

ΠΛΗΡΟΦΟΡΙΚΗ

ΤΜΗΜΑΤΙΚΌΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΌΣ



10.1 ΤΜΗΜΑΤΙΚΌΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΌΣ

- **Τμηματικός προγραμματισμός** ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων
- Όταν ένα τμήμα προγράμματος επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα, τότε αναφερόμαστε σε **υποπρόγραμμα (subprogram)**.

10.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

- **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο.** Στην πραγματικότητα κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό που γίνεται πάντοτε από την αρχή του, εκτελεί ορισμένες ενέργειες, και απενεργοποιείται με την έξοδο από αυτό που γίνεται πάντοτε από το τέλος του.
- **Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.** Αυτό σημαίνει ότι κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα. Στην πράξη βέβαια η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.
- **Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο.** Η έννοια του μεγάλου προγράμματος είναι υποκειμενική, αλλά πρέπει κάθε υποπρόγραμμα να είναι τόσο, ώστε να είναι εύκολα κατανοητό για να μπορεί να ελέγχεται. Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

10.3 ΠΛΕΟΝΕΚΤ'ΗΜΑΤΑ ΤΟΥ ΤΜΗΜΑΤΙΚΟ'Υ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ'Υ

Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος

- Επιτρέπει την εξέταση και την επίλυση απλών προβλημάτων και όχι στην αντιμετώπιση του συνολικού προβλήματος. Με τη σταδιακή επίλυση των υποπροβλημάτων και τη δημιουργία των αντιστοίχων υποπρογραμμάτων τελικά επιλύεται το συνολικό πρόβλημα.

Διευκολύνει την κατανόηση και διόρθωση του προγράμματος

- Ο χωρισμός του προγράμματος σε μικρότερα αυτοτελή τμήματα επιτρέπει τη γρήγορη διόρθωση ενός συγκεκριμένου τμήματός του χωρίς οι αλλαγές αυτές να επηρεάσουν όλο το υπόλοιπο πρόγραμμα. Επίσης διευκολύνει οποιονδήποτε χρειαστεί να διαβάσει και να κατανοήσει τον τρόπο που λειτουργεί το πρόγραμμα. Όπως έχει πολλές φορές τονιστεί αυτό είναι πολύ σημαντικό χαρακτηριστικό του σωστού προγραμματισμού, αφού ένα μεγάλο πρόγραμμα στον κύκλο της ζωής του χρειάζεται να συντηρηθεί από διαφορετικούς προγραμματιστές.

Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος

- Πολύ συχνά χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία ενός προγράμματος. Από τη στιγμή που ένα υποπρόγραμμα έχει γραφεί, μπορεί το ίδιο να καλείται από πολλά σημεία του προγράμματος. Έτσι μειώνονται το μέγεθος του προγράμματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται πιο εύληπτο και κατανοητό.

Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού

- Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πολύ εύκολα και σε άλλα προγράμματα. Από τη στιγμή που έχει δημιουργηθεί, η χρήση του δεν διαφέρει από τη χρήση των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού, όπως για τον υπολογισμό του ημίτονου ή του συνημίτονου ή την εντολή με την οποία εκτελεί μία συγκεκριμένη διαδικασία, για παράδειγμα γράφει στην οθόνη (εντολή ΓΡΑΨΕ). Αν λοιπόν χρειάζεται συχνά κάποια λειτουργία που δεν υποστηρίζεται απευθείας από τη γλώσσα, όπως για παράδειγμα η εύρεση του μικρότερου δύο αριθμών, τότε μπορεί να γραφεί το αντίστοιχο υποπρόγραμμα. Η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά, ουσιαστικά επεκτείνουν την ίδια τη γλώσσα προγραμματισμού.

10.4 ΠΑΡΆΜΕΤΡΟΙ

- Μία **παράμετρος** είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.

