

ΜΑΘΗΜΑ 2

ΑΛΓΟΡΙΘΜΟΙ

ΣΤΟ ΠΡΟΗΓΟΥΜΕΝΟ ΜΑΘΗΜΑ...

Ολοκληρώσαμε την θεωρία του 1^{ου} κεφαλαίου και κάποια βασικά θέματα που εξετάσαμε ήταν:

- Ορισμός προβλήματος
- Ορισμός δεδομένου, πληροφορίας και επεξεργασίας δεδομένων
- Το στάδιο της κατανόησης ενός προβλήματος (τι προϋποθέτει και που αποσκοπεί)
- Το στάδιο της ανάλυσης ενός προβλήματος και τι είναι η δομή ενός προβλήματος
- Τα 3 στάδια αντιμετώπισης ενός προβλήματος

Ξεκινήσαμε την θεωρία του 2^{ου} κεφαλαίου και κάναμε μια αναφορά στα παρακάτω

- Ορισμός αλγορίθμου
- Κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος
- Τρόποι αναπαράστασης ενός αλγορίθμου
- Διαφορές στο συντακτικό μεταξύ αλγορίθμου και προγράμματος

01

ΣΩΣΤΟ / ΛΑΘΟΣ

Οι αλγόριθμοι συνδέονται αποκλειστικά και μόνο με προβλήματα της πληροφορικής

Λ

Η περατότητα είναι ένα από τα αλγοριθμικά κριτήρια

Σ

Ένας αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο ενεργειών

Σ

Ένας αλγόριθμος στοχεύει στην επίλυση ενός προβλήματος

Σ

Ένας αλγόριθμος μπορεί να μην διαθέτει έξοδο

Λ

Οι ενέργειες που ορίζονται σε κάποιον αλγόριθμο πρέπει να είναι αυστηρά καθορισμένες

Σ

Όλα τα προβλήματα μπορούν να επιλυθούν αλγοριθμικά

Λ

Ένας αλγόριθμος που αναπαριστάται με φυσική γλώσσα κατά βήματα μπορεί να παραβιάζει το κριτήριο της αποτελεσματικότητας

Λ

Ένας αλγόριθμος που αναπαριστάται με ελεύθερο κείμενο μπορεί να παραβιάζει το κριτήριο της αποτελεσματικότητας

Σ

Ένας αλγόριθμος πρέπει απαραίτητα να έχει έξοδο

Σ

Μια διαδικασία που δεν τελειώνει έπειτα από συγκεκριμένο αριθμό βημάτων αποτελεί υπολογιστική διαδικασία

Σ

Κάθε εντολή ενός αλγορίθμου πρέπει να καθορίζεται χωρίς αμφιβολία για τον τρόπο εκτέλεσης της

Σ

Η αναπαράσταση ενός αλγορίθμου μπορεί να γίνει μόνο με χρήση ελεύθερου κειμένου και φυσικής γλώσσας

Λ

Μια υπολογιστική διαδικασία που δεν τελειώνει έπειτα από συγκεκριμένο αριθμό βημάτων αποτελεί αλγόριθμο

Λ

Ο πιο αποδοτικός τρόπος παρουσίασης αλγορίθμων είναι η κωδικοποίηση

Σ

Στη δομή ενός προβλήματος περιλαμβάνονται τα συστατικά του μέρη

Σ

Η κατανόηση της δομής ενός προβλήματος προϋποθέτει ότι έχει αρχίσει η διαδικασία της ανάλυσης ενός προβλήματος

Σ

Η κατανόηση της δομής ενός προβλήματος προϋποθέτει την κατανόηση του προβλήματος

Σ

Για τη γραφική απεικόνιση της δομής ενός προβλήματος χρησιμοποιείται συχνά η διαγραμματική αναπαράσταση

Σ

Η κατανόηση της δομής ενός προβλήματος προϋποθέτει την κατανόηση του προβλήματος

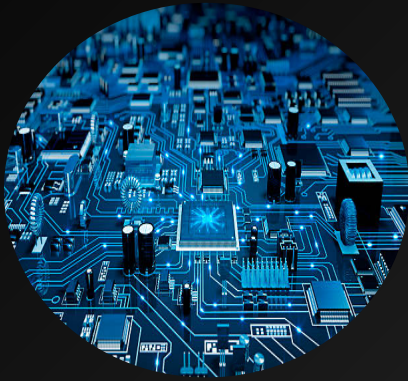
Σ

A series of thin, dark, wavy lines that create a sense of motion and depth, flowing across the bottom of the slide.

