

# Θέματα Πανελλαδικών

Θεωρία Κεφαλαίου 10

Υποπρογράμματα

**Απαντήσεις**

Δημήτρης Παπαδάκης  
dimitrisp@easylearn.gr

EasyLearn.gr

## Πίνακας Περιεχομένων

|                                     |    |
|-------------------------------------|----|
| Ερωτήσεις Σωστού/Λάθους.....        | 3  |
| Ερωτήσεις Ανάπτυξης.....            | 5  |
| Ερωτήσεις Πολλαπλής Επιλογής.....   | 8  |
| 2007 – Θέμα Β2 – Ημερήσια.....      | 8  |
| Ασκήσεις.....                       | 9  |
| 2007 – Θέμα 2 – Ημερήσια.....       | 9  |
| Απάντηση – 1.....                   | 9  |
| Απάντηση – 2.....                   | 10 |
| 2018 – Θέμα Β2 – Επαναληπτικές..... | 10 |
| Απάντηση.....                       | 11 |
| 2021 – Θέμα Β1.....                 | 11 |
| Απάντηση.....                       | 11 |
| 2022 – Θέμα Α4 – Επαναληπτικές..... | 12 |
| Απάντηση.....                       | 13 |

EasyLearn.gr

## Ερωτήσεις Σωστού/Λάθους

Να γράψετε στο τετράδιο σας τον αριθμό κάθε πρότασης και δίπλα το γράμμα «Σ», αν είναι σωστή, ή το γράμμα «Λ», αν είναι λανθασμένη.

1. Ένα υποπρόγραμμα μπορεί να καλείται από ένα άλλο υποπρόγραμμα ή από το κύριο πρόγραμμα.

**Σωστό.**

2. Η λίστα των πραγματικών παραμέτρων καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.

**Σωστό.**

3. Ο τρόπος κλήσης των διαδικασιών και των συναρτήσεων είναι ίδιος, ενώ ο τρόπος σύνταξης τους είναι διαφορετικός.

**Λάθος.** Διαφέρουν τόσο στον τρόπο κλήσης όσο και στον τρόπο σύνταξης.

4. Στη διαδικασία η λίστα παραμέτρων είναι υποχρεωτική.

**Λάθος.**

5. Κατά την κλήση ενός υποπρογράμματος η πραγματική παράμετρος και η αντίστοιχη τυπική της είναι δυνατόν να έχουν το ίδιο όνομα.

**Σωστό.** Είναι δυνατόν αλλά δεν είναι απαραίτητο.

6. Όταν ένα υποπρόγραμμα καλείται από διαφορετικά σημεία του προγράμματος, οι πραγματικές παράμετροι πρέπει να είναι οι ίδιες.

**Λάθος.**

7. Ένα υποπρόγραμμα μπορεί να καλείται μόνο από το κύριο πρόγραμμα.

**Λάθος.**

8. Μεταξύ των εντολών του σώματος μιας συνάρτησης πρέπει υποχρεωτικά να υπάρχει τουλάχιστον μία εντολή εκχώρησης τιμής στο όνομα της συνάρτησης.

**Σωστό,** ώστε η συνάρτηση να επιστρέψει το αποτέλεσμα της.

9. Πραγματικές ονομάζονται οι παράμετροι που χρησιμοποιούνται κατά την κλήση ενός υποπρογράμματος.

**Σωστό.** (και τυπικές κατά τη δήλωσή του).

10. Μία συνάρτηση είναι δυνατό να επιστρέφει μόνον ακέραιες ή πραγματικές τιμές.

**Λάθος.** Επιστρέφει και αλφαριθμητικές και λογικές.

11. Σε μία συνάρτηση δεν επιτρέπεται η χρήση της εντολής Διάβασε.

**Σωστό.**

12. Όταν ένα υποπρόγραμμα καλείται από το κύριο πρόγραμμα, η διεύθυνση επιστροφής αποθηκεύεται από το μεταφραστή σε μια ουρά.

**Λάθος.** Αποθηκεύεται σε μια στοίβα.

13. Μια συνάρτηση υπολογίζει και επιστρέφει παραπάνω από μία τιμές με το όνομά της.

**Λάθος.** Επιστρέφει **μία μόνο** τιμή με το όνομά της.

14. Σε μία εντολή εκχώρησης δεν μπορεί να υπάρχει αναφορά σε περισσότερες από μία συναρτήσεις.

**Λάθος.** Παράδειγμα: Χρέωση  $\leftarrow$  Χρέωση\_Μηνυμάτων(SMS) + Χρέωση\_Δεδομένων(MB)

15. Κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα.

**Σωστό,** αν και στην πράξη είναι δύσκολο να επιτευχθεί η απόλυτη ανεξαρτησία.

16. Κάθε συνάρτηση επιστρέφει μόνο μία τιμή.

