως στον φυσικό κόσμο. Ισχύει $3 > 2 > 1$ και $'\gamma' > '\beta' > '\alpha'$, $'κακός' > 'καλός'$. Γενικά, η αλφαβητική σειρά ουσιαστικά είναι η αύξουσα διάταξη των αλφαριθμητικών.

Προσοχή! Στην περίπτωση των λογικών τιμών έχει νόημα η σύγκριση μόνο στο ίσο και στο διάφορο. Δεν έχει νόημα να συγκρίνουμε αν μια λογική μεταβλητή είναι μεγαλύτερη από κάποια άλλη.
Επιπλέον, Η σύγκριση μεταξύ δεδομένων **διαφορετικού τύπου** δεν έχει κανένα νόημα. Δηλαδή δεν είναι δυνατόν να συγκρίνουμε έναν αριθμό με ένα γράμμα ή με μία λογική τιμή.

29. Ποια είναι τα βασικά χαρακτηριστικά της απλής δομής επιλογής;

Η δομή απλής επιλογής χρησιμοποιείται όταν πρέπει να επιτελεστεί κάποια ενέργεια στην περίπτωση που ισχύει κάποια συνθήκη.
Η γενική σύνταξη της δομής επιλογής είναι:

Αν συνθήκη **τότε**
Εντολές
.....
Τέλος_αν

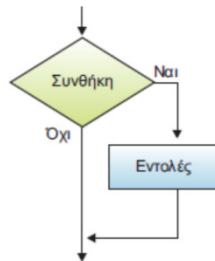
- Λειτουργία:

Η συνθήκη τοποθετείται μεταξύ των δεσμευμένων λέξεων **ΑΝ** και **ΤΟΤΕ**. Ελέγχεται η λογική συνθήκη και, αν αυτή είναι **ΑΛΗΘΗΣ**, τότε εκτελούνται οι εντολές μετά το **ΤΟΤΕ** και μέχρι το **ΤΕΛΟΣ_ΑΝ**. Στην περίπτωση που η λογική συνθήκη πάρει την τιμή **ΨΕΥΔΗΣ**, τότε η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το **ΤΕΛΟΣ_ΑΝ**.

Ή για μία μόνο εντολή, χωρίς **Τέλος αν** (ισχύει μόνο στην ψευδογλώσσα):

Αν συνθήκη **τότε** εντολή

- Διάγραμμα ροής



30. Ποια είναι τα χαρακτηριστικά της σύνθετης δομής επιλογής;

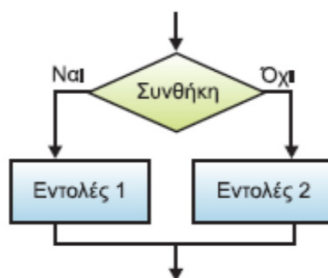
Η δομή σύνθετης επιλογής χρησιμοποιείται όταν πρέπει να εκτελεστεί μια ομάδα εντολών στην περίπτωση που ισχύει κάποια συνθήκη και μια άλλη ομάδα εντολών όταν η συνθήκη δεν ισχύει. Η συνθήκη τοποθετείται μεταξύ των δεσμευμένων λέξεων **Αν** και **τότε**. Οι εντολές που θα εκτελεστούν όταν η συνθήκη ισχύει τοποθετούνται μεταξύ των δεσμευμένων λέξεων **τότε** και **Αλλιώς** και αυτές που θα εκτελεστούν στην περίπτωση που η συνθήκη δεν ισχύει τοποθετούνται μεταξύ των δεσμευμένων λέξεων **Αλλιώς** και **Τέλος_αν**. Η γενική σύνταξη της δομής επιλογής είναι:

Αν συνθήκη **τότε**
Εντολές 1
.....
Αλλιώς
Εντολές 2
.....
Τέλος_αν

- Λειτουργία:

Ελέγχεται η λογική συνθήκη και, αν αυτή είναι **ΑΛΗΘΗΣ**, εκτελούνται οι **εντολές 1** ενώ, αν αυτή είναι **ΨΕΥΔΗΣ**, εκτελούνται οι **εντολές 2**. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το **ΤΕΛΟΣ_ΑΝ**.

