

ΑΣΚΗΣΕΙΣ

Αναδρομή (*Recursion*)

ΑΣΚΗΣΗ-1^η

Να γραφεί το πρόγραμμα Java που **υπολογίζει αναδρομικά το παραγοντικό** ενός αριθμού με τις παρακάτω δύο παραλλαγές:

- (1) - με την χρήση αντικειμένου
- (2) - με την χρήση στατικής μεθόδου

Τα αποτελέσματα θα εμφανίζονται στην `main()`.

ΑΣΚΗΣΗ-2^η

Να γραφεί το πρόγραμμα Java που **υπολογίζει το παραγοντικό** ενός αριθμού με την χρήση απλής εντολής επανάληψης (loop) (π.χ. `for...`).

ΑΣΚΗΣΗ-3^η

Να γραφεί το πρόγραμμα Java που **υπολογίζει αναδρομικά και εμφανίζει τον αριθμό της ακολουθίας Fibonacci**, δοθείσης της θέσης του στην ακολουθία. Π.χ. αν ζητήσουμε τον αριθμό στην θέση 7 θα μας εμφανίσει τον αριθμό 13 της ακολουθίας. . Δύο παραλλαγές:

- (1) με χρήση αντικειμένου,
- (2) με χρήση στατικής μεθόδου

ΑΣΚΗΣΗ-4^η

Να γραφεί το πρόγραμμα Java που υπολογίζει με την **τεχνική της αντιστροφής του αριθμού** αν ένας αριθμός είναι παλινδρομικός ή μη.

ΑΣΚΗΣΗ-5^η

Να γραφεί το πρόγραμμα Java που υπολογίζει με την **τεχνική της σύγκρισης των χαρακτήρων** αν ένα String είναι παλινδρομικό ή μη.

ΑΣΚΗΣΗ-6^η

Να γραφεί το πρόγραμμα Java που **ελέγχει με αναδρομή** ποιοι από τους 25 πρώτους φυσικούς αριθμούς (1-25) είναι πρώτοι αριθμοί.

ΑΣΚΗΣΗ-7^η

Να γραφεί το πρόγραμμα Java που **ελέγχει με αναδρομή** αν ένας αριθμός είναι μονός ή ζυγός. Για παράδειγμα αν εισάγουμε τον αριθμό 3 σαν παράμετρο στη μέθοδο **isEven(3)**, τότε αναδρομικά θα κληθούν με την σειρά οι μέθοδοι isOdd(3-1), δηλ. **isOdd(2)**, μετά isEven(2-1), δηλ. **isEven(1)** και τέλος isOdd(1-1), δηλ. **isOdd(0)**. Ο αλγόριθμος θα ισχύει για οποιοδήποτε αριθμό εισάγουμε σαν παράμετρο.

ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ:

ΑΣΚΗΣΗ-1^η

Να γραφεί το πρόγραμμα Java που υπολογίζει αναδρομικά την δύναμη ενός οποιοδήποτε αριθμού. Η άσκηση προσομοιώνει την μέθοδο **power(int a, int n)** που θα δέχεται σαν παραμέτρους δύο αριθμούς τον a=βάση και τον n=δύναμη. Είναι γνωστά ότι:

$$a^0 = 1$$

$$a^n = a^{n-1} \cdot a \quad (\text{για } n \geq 1)$$

Μια προτεινόμενη αναδρομική σχέση είναι: **p = a*power(a, n-1);**

Παράδειγμα:

```
power(2,3) => 2* power(2,2)
              = 2*2* power(2,1)
              = 2*2*2* power(2,0)
              = 2*2*2*1= 8
```

ΑΣΚΗΣΗ-2^η

Να γραφεί το πρόγραμμα Java που εμφανίζει τα ψηφία ενός ακέραιου αριθμού αντίστροφα. Χρησιμοποιήστε μία **στατική αναδρομική μέθοδο** π.χ.

```
public static void reverseDisplay(int value)
```

η οποία θα δέχεται σαν παράμετρο τον αριθμό του οποίου τα ψηφία θα αντιστραφούν. Για παράδειγμα αν καλέσουμε την μέθοδο : *reverseDisplay(87654)*, τότε θα εμφανίσει 45678.

ΑΣΚΗΣΗ-3^η (Πύργοι του Hanoi - towers of Hanoi)

Να γραφεί το πρόγραμμα Java που μετακινεί οποιοδήποτε πλήθος δίσκων (το πλήθος θα εισάγεται από το πληκτρολόγιο) από μία θέση (π.χ. A) σε μια άλλη (π.χ. C) με την βοήθεια μιας τρίτης (π.χ. B). Αν λάβετε υπόψιν τον αλγόριθμο που αναλύθηκε στην θεωρία, θα χρειαστείτε τις παρακάτω μεταβλητές:

- το πλήθος των δίσκων n
- τρεις μεταβλητές τύπου char τις 'A', 'B', και 'C'

Μία αναδρομική μέθοδος (σύμφωνα με τον προηγούμενο αλγόριθμο) θα ήταν της μορφής:

```
void moveDisks(int n, char A, char C, char B)
```