Σημείωση

- Όσον αφορά στις συναρτήσεις, το πέρασμα αυτό γίνεται από το τμήμα προγράμματος που καλεί τη Συνάρτηση προς τη Συνάρτηση και όχι το αντίθετο. Η Συνάρτηση μπορεί να περάσει/επιστρέψει τιμή μόνο χρησιμοποιώντας το όνομα της ως μεταβλητή (σε εντολή εκχώρησης). Αυτό σημαίνει πως σε μία Συνάρτηση δεν επιχειρώ να τροποήσω τις παραμέτρους της. Δεν απαγορεύεται, άρα δε θα οδηγήσει σε σφάλμα, αλλά σίγουρα είναι κακή πρακτική. Αν τροποποιήσω τις τιμές των τυπικών παραμέτρων, η αλλαγή αυτή δεν θα έχει αντίκτυπο στο τμήμα προγράμματος που κάλεσε τη Συνάρτηση, άρα δεν θα τροποποιηθούν οι πραγματικές παράμετροι.
- Από την άλλη, σε μια Διαδικασία μπορώ να τροποποιήσω τις τιμές των πραγματικών παραμέτρων, με τις οποίες κλήθηκε η Διαδικασία, τροποποιώντας τις αντίστοιχες τυπικές παραμέτρους. Όταν μία διαδικασία τερματίζει, οι τιμές των τυπικών παραμέτρων αντιγράφονται πίσω στις αντίστοιχες πραγματικές.
- Η διαδικασία αυτή είναι γνωστή ως `call-by-copy-restore` ή `value-result` ή `copy-in-copy-out`.

10.5 ΔΙΑΔΙΚΑΣΊΕΣ ΚΑΙ ΣΥΝΑΡΤΉΣΕΙΣ

- Η **συνάρτηση** είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).
- Η **διαδικασία** είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος.

10.5.1 ΟΡΙΣΜΌΣ ΚΑΙ ΚΛΉΣΗ ΣΥΝΑΡΤΉΣΕΩΝ

- Κάθε συνάρτηση έχει την ακόλουθη δομή:

ΣΥΝΑΡΤΗΣΗ Όνομα (<λίστα παραμέτρων>) : <τύπος συνάρτησης>

<Τμήμα δηλώσεων>

ΑΡΧΗ

...

Όνομα <- έκφραση

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

- Το όνομα της συνάρτησης είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ.
- Η λίστα παραμέτρων είναι μια λίστα μεταβλητών, των οποίων οι τιμές μεταβιβάζονται στη συνάρτηση κατά την κλήση.
- Οι συναρτήσεις μπορούν να επιστρέφουν τιμές όλων των τύπων δεδομένων που υποστηρίζει η γλώσσα.
- Μια συνάρτηση λοιπόν μπορεί να είναι ΠΡΑΓΜΑΤΙΚΗ, ΑΚΕΡΑΙΑ, ΧΑΡΑΚΤΗΡΑΣ, ΛΟΓΙΚΗ.
- Στις εντολές του σώματος της συνάρτησης πρέπει υποχρεωτικά να υπάρχει μία εντολή εκχώρησης τιμής στο όνομα της συνάρτησης, π.χ. Εμβαδό_κύκλου <- $\pi * R^2$.
- Κάθε συνάρτηση εκτελείται, όπως ακριβώς εκτελούνται οι ενσωματωμένες συναρτήσεις της γλώσσας. Απλώς αναφέρεται το όνομά της σε μια έκφραση ή σε μία εντολή και επιστρέφεται η τιμή της, π.χ. Εμ <- Εμβαδό_κύκλου(R).

10.5.2 ΟΡΙΣΜΌΣ ΚΑΙ ΚΛΉΣΗ ΔΙΑΔΙΚΑΣΙΏΝ

- Κάθε διαδικασία έχει την ακόλουθη δομή:
ΔΙΑΔΙΚΑΣΙΑ Όνομα(<λίστα παραμέτρων>)

<Τμήμα δηλώσεων>

ΑΡΧΗ

<εντολές>

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

- Το όνομα της διαδικασίας είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ.
- Η λίστα παραμέτρων είναι μια λίστα μεταβλητών, των οποίων οι τιμές μεταβιβάζονται προς τη διαδικασία κατά την κλήση ή/και επιστρέφονται στο κύριο πρόγραμμα μετά το τέλος της διαδικασίας.
- Στο σώμα της διαδικασίας μπορούν να υπάρχουν οποιεσδήποτε εντολές της γλώσσας.
- Κάθε διαδικασία εκτελείται όταν καλείται από το κύριο πρόγραμμα ή άλλη διαδικασία.
- Η κλήση σε διαδικασία πραγματοποιείται με την εντολή ΚΑΛΕΣΕ, που ακολουθείται από το όνομα της διαδικασίας συνοδευόμενο μέσα σε παρενθέσεις με τη λίστα παραμέτρων.