- Διάγραμμα ροής



31. Ποια είναι τα χαρακτηριστικά της πολλαπλής δομής επιλογής;

Η δομή πολλαπλής επιλογής χρησιμοποιεί διαδοχικές εντολές **Αλλιώς_αν** (με αντίστοιχες συνθήκες), ώστε να μπορεί να καλύψει περισσότερες από δύο περιπτώσεις. Εκτελούνται οι εντολές που βρίσκονται στο αντίστοιχο τμήμα όταν η συνθήκη είναι αληθής. Αν καμία συνθήκη δεν είναι αληθής, εκτελούνται οι εντολές που βρίσκονται στο τμήμα **Αλλιώς**. Αν εισέλθει η ροή σε κάποιο τμήμα, στη συνέχεια η ροή μεταβαίνει στο Τέλος_αν και δεν ελέγχονται οι συνθήκες που ακολουθούν.

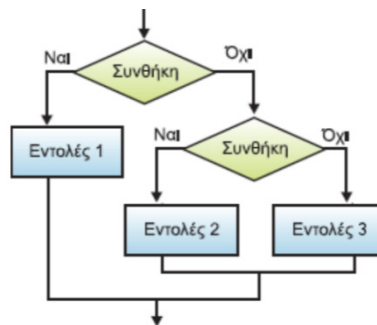
Αν συνθήκη 1 τότε
 Εντολές 1
Αλλιώς_αν συνθήκη2 τότε
 Εντολές 2

Αλλιώς
 Εντολές N
 Τέλος_αν

- Λειτουργία:

Ελέγχεται η **λογική συνθήκη1** και, αν αυτή είναι **ΑΛΗΘΗΣ**, εκτελούνται οι **εντολές 1** ενώ, αν αυτή είναι **ΨΕΥΔΗΣ**, ελέγχεται η **λογική συνθήκη2** και εκτελούνται οι **εντολές 2** κτλ. Στην περίπτωση που όλες οι συνθήκες είναι ψευδείς, τότε εκτελείται η ομάδα εντολών που βρίσκεται μετά τη λέξη **αλλιώς**. Η εντολή αλλιώς τοποθετείται μόνο μία φορά στο τέλος της δομής και **δεν** είναι υποχρεωτική. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το **ΤΕΛΟΣ_ΑΝ**.

- Διάγραμμα ροής



Η δομή **σύνθετης επιλογής** μπορεί να καλύψει μόνο δυο περιπτώσεις με έλεγχο κάποιας συνθήκης, καθώς μια συνθήκη έχει ακριβώς δύο απαντήσεις. Η **δομή πολλαπλής επιλογής** μπορεί να καλύψει όσες περιπτώσεις χρειάζονται με έλεγχο αντίστοιχων συνθηκών.

32. Τι είναι τα εμφωλευμένα Αν;

Εμφωλευμένα Αν ονομάζονται δυο ή περισσότερες δομές επιλογής που περιέχονται η μια μέσα στην άλλη (η εσωτερική δομή επιλογής πρέπει να περικλείεται ολόκληρη εντός της εξωτερικής). Η γενική μορφή μιας τέτοιας δομής θα μπορούσε να είναι:

Αν Συνθήκη 1 τότε
 Αν Συνθήκη2 τότε
 Εντολές 1
 Αλλιώς
 Εντολές2
 Τέλος_αν
 Τέλος_αν

*Η οποιοσδήποτε σχετικός συνδυασμός.