02

ΣΠΟΥΔΑΙΟΤΗΤΑ ΑΛΓΟΡΙΘΜΩΝ

Σκοπιές από τις οποίες μελετά τους αλγορίθμους η πληροφορική



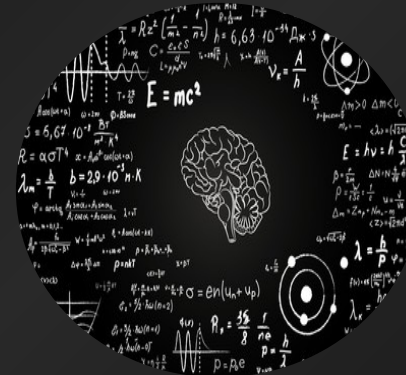
Υλικού

Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού και την αρχιτεκτονική του υπολογιστή



Γλωσσών προγραμματισμού

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την υλοποίηση ενός αλγορίθμου επηρεάζει την ταχύτητα εκτέλεσης του. Αν χρησιμοποιούμε γλώσσα χαμηλότερου επιπέδου (πχ C++) τότε η ταχύτητα εκτέλεσης θα αυξηθεί σε σχέση με την χρήση μιας γλώσσας υψηλότερου επιπέδου (πχ JavaScript)



Θεωρητική

Διερευνά αν υπάρχει ή όχι αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος και προσδιορίζει τα όρια της λύσης



Αναλυτική

Μελετώνται οι υπολογιστικοί πόροι που απαιτούνται από έναν αλγόριθμο για την εκτέλεση του όπως το μέγεθος της κύριας και δευτερεύουσας μνήμης, CPU και λειτουργίες εισόδου εξόδου

03

ΣΥΝΤΑΚΤΙΚΟ
ΨΕΥΔΟΓΛΩΣΣΑΣ
ΚΑΙ
ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΡΟΓΡΑΜΜΑ Υπολογισμός
ΣΤΑΘΕΡΕΣ

ΦΠΑ = 0.24

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: τιμή, τελική_τιμή
ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε τιμή προϊόντος'

ΔΙΑΒΑΣΕ τιμή

τελική_τιμή < - τιμή + τιμή * ΦΠΑ

ΓΡΑΨΕ 'τελική τιμή: ', τελική_τιμή
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Χωρίς να μιλήσουμε για το συντακτικό,
μπορείτε να μαντέψετε τι κάνει το παραπάνω
πρόγραμμα;

- Στο διπλανό παράδειγμα βλέπουμε ένα απλό πρόγραμμα
- Θα αναλύσουμε τα επιμέρους στοιχεία, που χρησιμεύουν, πως ονομάζονται
- Γενικότερα, θα αναλύσουμε το συντακτικό της ψευδογλώσσας, ώστε να υλοποιούμε κατάλληλους αλγορίθμους για την επίλυση διαφόρων υπολογιστικών προβλημάτων

03.1

ΜΕΤΑΒΛΗΤΕΣ

Μια μεταβλητή είναι ένα γλωσσικό αντικείμενο που χρησιμοποιείται για να αναπαραστήσει ένα στοιχείο δεδομένου. Το περιεχόμενο μιας μεταβλητής μπορεί να τροποποιείται κατά την διάρκεια εκτέλεσης ενός αλγορίθμου. (αυτό σημαίνει πως μια μεταβλητή μπορεί να αλλάζει τιμή αλλά **όχι τον τύπο της**). Ο τύπος μιας μεταβλητής καθορίζεται από το περιεχόμενο της.

ΕΙΔΟΣ	ΤΥΠΟΣ	ΠΑΡΑΔΕΙΓΜΑ
ΑΡΙΘΜΗΤΙΚΕΣ	ΠΡΑΓΜΑΤΙΚΕΣ	5.7
	ΑΚΕΡΑΙΕΣ	5
ΑΛΦΑΡΙΘΜΗΤΙΚΕΣ	ΧΑΡΑΚΤΗΡΕΣ	“Κώστας” “ΑΛΗΘΗΣ” “5@34AK”
ΛΟΓΙΚΕΣ	ΛΟΓΙΚΕΣ	ΑΛΗΘΗΣ ΨΕΥΔΗΣ 3 > 5 19 <> 0

ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΡΟΓΡΑΜΜΑ Υπολογισμός
ΣΤΑΘΕΡΕΣ

ΦΠΑ = 0.24

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: τιμή, τελική_τιμή
ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε τιμή προϊόντος'

ΔΙΑΒΑΣΕ τιμή

τελική_τιμή < - τιμή + τιμή * ΦΠΑ

ΓΡΑΨΕ 'τελική τιμή: ', τελική_τιμή
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- Τι είναι το τιμή και το τελική_τιμή;
- Τι τύπου μεταβλητές είναι;

Για να χρησιμοποιήσουμε κάποια μεταβλητή σε κάποια έκφραση ή εντολή εξόδου θα πρέπει απαραίτητα να της έχει αποδοθεί τιμή προηγουμένως

03.2

ΣΤΑΘΕΡΕΣ

Οι σταθερές είναι προκαθορισμένες τιμές που δεν τροποποιούνται κατά την διάρκεια εκτέλεσης ενός αλγορίθμου. Συμβολική σταθερά ονομάζεται το όνομα της και σταθερά ονομάζεται η τιμή που αποθηκεύεται σε αυτήν

ΠΑΡΑΔΕΙΓΜΑ ΜΕ ΣΤΑΘΕΡΑ

ΠΡΟΓΡΑΜΜΑ Υπολογισμός

ΣΤΑΘΕΡΕΣ

$\Phi\text{ΠΑ} = 0.24$

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: τιμή

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε τιμή προϊόντος'