10.5.2 ΟΡΙΣΜΌΣ ΚΑΙ ΚΛΉΣΗ

- Συνάρτηση

Η συνάρτηση μπορεί να χρησιμοποιηθεί όπως οποιαδήποτε μεταβλητή :

$Y \leftarrow \text{Όνομα συνάρτησης}(<\text{λίστα παραμέτρων}>)$

Αν Όνομα συνάρτησης(<λίστα παραμέτρων>) > 100 τότε

Γράψε Όνομα συνάρτησης(<λίστα παραμέτρων>)

- Διαδικασία

Η γενική μορφή της εντολής ΚΑΛΕΣΕ είναι :

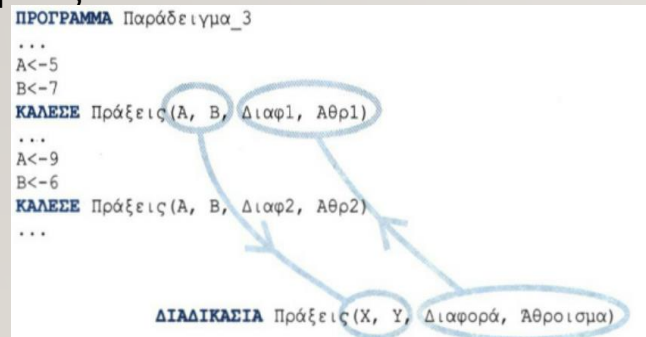
ΚΑΛΕΣΕ <Όνομα διαδικασίας> (<λίστα παραμέτρων>)

Λειτουργία

Η εκτέλεση του προγράμματος διακόπτεται και εκτελούνται οι εντολές της διαδικασίας που καλείται. Μετά το τέλος της διαδικασίας η εκτέλεση του προγράμματος συνεχίζεται από την εντολή που ακολουθεί. Η λίστα των παραμέτρων ορίζει τις τιμές που περνούν στη διαδικασία και τις τιμές που αυτή επιστρέφει. Η λίστα παραμέτρων δεν είναι υποχρεωτική.

10.5.3 ΠΡΑΓΜΑΤΙΚΈΣ ΚΑΙ ΤΥΠΙΚΈΣ ΠΑΡΆΜΕΤΡΟΙ

- Η λίστα των **τυπικών παραμέτρων (formal parameter list)** καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.
- Η λίστα των **πραγματικών παραμέτρων (actual parameter list)** καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.
- Μερικές γλώσσες προγραμματισμού ονομάζουν **ορίσματα** τις τυπικές παραμέτρους και απλά **παραμέτρους** τις πραγματικές παραμέτρους.



Οι λίστες των παραμέτρων πρέπει να ακολουθούν τους εξής **κανόνες**:

- Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.
- Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση. Για παράδειγμα, η πρώτη της λίστας των τυπικών παραμέτρων στην πρώτη της λίστας των πραγματικών παραμέτρων κ.ο.κ.
- Η τυπική παράμετρος και η αντίστοιχή της πραγματική πρέπει να είναι του ίδιου τύπου.

10.5 ΔΙΑΔΙΚΑΣΊΕΣ ΚΑΙ ΣΥΝΑΡΤΉΣΕΙΣ

Μια ΣΥΝΑΡΤΗΣΗ:

- δεν μπορεί να έχει εντολές εισόδου-εξόδου
- επιστρέφει πάντα ακριβώς μία τιμή
- δεν μπορεί να τροποποιήσει τις τιμές των πραγματικών παραμέτρων
- δεν μπορεί να επιστρέψει πίνακα