33. Πότε χρησιμοποιούνται οι δομές επανάληψης σε έναν αλγόριθμο;

Οι δομές επανάληψης χρησιμοποιούνται όταν μια ακολουθία εντολών πρέπει να εκτελεστεί σε ένα σύνολο περιπτώσεων που έχουν κάτι κοινό. Το τμήμα εντολών που επαναλαμβάνεται και περιλαμβάνεται εντός της δομής επανάληψης αποκαλείται **βρόχος**.

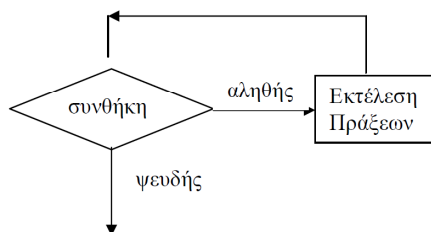
Μια δομή επανάληψης που βρίσκεται στο εσωτερικό μια άλλης δομής επανάληψης, χαρακτηρίζεται ως **εμφωλευμένη**. Εμφωλευμένες μπορούν να παρουσιαστούν με κάθε δυνατό συνδυασμό δομών επανάληψης, όπως, Για μέσα σε Για, Όσο μέσα σε Για, Αρχή_Επανάληψης μέσα σε Όσο, Όσο μέσα σε Όσο κ.ο.κ.

34. Να γράψετε τη σύνταξη της εντολής ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ και να περιγράψετε τη λειτουργία της. (Εξετάσεις 2015 & 2018)

Η δομή **Όσο** χρησιμοποιείται όταν επιθυμούμε να εκτελέσουμε κατ' επανάληψη ένα μπλοκ εντολών, όσο ισχύει μια συνθήκη (ονομάζεται συνθήκη συνέχειας), που βρίσκεται στην αρχή της επανάληψης. Η δομή ολοκληρώνεται με τη δεσμευμένη λέξη **Τέλος_επανάληψης**, η οποία κατευθύνει τη ροή εκτέλεσης ξανά στο **Όσο**. Γενική μορφή - σύνταξη:

Όσο συνθήκη **επανάλαβε**
 Εντολές
Τέλος_επανάληψης

Αναπαράσταση σε λογικό διάγραμμα της **Όσο**.



Κανόνες της Όσο:

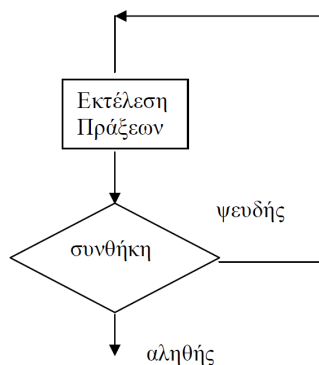
1. Μπορεί να μην εκτελεστεί και καμία φορά.
2. Η συνθήκη πρέπει να είναι αληθής για να γίνει η επανάληψη.
3. Τερματίζεται όταν η συνθήκη γίνει ψευδής.
4. Πρέπει η μεταβλητή της συνθήκης υποχρεωτικά, να έχει αρχική τιμή πριν την επανάληψη.
5. Πρέπει να υπάρχει αλλαγή της τιμής της μεταβλητής της συνθήκης της μέσα στο βρόγχο με τέτοιο τρόπο ώστε να τερματίζεται ο βρόγχος.

35. Για την εντολή ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ ΟΤΟΥ να γράψετε τη σύνταξή της και να περιγράψετε τη λειτουργία της. (Εξετάσεις 2003)

Η δομή **Μέχρις_ότου** χρησιμοποιείται όταν επιθυμούμε να εκτελέσουμε κατ' επανάληψη ένα μπλοκ εντολών μέχρι να ισχύσει κάποια συνθήκη. Η συνθήκη βρίσκεται στο τέλος της επανάληψης και αποκαλείται συνθήκη τερματισμού. Γενική μορφή - σύνταξη:

Αρχή_επανάληψης
 Διαδικασία
Μέχρις_ότου <συνθήκη>

Αναπαράσταση σε λογικό διάγραμμα της **Μέχρις_ότου**.