**Σωστό.**

17. Η λειτουργία των διαδικασιών είναι πιο περιορισμένη από τη λειτουργία των συναρτήσεων.

**Λάθος.** Ισχύει το αντίθετο, η λειτουργία των συναρτήσεων είναι πιο περιορισμένη από τη λειτουργία των διαδικασιών.

18. Η τυπική παράμετρος και η αντίστοιχή της πραγματική μπορεί να είναι διαφορετικού τύπου.

**Λάθος.** Πρέπει να είναι του ίδιου τύπου.

19. Η λίστα των τυπικών παραμέτρων καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.

**Σωστό,** ενώ η λίστα των πραγματικών παραμέτρων καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.

20. Μεταξύ των υποπρογραμμάτων δεν πρέπει να υπάρχει ανεξαρτησία.

**Λάθος.**

## Ερωτήσεις Ανάπτυξης

1. Να αιτιολογήσετε γιατί ένα από τα πλεονεκτήματα του τμηματικού προγραμματισμού είναι το ότι 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.

2. Τι είναι συνάρτηση (σε προγραμματιστικό περιβάλλον);

Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).

3. Τι είναι διαδικασία (σε προγραμματιστικό περιβάλλον);

Η διαδικασία είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος.

4. Να αναφέρετε τέσσερα πλεονεκτήματα του τμηματικού προγραμματισμού.

- α. Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος.
- β. Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.
- γ. Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.
- δ. Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.

5. Να αναπτύξετε δύο από τα πλεονεκτήματα του τμηματικού προγραμματισμού.

- α. 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.

- β. 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.

6. Να αναπτύξετε τρία χαρακτηριστικά των υποπρογραμμάτων.

- α. 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.
- β. 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.

- γ. 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.
7. Να αναφέρετε τους κανόνες που πρέπει να ακολουθούν οι λίστες των παραμέτρων κατά την κλήση ενός υποπρογράμματος.
- α. Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.
- β. 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.
- γ. Η τυπική παράμετρος και η αντίστοιχη της πραγματική πρέπει να είναι του ίδιου τύπου.
8. Τι ονομάζεται τμηματικός προγραμματισμός;
- Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.
9. Τι λέγεται υποπρόγραμμα;
- Υποπρόγραμμα λέγεται ένα τμήμα προγράμματος που επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα.
10. Τι ονομάζεται παράμετρος ενός υποπρογράμματος;
- Παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.
11. Ποια η διαφορά μεταξύ μεταβλητών και παραμέτρων;
- Οι παράμετροι είναι μεταβλητές που επιτρέπουν το πέρασμα των τιμών τους από ένα τμήμα προγράμματος σε ένα άλλο.
12. Ποια η διαφορά μεταξύ τυπικών και πραγματικών παραμέτρων;
- Η λίστα των τυπικών παραμέτρων (ή αλλιώς ορισμάτων) καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.
- Η λίστα των πραγματικών παραμέτρων (ή απλά παραμέτρων) καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.
13. Αναφέρετε τις ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα.
- α. Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο.
- β. Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.
- γ. Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο.
14. Ποιες μεταβλητές ονομάζονται καθολικές;
- Καθολικές λέγονται οι μεταβλητές, οι οποίες είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν.
15. Η χρήση καθολικών μεταβλητών σε ένα πρόγραμμα καταστρατηγεί μία από τις βασικές αρχές του τμηματικού προγραμματισμού (ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα). Να αναφέρετε ποια είναι αυτή η ιδιότητα και να εξηγήσετε γιατί καταστρατηγείται.

Η απεριόριστη εμβέλεια και η χρήση καθολικών μεταβλητών 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.

16. Να δώσετε τον ορισμό της διαδικασίας και τον ορισμό της συνάρτησης.

Δες τις απαντήσεις των ερωτήσεων 2 και 3.

17. Τι ονομάζεται εμβέλεια σε προγραμματιστικό περιβάλλον;

Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται εμβέλεια μεταβλητών.

18. Τι εμβέλεια χρησιμοποιείται στη Γλώσσα; Να αναφέρετε και να περιγράψετε το είδος της εμβέλειας (σταθερών, μεταβλητών) που υπάρχει στη «Γλώσσα».

Στη «Γλώσσα» έχουμε περιορισμένη εμβέλεια. 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.