ΔΙΑΒΑΣΕ τιμή

τελική_τιμή $< -$ τιμή $+$ τιμή $*$ $\Phi\text{ΠΑ}$

ΓΡΑΨΕ 'τελική τιμή: ', τελική_τιμή

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- Εφόσον χρησιμοποιούμε σταθερά ή σταθερές, αυτές δηλώνονται πάντα στην αρχή του προγράμματος πάνω από τις μεταβλητές
- Για τις σταθερές δεν δηλώνεται ο τύπος τους όπως στην περίπτωση των μεταβλητών
- Οι σταθερές σε αντίθεση με τις μεταβλητές ΔΕΝ ΜΠΟΡΟΥΝ ΝΑ ΑΛΛΑΞΟΥΝ ΤΙΜΗ

03.3

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ

Σε αυτήν την ενότητα θα εξηγήσουμε τις βασικές εντολές που έχουμε διαθέσιμες για την υλοποίηση αλγορίθμων είτε με ψευδογλώσσα είτε με πρόγραμμα

	ΨΕΥΔΟΓΛΩΣΣΑ	ΠΡΟΓΡΑΜΜΑ	ΕΠΕΞΗΓΗΣΗ	ΠΑΡΑΔΕΙΓΜΑ
ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ	Διάβασε	ΔΙΑΒΑΣΕ	Κατά την εκτέλεση της εντολής ΔΙΑΒΑΣΕ διακόπτεται η εκτέλεση και αναμένει τιμές από τον χρήστη για να τις αποθηκεύσει στις μεταβλητές που βρίσκονται μετά το ΔΙΑΒΑΣΕ	ΔΙΑΒΑΣΕ μεταβλητή1, μεταβλητή2, ..., μεταβλητήN
ΕΝΤΟΛΕΣ ΕΞΟΔΟΥ	Εκτύπωσε, Εμφάνισε	ΓΡΑΨΕ	Ότι βρίσκεται μέσα στα '...' το εμφανίζει ακριβώς όπως είναι ως κείμενο και για ότι δεν είναι μέσα σε '...' εμφανίζει την αξία του(περιεχόμενο μεταβλητής)	ΓΡΑΨΕ 'Χρήστη δώσε έναν αριθμό' ΓΡΑΨΕ αριθμός ΓΡΑΨΕ 'Ο αριθμός είναι', number
ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ	←	<	Το αποτέλεσμα της έκφρασης δεξιά αποθηκεύεται (καταχωρίζεται / εκχωρείται) στην μεταβλητή αριστερά. Προσοχή δεν το μπερδεύουμε με ισότητα. Η έκφραση και η μεταβλητή πρέπει να είναι του ίδιου τύπου	$x \leftarrow 3$ $x \leftarrow x + 2$

03.4

ΤΕΛΕΣΤΕΣ

Σε αυτήν την ενότητα θα εξετάσουμε τους τελεστές που έχουμε στη διάθεση μας για την δημιουργία εκφράσεων

ΑΡΙΘΜΗΤΙΚΟΙ	ΠΡΑΞΗ	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ
+	Πρόσθεση	$59 + 38$
-	Αφαίρεση	$59 - 38$
*	Πολ/σμός	$59 * 38$
/	Διαίρεση	$59 / 38$
^	Δύναμη	$2 ^ 2$
div	Πηλίκο ακέραιας διαίρεσης	$9 \text{ div } 2 (4)$
mod	Υπόλοιπο ακέραιας διαίρεσης	$9 \text{ mod } 2 (1)$

1. ΠΑΡΕΝΘΕΣΕΙΣ (ορίζουμε εμείς την ιεραρχία των πράξεων)
2. ^ (ύψωση σε δύναμη)
3. * / mod div
4. + -

Όταν έχουμε πράξεις που ανήκουν στην ίδια ιεραρχία, τότε η σειρά εκτέλεσης τους πραγματοποιείται από αριστερά προς τα δεξιά

ΣΥΓΚΡΙΤΙΚΟΙ	ΛΟΓΙΚΗ ΠΡΑΞΗ	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ
>	Μεγαλύτερο	$59 > 38$
<	Μικρότερο	$59 < 38$
>=	Μεγαλύτερο ή ίσο	$5 >= 5$
<=	Μικρότερο ή ίσο	$59 <= 38$
<>	Διάφορο	$2 <> 2$

Οι λογικές εκφράσεις επιστρέφουν **ΠΑΝΤΑ** ως αποτέλεσμα λογική τιμή (ΑΛΗΘΕΣ ή ΨΕΥΔΕΣ)

ΛΟΓΙΚΟΙ	ΛΟΓΙΚΗ ΠΡΑΞΗ	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ
ΚΑΙ	Σύζευξη	λογπρ1 ΚΑΙ λογπρ2
Η	Διάζευξη	λογπρ1 Η λογπρ2
ΟΧΙ	Άρνηση	ΟΧΙ λογπρ

Από τους 3 λογικούς τελεστές, οι τελεστές ΚΑΙ / Η χρησιμοποιούνται για τη σύνδεση λογικών συνθηκών. Οι λογικοί τελεστές επιστρέφουν όπως και οι συγκριτικοί τελεστές πάντα λογικές τιμές ως αποτέλεσμα

- ❖ Ο τελεστής ΚΑΙ αποδίδει την τιμή αληθής, μόνο όταν και οι 2 (ή περισσότερες) λογικές συνθήκες που συνδέει είναι αληθείς
- ❖ Ο τελεστής Η αποδίδει την τιμή αληθής, όταν έστω 1 από τις 2 (ή περισσότερες) λογικές συνθήκες που συνδέει είναι αληθής
- ❖ Ο τελεστής ΟΧΙ προσκολλάται στην λογική συνθήκη η οποία ακολουθεί και την αντιστρέφει.