Κανόνες της Αρχής_Επανάληψης:

1. Θα εκτελεστεί τουλάχιστον μία φορά.
2. Η συνθήκη πρέπει να είναι ψευδής για να γίνει η επανάληψη.
3. Τερματίζεται όταν η συνθήκη γίνει Αληθής.
4. Δεν πρέπει η μεταβλητή της συνθήκης να έχει υποχρεωτικά, αρχική τιμή πριν την επανάληψη.
5. Πρέπει να υπάρχει αλλαγή της τιμής της μεταβλητής της συνθήκης της μέσα στο βρόγχο με τέτοιο τρόπο ώστε να τερματίζεται ο βρόγχος.

36. Για την εντολή **ΓΙΑ..** να γράψετε τη σύνταξή της και να περιγράψετε τη λειτουργία της.

Για μεταβλητή από τ_1 μέχρι τ_2 με_βήμα β
 Εντολές βρόχου
Τέλος_επανάληψης

ΚΑΝΟΝΕΣ

1. $\tau_1 \leq \tau_2$ με $\beta > 0$
2. $\tau_1 > \tau_2$ αρκεί $\beta < 0$

Κανόνες:

1. Η **Μεταβλητή** μπορεί να είναι οτιδήποτε, αλλά προτείνεται η χρήση του **i** σαν μεταβλητή του "**για**", για λόγους ομοιομορφίας και εύκολης αναγνώρισης.
2. Το τ_1 μπορεί να είναι ένας οποιοδήποτε αριθμός που δηλώνει την αρχή μέτρησης, ενώ το τ_2 επίσης οποιαδήποτε μεταβλητή που όμως θα δηλώνει το τέλος μέτρησης.
3. Το βήμα είναι ένας αριθμός που δηλώνει το βήμα αύξησης του μετρητή από το τ_1 μέχρι το τ_2 και μπορεί να παραληφθεί αν το βήμα είναι ο αριθμός 1.
4. Αν το βήμα είναι **μηδέν** (0) τότε δημιουργείται **ατέρμωνας βρόγχος**.
5. Απαγορεύεται η αλλαγή τιμής της **μεταβλητής – μετρητή** της **Για** μέσα στον βρόγχο. Η μεταβλητή παίρνει τις τιμές της αποκλειστικά με την **Για**. Το ίδιο ισχύει και για την αρχική, τελική τιμή και την τιμή του βήματος, αν αυτές οριστούν ως μεταβλητές στην δομή.
6. Το βήμα **δεν** είναι απαραίτητο να είναι ακέραιος αριθμός.
7. Η **Για** δεν έχει δικό της λογικό διάγραμμα. Την μετατρέπουμε σε **όσο** και την εμφανίζουμε με αυτό το διάγραμμα.

Η δομή Για χρησιμοποιείται όταν επιθυμούμε να εκτελέσουμε ένα μπλοκ εντολών για συγκεκριμένο πλήθος επαναλήψεων. Παράδειγμα: Διαβάζονται 10 αριθμοί και εκτυπώνονται οι διπλάσιοι τους.

Για i από 1 μέχρι 10
 Διάβασε α
 Εκτύπωσε α*2
Τέλος_επανάληψης

37. Πως υπολογίζω το πλήθος των επαναλήψεων στην **ΓΙΑ**;

$$(87-12) \text{ div } 2 + 1 = 75 \text{ div } 2 + 1 = 37+1 = 38$$

$$\text{Τύπος: } (TT - AT) \text{ div } B + 1$$

Για κ από 12 μέχρι 87 με_βήμα 2
Εμφάνισε "ALGORITHMOS"
Τέλος_επανάληψης

38. Ποιες είναι οι διαφορές μεταξύ των δομών Όσο και Μέχρις_ότου;