19. Πώς ορίζεται ο τμηματικός προγραμματισμός;

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

## Ερωτήσεις Πολλαπλής Επιλογής

### 2007 – Θέμα Β2 – Ημερήσια

Για ποιες από τις παρακάτω περιπτώσεις μπορεί να χρησιμοποιηθεί συνάρτηση:

1. εισαγωγή ενός δεδομένου

**Λάθος.** Η συνάρτηση δεν μπορεί να εκτελέσει τις εντολές Διάβασε και Γράψε.

2. υπολογισμός του μικρότερου από πέντε ακεραίου

**Σωστό.**

3. υπολογισμός των δύο μικρότερων από πέντε ακεραίου

**Λάθος.** Η συνάρτηση επιστρέφει ένα μόνο αποτέλεσμα.

4. έλεγχος αν δύο αριθμοί είναι ίσοι

**Σωστό.** Επιστρέφει μια λογική τιμή.

5. ταξινόμηση πέντε αριθμών

**Λάθος.** Η ταξινόμηση επιστρέφει έναν πίνακα, δηλαδή πολλά αποτελέσματα.

6. έλεγχος αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο.

**Σωστό.** Επιστρέφει μια λογική τιμή.

## Ασκήσεις

### 2007 – Θέμα 2 – Ημερήσια

Δίνεται παρακάτω ένα πρόγραμμα με ένα υποπρόγραμμα:

Πρόγραμμα Υπολογισμοί

Μεταβλητές

Πραγματικές:  $\alpha, \beta, \gamma$

Αρχή

Διάβασε  $\alpha, \beta$

$\gamma \leftarrow \alpha + \text{Πράξη}(\alpha, \beta)$

Γράψε  $\gamma$

Τέλος\_Προγράμματος

Συνάρτηση Πράξη ( $x, \psi$ ): Πραγματική

Μεταβλητές

Πραγματικές:  $x, \psi$

Αρχή

ΑΝ  $x \geq \psi$  Τότε

Πράξη  $\leftarrow x - \psi$

Αλλιώς

Πράξη  $\leftarrow x + \psi$

Τέλος\_Αν

Τέλος\_Συνάρτησης

1. Να ξαναγράψετε το πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χρησιμοποιώντας διαδικασία αντί συνάρτησης.
2. Να ξαναγράψετε το πρόγραμμα που δόθηκε αρχικά, ώστε να επιτελεί την ίδια λειτουργία χωρίς τη χρήση υποπρογράμματος.

### Απάντηση – 1

Πρόγραμμα Υπολογισμοί

Μεταβλητές

Πραγματικές:  $\alpha, \beta, \gamma$

Αρχή

Διάβασε  $\alpha, \beta$

**Κάλεσε Πράξη( $\alpha, \beta, \gamma$ )**

**$\gamma \leftarrow \alpha + \gamma$**

Γράψε  $\gamma$

Τέλος\_Προγράμματος

Διαδικασία Πράξη ( $x$ ,  $\psi$ , **αποτέλεσμα**)

Μεταβλητές

Πραγματικές:  $x$ ,  $\psi$ , **αποτέλεσμα**

Αρχή

Αν  $x \geq \psi$  Τότε

**αποτέλεσμα**  $\leftarrow x - \psi$

Αλλιώς

**αποτέλεσμα**  $\leftarrow x + \psi$

Τέλος\_Αν

Τέλος\_Διαδικασίας

## Απάντηση – 2

Πρόγραμμα Υπολογισμοί

Μεταβλητές

Πραγματικές:  $a$ ,  $b$ ,  $\gamma$

Αρχή

Διάβασε  $a$ ,  $b$

Αν  $a \geq b$  Τότε

$\gamma \leftarrow a - b$

Αλλιώς

$\gamma \leftarrow a + b$

Τέλος\_Αν

$\gamma \leftarrow a + \gamma$

Γράψε  $\gamma$

Τέλος\_Προγράμματος

## 2018 – Θέμα Β2 – Επαναληπτικές

Δίνεται το παρακάτω υποπρόγραμμα σε «Γλώσσα» με όνομα Π\_Μ το οποίο ελέγχοντας τα στοιχεία 200 ατόμων υπολογίζει το πλήθος των ανήλικων ατόμων που έχουν κάποιο συγκεκριμένο όνομα.

(1) .....

Σταθερές

(2) .....

Μεταβλητές

(3) Ακέραιες: .....

(4) Χαρακτήρες: .....

Αρχή

$\Pi \leftarrow 0$

Για  $I$  από 1 μέχρι  $N$

Αν  $H[I] < 18$  και  $O[I] = X$  Τότε

$\Pi \leftarrow \Pi + 1$

Τέλος\_Αν

Τέλος\_Επανάληψης  
 $\Pi\_M \leftarrow \Pi$   
 (5) .....

Να γράψετε στο τετράδιό σας τους αριθμούς 1 έως 5 των γραμμών και δίπλα από κάθε αριθμό ό,τι χρειάζεται να συμπληρωθεί ώστε να είναι σωστή και πλήρης η σύνταξη του υποπρογράμματος.