Όταν μιλάμε για αντιστροφή λογικής συνθήκης, αρκεί να αντιστρέψουμε τους συγκριτικούς τελεστές (το $>$ να γίνει $\leq$, το $=$ να γίνει $\neq$ κλπ.) Για παράδειγμα η λογική συνθήκη $\alpha > 3$ θα γίνει $\alpha \leq 3$ και η λογική συνθήκη $\beta = 0$ θα γίνει $\beta \neq 0$

Όταν η συνθήκη είναι σύνθετη (περιέχει και λογικούς τελεστές), θα πρέπει να αντιστραφούν και αυτοί (το και να γίνει η και το η θα γίνει και). Για παράδειγμα, ή συνθήκη $(\rho = 2)$ ή $(\delta \leq 4)$ θα γίνει $(\rho \neq 2)$ και $(\delta > 4)$

ΙΕΡΑΡΧΙΑ ΤΕΛΕΣΤΩΝ

Α. Αριθμητικοί	1. ^
	2. * , / , div , mod
	3. + , -
Β. Συγκριτικοί	1. > , < , = , <> , >= , <=
Γ. Λογικοί	1. ΟΧΙ
	2. ΚΑΙ
	3. Η

03.5

ΣΥΝΑΡΤΗΣΕΙΣ

Σε αυτή την ενότητα θα εξετάσουμε κάποιες συναρτήσεις που είναι διαθέσιμες μόνο όταν υλοποιούμε κωδικοποίηση με πρόγραμμα

ΣΥΝΑΡΤΗΣΗ	ΠΕΡΙΓΡΑΦΗ	ΠΑΡΑΔΕΙΓΜΑ
T_P(X)	Υπολογισμός τετραγωνικής ρίζας X	T_P(4)
A_T(X)	Απόλυτη τιμή του X	A_Π(-20+3)
A_M(X)	Ακέραιο μέρος του X	A_M(1.32)
HM(X)	Ημίτονο του X	HM(90)
ΣΥΝ(X)	Συνημίτονο του X	ΣΥΝ(90)
ΕΦ(X)	Εφαπτομένη του X	ΕΦ(90)
ΛΟΓ(X)	Λογάριθμος του X	ΛΟΓ(5)
E(X)	e^x	E(2)

A series of thin, dark, wavy lines that create a sense of motion and depth, flowing across the bottom of the slide.

04

ΑΛΓΟΡΙΘΜΙΚΕΣ ΔΟΜΕΣ

Τι είναι οι αλγοριθμικές δομές;

- Δώσαμε τον ορισμό του αλγορίθμου και που στοχεύει
- Εξετάσαμε τους 4 τρόπους αναπαράστασης ενός αλγορίθμου και αναφέραμε πως ο αποδοτικότερος τρόπος είναι η **κωδικοποίηση (με ψευδοκώδικα)**

Στο σημείο αυτό θα επικεντρωθούμε στην αναπαράσταση ενός αλγορίθμου με κωδικοποίηση(αλγόριθμο σε ψευδογλώσσα / πρόγραμμα). Σε κάθε περίπτωση όταν γράφουμε ψευδοκώδικα, έχουμε 3 βασικές αλγοριθμικές δομές που μπορούμε να χρησιμοποιήσουμε, ανάλογα την περίπτωση και τα ζητούμενα του προβλήματος

ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ

ΔΟΜΗ ΕΠΙΛΟΓΗΣ

ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

04.1

ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ

Αρχικά θα εξετάσουμε την **Δομή ακολουθίας**.

Η ακολουθιακή δομή εντολών χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών

“

ΠΑΡΑΔΕΙΓΜΑ ΑΚΟΛΟΥΘΙΑΚΗΣ ΔΟΜΗΣ

Να γραφεί αλγόριθμος σε ψευδογλώσσα που θα δέχεται 3 βαθμούς και θα εμφανίζει τον μέσο όρο τους. Στην συνέχεια δημιουργήστε το διάγραμμα ροής του αλγορίθμου.

”

ΕΠΙΛΥΣΗ (ΨΕΥΔΟΓΛΩΣΣΑ)

Αλγόριθμος υπολογισμός

Εμφάνισε “Δώσε 3 βαθμούς”

Διάβασε $\beta_1, \beta_2, \beta_3$

$\mu_o \leftarrow (\beta_1 + \beta_2 + \beta_3) / 3$

Εμφάνισε “Ο μέσος όρος είναι: ”, μ_o

Τέλος υπολογισμός

	ΨΕΥΔΟΓΛΩΣΣΑ	ΠΡΟΓΡΑΜΜΑ
ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ	Διάβασε	ΔΙΑΒΑΣΕ
ΕΝΤΟΛΕΣ ΕΞΟΔΟΥ	Εκτύπωσε, Εμφάνισε	ΓΡΑΨΕ
ΕΙΣΑΓΩΓΙΚΑ	Διπλά	Μονά
ΔΕΣΜΕΥΜΕΝΕΣ ΛΕΞΕΙΣ	Πεζά	Κεφαλαία
ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ	←	<
ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ		ΥΠΟΧΡΕΩΤΙΚΗ