Η δομή **Όσο** ελέγχει τη συνθήκη στην αρχή του βρόχου, ενώ η δομή **Μέχρις_ότου** στο τέλος. Αυτό έχει ως συνέπεια για την υλοποίηση του ίδιου βρόχου με τις δομές Όσο και Μέχρις_ότου οι συνθήκες που χρησιμοποιούμε να είναι αντίθετες. Επίσης, ένας βρόχος **Μέχρις_ότου** εκτελείται τουλάχιστον μία φορά, ενώ σε μια δομή Όσο υπάρχει περίπτωση να μην εκτελεστεί καμία επανάληψη. Υπάρχουν περιπτώσεις που η χρήση της **Μέχρις_ότου** οδηγεί σε απλούστερους και πιο ευκολονόητους αλγορίθμους. Η δομή Μέχρις_ότου προτιμάται:

1. Όταν πρέπει να εκτελεστεί οπωσδήποτε μια επανάληψη
2. στον έλεγχο αποδεκτών τιμών,
3. καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού.

39. Ποια είναι η βασική διαφορά της Για από τις άλλες δυο δομές;

Η δομή Για χρησιμοποιείται για προκαθορισμένο αριθμό επαναλήψεων, ενώ για τις άλλες δυο δομές δεν είναι γνωστός εκ των προτέρων και εξαρτάται από τις συνθήκες που τις ελέγχουν.

40. Περιγράψτε τον αλγόριθμο του πολλαπλασιασμού αλά ρωσικά.

Ο αλγόριθμος του πολλαπλασιασμού αλά ρωσικά είναι ο τρόπος που ο υπολογιστής υπολογίζει το γινόμενο δύο θετικών ακεραίων πραγματοποιώντας συνεχώς πολλαπλασιασμό και διαίρεση με το 2.

Έχουμε έναν πρώτο αριθμό **M1** που συνεχώς διπλασιάζεται και έναν δεύτερο **M2** που συνεχώς υποδιπλασιάζεται με το **DIV2**. Νωρίτερα όμως ένας αθροιστής **Σ** αυξάνεται κατά τον πρώτο αριθμό μόνο αν ο δεύτερος είναι μονός (περιττός). Η διαδικασία γίνεται επαναληπτικά μέχρι ο δεύτερος αριθμός να γίνει **0**. Η μέθοδος αυτής χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ότι ο γνωστός μας χειρονακτικός τρόπος πολλαπλασιασμού.

Αλγόριθμος Πολλαπλασιασμός_αλά_Ρωσικά

$\Sigma \leftarrow 0$

διάβασε M1, M2

όσο M2 $\neq 0$ επανέλαβε

αν M2 mod 2 = 1 τότε

$\Sigma \leftarrow \Sigma + M1$

τέλος_αν

$M1 \leftarrow M1 * 2$

$M2 \leftarrow M2 \text{ div } 2$

Τέλος_επανάληψης

Γράψε Σ

Τέλος Πολλαπλασιασμός_αλά_Ρωσικά

Παράδειγμα: M1= 45 & M2=19

M1	M2	P
45	19	45
90	9	135
180	4	
360	2	
720	1	855
1140	0	

41. Τι είναι η ολίσθηση(Shift);

Στα κυκλώματα του υπολογιστή τα δεδομένα αποθηκεύονται σε δυαδική μορφή, δηλαδή 0 και 1, ανεξάρτητα από το πώς τα ορίζει ο προγραμματιστής (όπως ακέραια, πραγματικά, χαρακτήρες ή λογικά δεδομένα). Ο αριθμός **17** στο δεκαδικό σύστημα αντιστοιχεί για παράδειγμα στην ακολουθία **00010001** του δυαδικού συστήματος. Δηλαδή:

$$(17)_{10} = (00010001)_2$$

Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα αριστερά (shift left), δηλαδή αν προσθέσουμε ένα 0 στο τέλος του αριθμού και αγνοήσουμε το αρχικό 0, τότε θα προκύψει ο αριθμός **00100010** ο οποίος αντιστοιχεί στον αριθμό **34** του δεκαδικού συστήματος