## Απάντηση

1. Συνάρτηση  $\Pi\_M(H\Lambda, O, X)$ : Ακέραιο
2.  $N = 200$
3.  $H\Lambda[N^1]$ ,  $I$ ,  $\Pi$
4.  $O[N]$ ,  $X$
5. Τέλος\_Συνάρτησης

Παρακάτω δίνεται το υποπρόγραμμα συμπληρωμένο και με έντονη γραφή οι απαντήσεις.

**Συνάρτηση  $\Pi\_M(H\Lambda, O, X)$ : Ακέραιο**

Σταθερές

**$N = 200$**

Μεταβλητές

Ακέραιοι:  **$H\Lambda[N]$ ,  $I$ ,  $\Pi$**

Χαρακτήρες:  **$O[N]$ ,  $X$**

Αρχή

$\Pi \leftarrow 0$

Για  $I$  από 1 μέχρι  $N$

Αν  $H\Lambda[I] < 18$  και  $O[I] = X$  Τότε

$\Pi \leftarrow \Pi + 1$

Τέλος\_Αν

Τέλος\_Επανάληψης

$\Pi\_M \leftarrow \Pi$

**Τέλος\_Συνάρτησης**

## 2021 – Θέμα Β1

Να γράψετε υποπρόγραμμα, το οποίο να διαβάζει 1000 ακέραιους αριθμούς με έλεγχο εγκυρότητας, ώστε να είναι θετικοί. Το υποπρόγραμμα να επιστρέφει το πλήθος των αριθμών που είναι πολλαπλάσια του 3 και το άθροισμα των τριψήφιων.

## Απάντηση

Διαδικασία Υποπρόγραμμα(Πλήθος,  $\Sigma$ )

Μεταβλητές

Ακέραιοι:  $I$ ,  $X$ , Πλήθος,  $\Sigma$

Αρχή

- 1 Όπως και στην επόμενη γραμμή, όπου δηλώνεται ο πίνακας  $O[N]$ , θα μπορούσαμε να δηλώσουμε  $H\Lambda[200]$  αντί  $H\Lambda[N]$  και  $O[200]$  αντί  $O[N]$ . Όμως, καλύτερα να χρησιμοποιηθεί η σταθερά  $N$ , αφού υπάρχει η αντίστοιχη δήλωση.

**! Μηδενίζουμε πλήθος και άθροισμα**

Πλήθος  $\leftarrow 0$

$\Sigma \leftarrow 0$

Για I από 1 μέχρι 1000

**! Διαβάζουμε τον κάθε αριθμό ελέγχοντας ότι δίνεται θετικός**

Αρχή\_Επανάληψης

Διάβασε X

Μέχρις\_Ότου  $X > 0$

**! Εξετάζουμε αν ο αριθμός είναι πολλαπλάσιο του 3**

Αν  $X \bmod 3 = 0$  Τότε

Πλήθος  $\leftarrow$  Πλήθος + 1

Τέλος\_Αν

**! Εξετάζουμε αν ο αριθμός είναι τριψήφιος**

Αν  $X \geq 100$  και  $X \leq 999$  Τότε

$\Sigma \leftarrow \Sigma + X$

Τέλος\_Αν

Τέλος\_Επανάληψης

Τέλος\_Διαδικασίας

## 2022 – Θέμα Α4 – Επαναληπτικές

Δίνεται η παρακάτω Συνάρτηση:

Συνάρτηση ΠΑΡ(X, Y): Ακέραια

Μεταβλητές

Ακέραιες: X, Y, Z

Αρχή

$Z \leftarrow 0$

Όσο  $X > 0$  Επανάλαβε

Αν  $X \bmod 2 = 1$  Τότε

$Z \leftarrow Z + Y$

Τέλος\_Αν

$X \leftarrow X \text{ DIV } 2$

$Y \leftarrow Y * 2$

Τέλος\_Επανάληψης

$\text{ΠΑΡ} \leftarrow Z$

Τέλος\_Συνάρτησης

Η Συνάρτηση 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 **Διαδικασία Δ1** αντί της **Συνάρτησης ΠΑΡ**.
- β. Να κατασκευάσετε την **Διαδικασία Δ1**, ώστε να επιτελεί την ίδια λειτουργία με τη **Συνάρτηση ΠΑΡ**.

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

## Απάντηση

Διαδικασία ΠΑΡ(X, Y, Z)

Μεταβλητές

Ακέραιες: X, Y, Z, Αρχική\_Τιμή\_X, Αρχική\_Τιμή\_Y

Αρχή

**Αρχική\_Τιμή\_X** ← X

**Αρχική\_Τιμή\_Y** ← Y

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

**X** ← Αρχική\_Τιμή\_X

**Y** ← Αρχική\_Τιμή\_Y

Τέλος\_Διαδικασίας

Και η κλήση του υποπρογράμματος

...

Διάβασε A, B

**Κάλεσε ΠΑΡ(A, B, Γ)**

**Γράψε Γ**

Γράψε A, B

...