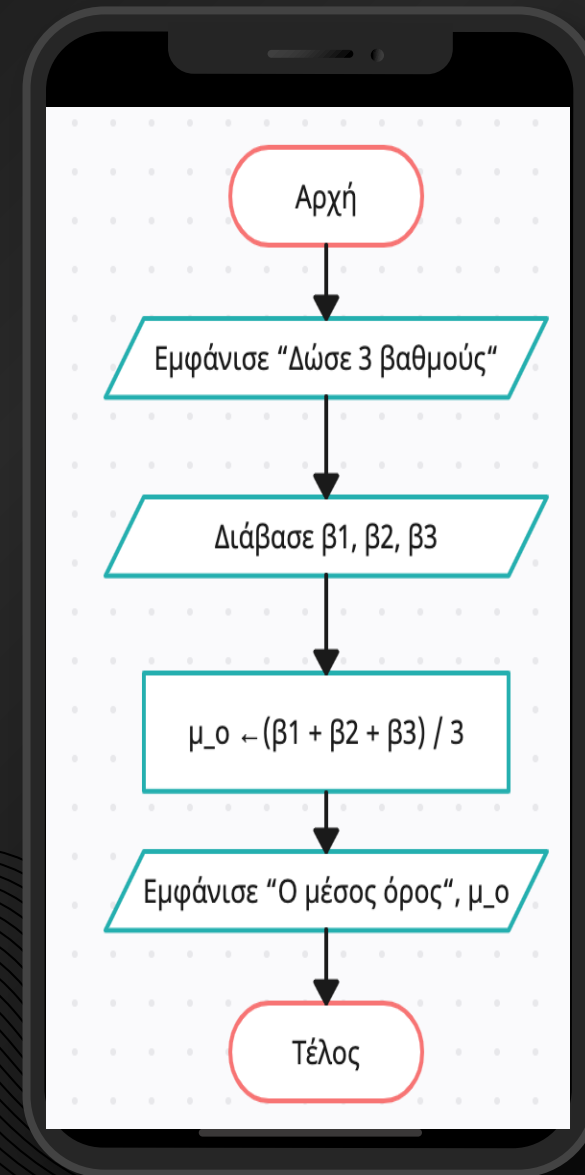
ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

ΕΙΣΑΓΩΓΗ /
ΕΞΟΔΟΣ
ΔΕΔΟΜΕΝΩΝ

ΕΝΤΟΛΗ

ΑΡΧΗ / ΤΕΛΟΣ

ΣΥΝΘΗΚΗ



ΕΠΙΛΥΣΗ (ΠΡΟΓΡΑΜΜΑ)

ΠΡΟΓΡΑΜΜΑ Υπολογισμός
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: $\beta_1, \beta_2, \beta_3$

ΠΡΑΓΜΑΤΙΚΕΣ: μ_o

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε 3 βαθμούς'

ΔΙΑΒΑΣΕ $\beta_1, \beta_2, \beta_3$

$\mu_o < -(\beta_1 + \beta_2 + \beta_3) / 3$

ΓΡΑΨΕ 'Ο μέσος όρος είναι: ', μ_o

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

	ΨΕΥΔΟΓΛΩΣΣΑ	ΠΡΟΓΡΑΜΜΑ
ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ	Διάβασε	ΔΙΑΒΑΣΕ
ΕΝΤΟΛΕΣ ΕΞΟΔΟΥ	Εκτύπωσε, Εμφάνισε	ΓΡΑΨΕ
ΕΙΣΑΓΩΓΙΚΑ	Διπλά	Μονά
ΔΕΣΜΕΥΜΕΝΕΣ ΛΕΞΕΙΣ	Πεζά	Κεφαλαία
ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ	←	<
ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ		ΥΠΟΧΡΕΩΤΙΚΗ

05

ΣΩΣΤΟ / ΛΑΘΟΣ

Η δομή ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση περίπλοκων προβλημάτων

Λ

Το πλάγιο παραλληλόγραμμο χρησιμοποιείται για την είσοδο / έξοδο δεδομένων σε ένα διάγραμμα ροής

Σ

Το διάγραμμα ροής είναι ένας τρόπος περιγραφής αλγορίθμου

Σ

Σε ένα διάγραμμα ροής ο ρόμβος δηλώνει την αρχή και το τέλος του αλγορίθμου

Λ

Τα είδη των μεταβλητών που χρησιμοποιούμε είναι οι αριθμητικές, οι αλφαριθμητικές και οι σταθερές

Λ

Η τιμή μιας μεταβλητής δεν μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου

Λ

Μια μεταβλητή μπορεί να αποθηκεύσει αλφαριθμητικά δεδομένα

Σ

Μια μεταβλητή μπορεί να αλλάξει τιμή και τύπο δεδομένων κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου

Λ

Ενώ η τιμή μιας μεταβλητής μπορεί να αλλάξει κατά την εκτέλεση του προγράμματος, αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της