$$(00100010)_2 = (34)_{10} \text{ (πολλαπλασιάσαμε το 17 επί 2)}$$

Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα δεξιά (shift right), δηλαδή αν κόψουμε το τελευταίο 1 και εισάγουμε ένα 0 στην αρχή, τότε θα προκύψει ο αριθμός **00001000** ο οποίος αντιστοιχεί στον αριθμό **8** του δεκαδικού συστήματος

$$(00001000)_2 = (8)_{10} \text{ (διαίρεσαμε το 17 δια 2, με ακέραια διαίρεση, δηλαδή div)}$$

Επομένως, στο δυαδικό σύστημα:

- η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί 2, και
- η ολίσθηση προς τα δεξιά ισοδυναμεί με ακέραια διαίρεση (div) δια 2

42. Ποιά η πρακτική σημασία του πολλαπλασιασμού αλλά ρώσικα;

Αν σε έναν αριθμό στο δυαδικό σύστημα προσθέσουμε ένα ψηφίο 0 από δεξιά μη λαμβάνοντας υπόψη το πρώτο από αριστερά ψηφίο, τότε ο αριθμός που προκύπτει είναι ο **διπλάσιος** του αρχικού. Αν προσθέσουμε ένα ψηφίο 0 από αριστερά και μη λαμβάνοντας υπόψη το πρώτο από δεξιά ψηφίο, τότε ο αριθμός που προκύπτει είναι το **μισό** (ακέραιο) του αρχικού αριθμού.

Σε επίπεδο κυκλωμάτων_υπολογιστή ο πολλαπλασιασμός επί 2 και η διαίρεση δια του 2 μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης, σε αντίθεση με τον «παραδοσιακό» πολλαπλασιασμό, που θεωρείται χρονοβόρα διαδικασία.

Η ολίσθηση προς τα **αριστερά** ισοδυναμεί με **πολλαπλασιασμό με το δυο** και ολίσθηση προς τα **δεξιά** ισοδυναμεί με **ακέραια διαίρεση διά δύο** στο δυαδικό πάντα σύστημα. Με τον αλγόριθμο του πολλαπλασιασμού αλλά ρωσικά πραγματοποιείται ουσιαστικά ο πολλαπλασιασμός στον ηλεκτρονικό υπολογιστή.

43. Να γράψετε τους κανόνες που πρέπει να ακολουθούνται στη χρήση των εμφωλευμένων βρόχων. (Εξετάσεις 2007 & 2014)

1. Ο εσωτερικός βρόχος πρέπει να βρίσκεται εξολοκλήρου μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος ολοκληρώνεται πρώτος.
2. Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του.
3. Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

44. Τι γνωρίζετε για την χρήση τιμών για τον τερματισμό μιας επαναληπτικής διαδικασίας;

Είναι συχνά επιθυμητό να τερματίζεται μια επανάληψη όταν η τιμή μιας ή περισσότερων μεταβλητών πάρει μια συγκεκριμένη τιμή. Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή πρέπει να είναι τέτοια ώστε να μην είναι λογικά σωστή για το πρόβλημα.

Έτσι πχ όταν διαβάζω τα ύψη πολλών ανθρώπων η τιμή τερματισμού μπορεί να είναι -1 αφού αποκλείεται να υπάρχει ύψος ανθρώπου -1. Μια καλή πρακτική είναι η χρήση λογικών μεταβλητών (σημαίες - flags) οι οποίες έχουν αρχική τιμή πριν την επανάληψη Αληθής, να αλλάζουν μέσα στην επανάληψη σε Ψευδής, και η συνθήκη τερματισμού να εξαρτάται από την τιμή αυτή.

Η τιμή αυτή συχνά αποκαλείται και «**τιμή φρουρός**».