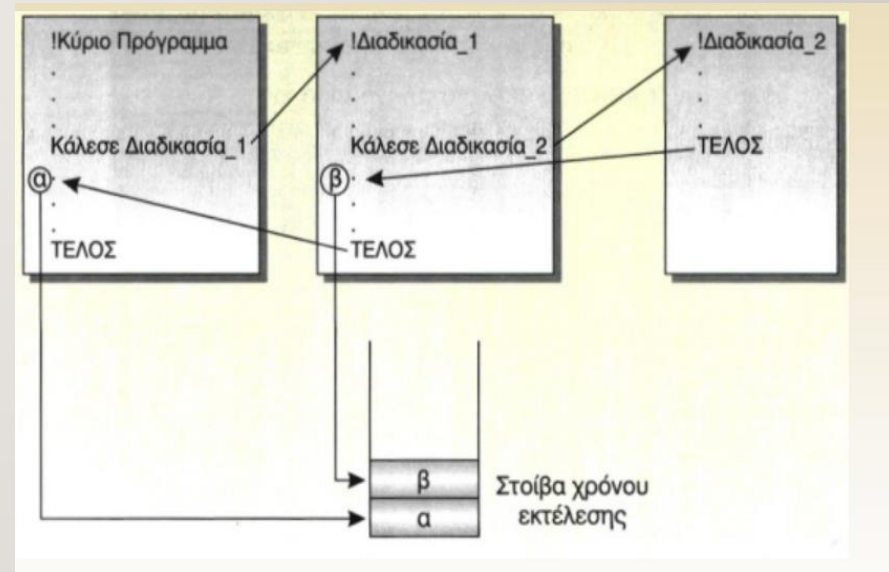
Αν πρέπει να διαφοροποιηθώ από τα παραπάνω,
πρέπει να χρησιμοποιήσω ΔΙΑΔΙΚΑΣΙΑ

ΠΡΟΣΟΧΗ

- Να επισημανθεί ιδιαίτερα ότι οι συναρτήσεις δεν μπορούν να έχουν εντολές εισόδου-εξόδου και ως εκ τούτου δε δύναται να πραγματοποιηθεί κλήση διαδικασίας από συνάρτηση.
- ... μια συνάρτηση που αλλάζει το περιεχόμενο των παραμέτρων της δεν επιστρέφει τις αλλαγές αυτές στις πραγματικές παραμέτρους.

10.5.3 Η ΧΡΗΣΗ ΣΤΟΙΒΑΣ ΣΤΗΝ ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΩΝ

- Η έννοια της στοίβας είναι πολύ χρήσιμη στο ίδιο το λογισμικό των γλωσσών προγραμματισμού. Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται *διεύθυνση επιστροφής* (return address), αποθηκεύεται από το μεταφραστή σε μία στοίβα που ονομάζεται *στοίβα χρόνου εκτέλεσης* (execution time stack).
- Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Η τεχνική αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μία διαδικασία ή συνάρτηση καλεί μία διαδικασία ή συνάρτηση. Για παράδειγμα, έστω ότι μία διαδικασία a καλεί τη διαδικασία b, που με τη σειρά της καλεί τη διαδικασία c κ.ο.κ.
- Στην περίπτωση αυτή οι διευθύνσεις επιστροφής εμφανίζονται στη στοίβα με σειρά c, b, a. Μετά την εκτέλεση κάθε διαδικασίας, η διεύθυνση επιστροφής απωθείται από τη στοίβα και ο έλεγχος μεταβιβάζεται στη διεύθυνση αυτή. Το παράδειγμα αυτό δείχνει μία από τις πολλές χρησιμότητες της LIFO ιδιότητας της στοίβας.



10.6 ΕΜΒΕΛΕΙΑ ΜΕΤΑΒΛΗΤΩΝ – ΣΤΑΘΕΡΩΝ

- Κάθε κύριο πρόγραμμα όπως και κάθε υποπρόγραμμα περιλαμβάνει τις δικές του μεταβλητές και σταθερές.
- Οι μεταβλητές αυτές στη ΓΛΩΣΣΑ είναι γνωστές στο αντίστοιχο υποπρόγραμμα που δηλώνονται και μόνο σε αυτό. Όλες οι μεταβλητές (και οι σταθερές) είναι τοπικές στο συγκεκριμένο τμήμα προγράμματος.
- Ο μόνος τρόπος για να περάσει μία τιμή από ένα υποπρόγραμμα σε ένα άλλο ή από το κυρίως πρόγραμμα σε ένα υποπρόγραμμα είναι διαμέσου των παραμέτρων κατά το στάδιο της κλήσης του υποπρογράμματος και μετά το τέλος της εκτέλεσης του υποπρογράμματος.
- Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται **εμβέλεια** (scope) μεταβλητών.