Σ

Οι λογικές μεταβλητές λαμβάνουν υποχρεωτικά είτε την τιμή ΑΛΗΘΗΣ είτε την τιμή ΨΕΥΔΗΣ

Σ

Μια έκφραση μπορεί να περιλαμβάνει μεταβλητές, σταθερές, τελεστές και παρενθέσεις

Σ

Στην αριθμητική έκφραση $A + B * \Gamma$ εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός

Λ

Μια σταθερά μπορεί να αλλάξει τιμή κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου

Λ

Σε μια εντολή εκχώρησης του αποτελέσματος μιας έκφρασης σε μια μεταβλητή, η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου

Λ

Σε μια εντολή εκχώρησης είναι δυνατόν μια έκφραση στο δεξί μέλος να περιέχει τη μεταβλητή που βρίσκεται στο αριστερό μέλος

Σ

Η εντολή εκχώρησης τιμής αποδίδει το αποτέλεσμα μιας έκφρασης σε μια μεταβλητή

Σ

Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή εισόδου Διάβασε

Λ

Η εντολή $X \leftarrow X * X$ είναι έγκυρη

Σ

Ο αριθμητικός τελεστής mod έχει τη μεγαλύτερη ιεραρχία από τους αριθμητικούς τελεστές

Λ

Ο τελεστής Mod χρησιμοποιείται για τον υπολογισμό του ακέραιου πηλίκου μιας διαίρεσης ακέραιων αριθμών

Λ

Στη δομή ακολουθίας εκτελούνται όλες οι εντολές του αλγορίθμου

Σ

Για κάθε ακέραιο θετικό αριθμό X ισχύει πως το $X \text{ div } 2$ είναι ίσο με το $A_M(X / 2)$

Σ

Στην έκφραση $X \leftarrow Z \text{ div } Y$ μπορεί κάποιο από τα X, Z, Y να είναι πραγματικός αριθμός

Λ

Ένα πρόγραμμα σε ΓΛΩΣΣΑ αποτελεί ουσιαστικά την υλοποίηση ενός αλγορίθμου σε κωδικοποίηση

Σ

Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα αντιστοιχίζονται από τον μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του Η/Υ

Σ

Η δήλωση των μεταβλητών που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική

Σ

Οι τύποι των μεταβλητών που υποστηρίζει η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ, ΑΚΕΡΑΙΕΣ και ΧΑΡΑΚΤΗΡΕΣ

Λ

Κατά την εκτέλεση του προγράμματος, η εντολή ΔΙΑΒΑΣΕ διακόπτει την εκτέλεση του και περιμένει την εισαγωγή τιμών από το πληκτρολόγιο

Σ

Η δομή ενός προγράμματος σε ΓΛΩΣΣΑ είναι αυστηρά καθορισμένη

Σ

Οι δεσμευμένες λέξεις της ΓΛΩΣΣΑΣ δεν μπορούν να χρησιμοποιηθούν ως ονόματα μεταβλητών σε ένα πρόγραμμα

Σ

Στη ΓΛΩΣΣΑ, ο μέσος όρος ενός συνόλου ακέραιων μεταβλητών πρέπει να αποθηκευτεί σε μια πραγματική μεταβλητή

Σ

Ο τύπος μιας μεταβλητής μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης ενός προγράμματος

Λ

Κάθε πρόγραμμα γραμμένο σε ΓΛΩΣΣΑ περιλαμβάνει οπωσδήποτε τμήμα δήλωσης σταθερών

Λ

Η ακολουθιακή δομή εντολών χρησιμοποιείται όταν είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών

Σ

Απαγορεύεται η αλλαγή τιμής των συμβολικών σταθερών

Σ

Κάθε μεταβλητή παίρνει τιμή μόνο με εντολή εκχώρησης

Λ

Οι τελεστές mod και div μπορούν να εφαρμοστούν μόνο σε ακέραιους τελεστέους

Σ

Η μεταβλητή X είναι ακέραιου τύπου στην εντολή εκχώρησης $X \leftarrow A_M(a) / 2$

Λ

Σε ένα διάγραμμα ροής το ορθογώνιο παριστάνει έναν υπολογισμό

Σ

Η έκφραση $T_P(X)$ μπορεί να παραβιάζει το κριτήριο της αποτελεσματικότητας αν το X είναι αρνητικό

Λ

Οι τελεστές div και mod μπορούν να οδηγήσουν στην παραβίαση του κριτηρίου της καθοριστικότητας

Σ

Η πράξη $A \bmod 3$ δεν ορίζεται αν το A είναι πραγματικός αριθμός

Σ

Μια εντολή εκχώρησης δεν μπορεί να εκτελεστεί αν δεν έχουν λάβει τιμή οι μεταβλητές που συμμετέχουν σε αυτήν

Σ

Σε ένα διάγραμμα ροής, το σχήμα της έλλειψης μπορεί να εμφανίζεται περισσότερες από 2 φορές

Λ

Στην κωδικοποίηση, μπορεί να χρησιμοποιείται μια μεταβλητή σε υπολογισμούς χωρίς ποτέ να χρησιμοποιείται σε εντολή εξόδου