10.6 ΑΠΕΡΙΟΡΙΣΤΗ ΕΜΒΕΛΕΙΑ

- Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν. Όλες οι μεταβλητές είναι **καθολικές**.
- Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

10.6 ΠΕΡΙΟΡΙΣΜ'ΕΝΗ ΕΜΒ'ΕΛΕΙΑ

- Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι **τοπικές**, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια.
- Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

10.6 ΜΕΡΙΚΩΣ ΠΕΡΙΟΡΙΣΜΈΝΗ ΕΜΒΈΛΕΙΑ

- Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι τοπικές και άλλες καθολικές. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται οι μεταβλητές ως τοπικές ή καθολικές.
- Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

ΠΑΡΑΔΕΙΓΜΑΤΑ (ΘΈΜΑΤΑ ΠΑΝΕΛΛΑΔΙΚΏΝ) ΈΞΟΔΟΣ ΠΡΟΓΡΆΜΜΑΤΟΣ / "ΤΙΜΈΣ ΠΟΥ ΘΑ ΕΜΦΑΝΙΣΤΟΎΝ;« Ι

Δίνεται το παρακάτω πρόγραμμα και ένα υποπρόγραμμα:

Πρόγραμμα Θέμα_B2	Διαδικασία Διαδ(a,b)
Μεταβλητές	Μεταβλητές
Ακέραιες: a,b	Ακέραιες: a,b
Αρχή	Αρχή
$a \leftarrow 1$	$b \leftarrow b+a$
$b \leftarrow 3$	$a \leftarrow a+8$
Όσο $a < 35$ επανάλαβε	Γράψε a
Κάλεσε Διαδ(a,b)	Τέλος_Διαδικασίας
Γράψε b	
Τέλος_επανάληψης	
Τέλος_Προγράμματος	

Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του προγράμματος με τη σειρά που θα εμφανιστούν.

Μονάδες 10

B2. Δίνεται το παρακάτω πρόγραμμα και ένα υποπρόγραμμα:

Πρόγραμμα ΘέμαB	Διαδικασία Διαδ(w,z)
Μεταβλητές	Μεταβλητές
Ακέραιες: z,w	Ακέραιες: z,w
Αρχή	Αρχή
$z \leftarrow 1$	$w \leftarrow w+z$
$w \leftarrow 3$	$z \leftarrow z+2$
Όσο $z \leq 35$ επανάλαβε	Γράψε z
Κάλεσε Διαδ(z,w)	Τέλος_Διαδικασίας
Γράψε z	
Τέλος_επανάληψης	
Τέλος_Προγράμματος	

Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του προγράμματος με τη σειρά που θα εμφανιστούν.

Μονάδες 10

ΠΑΡΑΔΕΪΓΜΑΤΑ (ΘΈΜΑΤΑ ΠΑΝΕΛΛΑΔΙΚΏΝ) ΈΞΟΔΟΣ ΠΡΟΓΡΆΜΜΑΤΟΣ / "ΤΙΜΈΣ ΠΟΥ ΘΑ ΕΜΦΑΝΙΣΤΟ'ΥΝ;« II

Δίνεται το παρακάτω πρόγραμμα:

```
01 Πρόγραμμα Κλήση_Υποπρογραμμάτων
02 Μεταβλητές
03 Ακέραιες: α, β, γ, π
04 Αρχή
05 Διάβασε α
06 β ← 3
07 γ ← Φ(α,β)
08 α ← α+γ
09 Αν α>20 τότε
10     γ ← Φ(β,α)
11 Αλλιώς
12     γ ← Φ(γ,α)
13 Τέλος_αν
14 π ← 0
15 Όσο γ mod 10=0 επανάλαβε
16     π ← π+1
17     γ ← γ div 10
18 Τέλος_επανάληψης
19 Γράψε γ,π
20 Τέλος_προγράμματος
```

```
21 Συνάρτηση Φ(μ, λ): Ακέραια
22 Μεταβλητές
23 Ακέραιες: κ, λ, μ
24 Αρχή
25 κ ← λ+μ
26 Φ ← κ^μ
27 Τέλος_συνάρτησης
```