Σ

Η εντολή εξόδου μπορεί να συνοδεύεται από μεταβλητές, σταθερές ή εκφράσεις

Σ

Στην έκφραση $a * b ^ 2 / 4$, οι πράξεις εκτελούνται από αριστερά προς τα δεξιά

Λ

Η συνάρτηση A_T μπορεί να εφαρμοστεί μόνο σε δεδομένα ακέραιου τύπου

Λ

06

**ΣΥΝΤΟΜΕΣ
ΑΣΚΗΣΕΙΣ**

ΣΥΜΠΛΗΡΩΣΤΕ ΤΟΝ ΠΙΝΑΚΑ

ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ	ΠΕΡΙΕΧΟΜΕΝΟ ΜΕΤΑΒΛΗΤΗΣ Χ	ΤΥΠΟΣ ΜΕΤΑΒΛΗΤΗΣ Χ
$X \leftarrow \text{'ΨΕΥΔΗΣ'}$	‘ΨΕΥΔΗΣ’	ΑΛΦΑΡΙΘΜΗΤΙΚΗ / ΧΑΡΑΚΤΗΡΕΣ
$X \leftarrow 11 - 3$	2	ΑΚΕΡΑΙΑ
$X \leftarrow 11.0 - 3.0$	2.0	ΠΡΑΓΜΑΤΙΚΗ
$X \leftarrow \text{ΑΛΗΘΗΣ}$	ΑΛΗΘΗΣ	ΛΟΓΙΚΗ
$X \leftarrow 4$	4	ΑΚΕΡΑΙΑ
$X \leftarrow 10 > 5$	ΑΛΗΘΗΣ	ΛΟΓΙΚΗ
$X \leftarrow 35.0 < 40.0$	ΑΛΗΘΗΣ	ΛΟΓΙΚΗ
$X \leftarrow 20 > 10 \text{ ΚΑΙ } 25 < 26$	ΑΛΗΘΗΣ	ΛΟΓΙΚΗ

ΥΠΟΛΟΓΙΣΤΕ

- ΤΗΝ ΤΙΜΗ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ Y ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΠΑΡΑΚΑΤΩ ΕΝΤΟΛΗΣ

$Y \leftarrow 5 * (X - 3) + X \wedge 3 - 2 + Z$, όταν $X = 5$ και $Z = 1$

134

- ΤΗΝ ΤΙΜΗ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ X ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΠΑΡΑΚΑΤΩ ΕΝΤΟΛΗΣ

$X \leftarrow 5 + 7 \text{ DIV } 3 * 4 - 10 \text{ MOD } 3$

12

ΓΡΑΨΤΕ ΤΙΣ ΠΑΡΑΚΑΤΩ ΜΑΘΗΜΑΤΙΚΕΣ ΕΚΦΡΑΣΕΙΣ ΣΕ ΓΛΩΣΣΑ

$$\frac{5X - 3Y}{A - B^2}$$

$$(5*X - 3*Y) / (A - B^2)$$

$$\sqrt{X^2 - Y^2}$$

$$T_P(X^2 - Y^2) \text{ ή } (X^2 - Y^2)^{1/2}$$

$$2X + \frac{3(X + 1)}{Y^2 + 1} e^X$$

$$2*X + (3*(X+1)) / (Y^2 + 1)*E(x)$$

$$\frac{5^{\frac{K}{3}} + \sqrt[3]{a}}{(a - b)^{\frac{2}{k}} + 2}$$

$$(5^{(K/3)} + a^{(1/3)}) / ((a-b)^{(2/k)} + 2)$$

Εφόσον τα A, B είναι ακέραιες μεταβλητές, συμπληρώστε στον παρακάτω πίνακα τι τύπος μεταβλητής θα είναι το K στις παρακάτω περιπτώσεις

ΕΝΤΟΛΗ	ΤΥΠΟΣ ΜΕΤΑΒΛΗΤΗΣ
$K \leftarrow (A+B) / 2$	ΠΡΑΓΜΑΤΙΚΗ
$K \leftarrow (A+B) \text{ DIV } 2$	ΑΚΕΡΑΙΑ
$K \leftarrow '12345'$	ΧΑΡΑΚΤΗΡΕΣ
$K \leftarrow '2017 - 2'$	ΧΑΡΑΚΤΗΡΕΣ
$K \leftarrow A_M(A+B) / 2$	ΠΡΑΓΜΑΤΙΚΗ
$K \leftarrow 5 ^ { (1/3)}$	ΠΡΑΓΜΑΤΙΚΗ
$K \leftarrow 35*B > Y+Z$	ΛΟΓΙΚΗ
$K \leftarrow B \text{ MOD } 3$	ΑΚΕΡΑΙΑ
$K \leftarrow B / 3$	ΠΡΑΓΜΑΤΙΚΗ
$K \leftarrow T_P(A_M(2345.67))$	ΠΡΑΓΜΑΤΙΚΗ

ΕΝΤΟΠΙΣΤΕ ΤΑ ΛΑΘΗ ΣΤΟΥΣ ΠΑΡΑΚΑΤΩ ΑΛΓΟΡΙΘΜΟΥΣ

Αλγόριθμος exercise3

Διάβασε α, β

$z \leftarrow \alpha \& \beta$

Εκτύπωσε z

Τέλος exercise1

Το & δεν είναι κάποιος τελεστής
κατανοητός από τον
μεταγλωττιστή της γλώσσας και
έτσι παραβιάζεται το κριτήριο της
αποτελεσματικότητας

Αλγόριθμος exercise2

Διάβασε $\alpha \leftarrow 3$

$\gamma \leftarrow \alpha - 40$

name $\leftarrow$ "Βούλης"

average $\leftarrow$ T_P(20 / α)

Εκτύπωσε "μ.ο", average

Τέλος exercise1

Με την εντολή Διάβασε δεν γίνεται
εκχώρηση

1. Δεν εξασφαλίζεται ότι το α είναι
διάφορο του 0, παραβιάζοντας το
κριτήριο της καθοριστικότητας
2. Ταυτόχρονα δεν εξασφαλίζεται
πως το υπόρριζο δεν μεγαλύτερο
ή ίσο του 0, παραβιάζοντας πάλι
το κριτήριο της καθοριστικότητας

Αλγόριθμος exercise1

Διάβασε α

$z \leftarrow 2 * \alpha + \beta$

Εκτύπωσε z

Τέλος exercise1

Η μεταβλητή β δεν έχει λάβει
προηγούμενως τιμή

Αλγόριθμος exercise

$a \leftarrow 3$

$b \leftarrow a + 14$

$c \leftarrow a * b - 20$

$a \leftarrow (c - a) \text{ div } 3$

$b \leftarrow b \bmod a$

$c \leftarrow c - (a + b)$

Εμφάνισε a, b, c

Τέλος exercise

ΣΥΜΠΛΗΡΩΣΤΕ ΤΟΝ ΠΙΝΑΚΑ ΤΙΜΩΝ

Δίνεται ο διπλανός αλγόριθμος. Παρουσιάστε τον πίνακα τιμών των μεταβλητών και τις τιμές που θα εμφανιστούν

a	b	c
3		
	17	
		31
9		
	8	
		14
Θα εμφανιστεί: 9, 8, 14		

07

ΑΣΚΗΣΕΙΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ

“

ΑΣΚΗΣΗ 1

Ο νόμος του Νεύτωνα για τη βαρύτητα λέει ότι κάθε σώμα στο σύμπαν έλκει κάθε άλλο σώμα με δύναμη που δίνεται από τον τύπο:

$$F = G \cdot \frac{m1 \cdot m2}{r^2}$$

όπου $m1$ και $m2$ είναι οι μάζες των 2 σωμάτων(σε κιλά) και r η απόσταση μεταξύ τους (σε μέτρα) και G είναι η παγκόσμια βαρυτική σταθερά. Να αναπτυχθεί πρόγραμμα που θα διαβάσει τις 2 μάζες, την απόσταση μεταξύ τους και θα υπολογίζει και θα εκτυπώνει τη δύναμη. Δίνεται η βαρυτική σταθερά G

$$G = 6.67 \times 10^{-11}$$

”

ΕΠΙΛΥΣΗ ΑΣΚΗΣΗΣ 1

```
1 ΠΡΟΓΡΑΜΜΑ GravityForce
2 ΣΤΑΘΕΡΕΣ
3   G = 6.67 * 10^(-11)
4 ΜΕΤΑΒΛΗΤΕΣ
5
6 ΑΡΧΗ
7
8
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Βήμα 1°

Δημιουργούμε την δομή ενός προγράμματος. Εφόσον υπάρχουν σταθερές τιμές, αυτές τις δηλώνουμε πάνω από το τμήμα δήλωσης των μεταβλητών. Τις μεταβλητές τις δηλώνουμε πάντα στο τέλος και αφού έχουμε ολοκληρώσει την κωδικοποίηση του προγράμματος

```
1 ΠΡΟΓΡΑΜΜΑ GravityForce
2 ΣΤΑΘΕΡΕΣ
3   G = 6.67 * 10^(-11)
4 ΜΕΤΑΒΛΗΤΕΣ
5   ΠΡΑΓΜΑΤΙΚΕΣ: m1, m2, r, F
6
7 ΑΡΧΗ
8   ΓΡΑΨΕ 'Δώσε μάζα 1'
9   ΔΙΑΒΑΣΕ m1
10  ΓΡΑΨΕ 'Δώσε μάζα 2'
11  ΔΙΑΒΑΣΕ m2
12  ΓΡΑΨΕ 'Δώσε απόσταση'
13  ΔΙΑΒΑΣΕ r
14  F <- G * (m1 * m2) / r ^ 2
15  ΓΡΑΨΕ 'Η βαρυτική δύναμη μεταξύ των 2 σωμάτων είναι', F
16 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Βήμα 3°

Δηλώνουμε τις μεταβλητές στο αντίστοιχο τμήμα δήλωσης τους

Βήμα 2°

Έχουμε κατανοήσει το πρόβλημα (σωστή διατύπωση, σωστή ερμηνεία και στη συνέχεια καταγραφή των δεδομένων και των ζητούμενων του) και αφού στο στάδιο της ανάλυσης δεν χρειάζεται να "σπάσουμε" το πρόβλημα σε μικρότερα λόγω της απλότητάς του, προχωράμε στην επίλυση (αναπαράσταση του αλγορίθμου με κωδικοποίηση σε πρόγραμμα)