Για την παρακολούθηση της εκτέλεσης του προγράμματος με τιμή εισόδου α=2, δίνεται το παρακάτω υπόδειγμα πίνακα τιμών, μερικώς συμπληρωμένο ως εξής:

- Στη στήλη με τίτλο «αριθμός γραμμής» καταγράφεται ο αριθμός γραμμής της εντολής που εκτελείται.
- Στη στήλη με τίτλο «έξοδος» καταγράφεται η τιμή εξόδου, εφόσον η εντολή που εκτελείται είναι εντολή εξόδου.
- Στη στήλη με τίτλο «συνθήκη» καταγράφεται η λογική τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ, εφόσον η εντολή που εκτελείται περιλαμβάνει συνθήκη.
- Οι υπόλοιπες στήλες του πίνακα αντιστοιχούν στις μεταβλητές του κυρίου προγράμματος.
- Σε όποια σημεία καλείται υποπρόγραμμα και κατά τη διάρκεια της εκτέλεσής του, ο πίνακας επεκτείνεται με μια στήλη για κάθε μεταβλητή του υποπρογράμματος.

Αριθμός γραμμής	Έξοδος	Συνθήκη	α	β	γ	π				
05			2							
06				3						
07							Φ	κ	λ	μ
25									3	2
26								5		
07					25		25			
.....					...	...	...	...	...	...

Να μεταφέρετε τον πίνακα στο τετράδιό σας και να προσθέσετε τις γραμμές που χρειάζονται, συνεχίζοντας την εκτέλεση του προγράμματος, ως εξής: Για κάθε εντολή που εκτελείται, να γράψετε τον αριθμό της γραμμής της εντολής σε νέα γραμμή του πίνακα και το αποτέλεσμα της εκτέλεσης της εντολής στην αντίστοιχη στήλη.

Μονάδες 20

ΠΑΡΑΔΕΪΓΜΑΤΑ (ΘΈΜΑΤΑ ΠΑΝΕΛΛΑΔΙΚΏΝ) ΜΕΤΑΤΡΟΠΉΣ

```
ΣΥΝΑΡΤΗΣΗ ΠΑΡ(X,Y): ΑΚΕΡΑΙΑ  
ΜΕΤΑΒΛΗΤΕΣ  
ΑΚΕΡΑΙΕΣ: X,Y,Z  
ΑΡΧΗ  
    Z ← 0  
    Όσο X>0 Επανάλαβε  
        Αν X MOD 2 =1 τότε  
            Z ← Z + Y  
  
    Τέλος_αν  
    X ← X DIV 2  
    Y ← Y·2  
    Τέλος_επανάληψης  
    ΠΑΡ ← Z  
Τέλος_συνάρτησης
```

Η ΣΥΝΑΡΤΗΣΗ καλείται από το παρακάτω τμήμα προγράμματος:

```
...  
Διάβασε A, B  
Γράψε ΠΑΡ(A, B)  
Γράψε A,B  
...
```

- α. Να ξαναγράψετε το τμήμα προγράμματος, ώστε να επιτελεί την ίδια λειτουργία, καλώντας ΔΙΑΔΙΚΑΣΙΑ Δ1 αντί της ΣΥΝΑΡΤΗΣΗΣ ΠΑΡ. (μονάδες 3)
β. Να κατασκευάσετε την ΔΙΑΔΙΚΑΣΙΑ Δ1, ώστε να επιτελεί την ίδια λειτουργία με τη ΣΥΝΑΡΤΗΣΗ ΠΑΡ. (μονάδες 9)

ΣΗΜΕΙΩΣΗ: Το τμήμα προγράμματος που θα κατασκευάσετε θα πρέπει, με τη χρήση της ΔΙΑΔΙΚΑΣΙΑΣ, για τις ίδιες τιμές εισόδου να εμφανίζει τις ίδιες τιμές εξόδου με το τμήμα προγράμματος και τη χρήση της συνάρτησης που δόθηκαν.

Μονάδες 12