



ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 7

ΘΕΜΑ: ΑΝΑΔΡΟΜΗ



ΠΕΡΙΕΧΟΜΕΝΟ

- Ορισμός
- Τύποι αναδρομής
- Παραδείγματα



ΑΝΑΔΡΟΜΗ

Ένας αλγόριθμος μπορεί να καλεί τον εαυτό του άμεσα ή έμμεσα.



ΠΑΡΑΔΕΙΓΜΑ

Αναδρομή

Είσοδος: Πλήθος επιπέδων n με $n=1$ αρχικά

Έξοδος: Αριθμοί επισκεπτόμενων επιπέδων

1. Επέστρεψε Level n
2. if ($n < 4$)
then
 - 2.1 Αναδρομή με είσοδο $n+1$
- Τέλοςif
3. Επέστρεψε Level n



ΕΞΟΔΟΣ

Level 1

Level 2

Level 3

Level 4

Level 4

Level 3

Level 2

Level 1



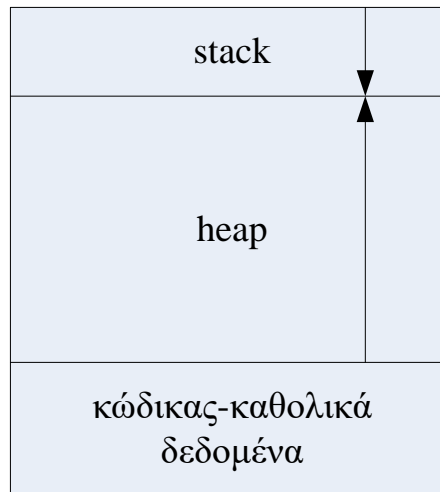
ΣΤΟΙΒΑ

- Όταν ένα πρόγραμμα φορτώνεται στη μνήμη για να εκτελεστεί, η περιοχή μνήμης όπου τοποθετούνται τα δεδομένα του προγράμματος ονομάζεται **περιοχή δεδομένων** και είναι καθορισμένη.
- Στο ένα άκρο της περιοχής δεδομένων αποθηκεύονται οι **καθολικές μεταβλητές** και στο άλλο άκρο οι **τοπικές μεταβλητές**.
- Οι **τοπικές μεταβλητές** τοποθετούνται σε μια **δομή στοίβας** (stack).

ΠΕΡΙΟΧΗ ΔΕΔΟΜΕΝΩΝ

Ο χώρος μνήμης μεταξύ στοίβας και καθολικών μεταβλητών είναι ελεύθερος και λέγεται **ελεύθερη περιοχή μνήμης (heap)**.

Η μνήμη στο heap κατανέμεται δυναμικά από τον προγραμματιστή.





ΛΕΙΤΟΥΡΓΙΑ ΣΤΟΙΒΑΣ

Στη στοίβα δυο λειτουργίες μπορούν να λάβουν χώρα:

- Εισαγωγή δεδομένων (push)
- Εξαγωγή δεδομένων (pop)



ΕΙΣΑΓΩΓΗ/ΕΞΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

- Τα δεδομένα εισάγονται στην πρώτη κενή θέση της στοίβας.
- Τα δεδομένα εξάγονται από την κορυφή της στοίβας.
- Το τελευταίο δεδομένο που εισάγαμε στη στοίβα είναι και το πρώτο υποψήφιο προς εξαγωγή.



ΠΑΡΑΔΕΙΓΜΑ

```
PROCEDURE updown(n:integer);  
  BEGIN  
    writeln('Level',n);  
    if n<4 then updown(n+1);  
    write('Level',n);  
  END;
```

Έστω ότι έχουμε την κλήση updown(1).

ΚΑΤΑΣΤΑΣΗ ΣΤΟΙΒΑΣ 1/2



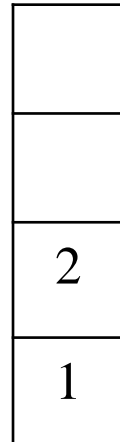
n

(α) Πριν την κλήση



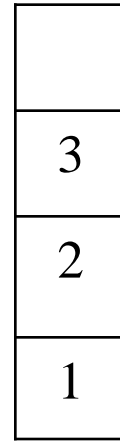
n

(β) updown(1)



n

(γ) updown(2)



n

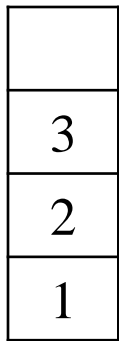
(δ) updown(3)



n

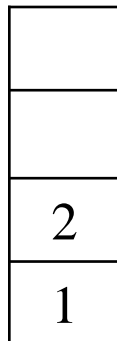
(ε) updown(4)

ΚΑΤΑΣΤΑΣΗ ΣΤΟΙΒΑΣ 2/2



n

(σ) write('Level',n);
//Level 4



n

(ζ) write('Level',n);
//Level 3



n

(η) write('Level',n);
//Level 2



n

(θ) write('Level',n);
//Level 1



ΤΥΠΟΙ ΑΝΑΔΡΟΜΗΣ

- **Αναδρομή τέλους:** κλήση αναδρομής στο τέλος του αλγορίθμου.
- **Αναδρομή και αντιστροφή:** κλήση αναδρομής πριν από εμφάνιση αποτελέσματος στην οθόνη. Το αποτέλεσμα εμφανίζεται αντίστροφα από τη σειρά των αναδρομικών κλήσεων του αλγορίθμου.



ΑΝΑΔΡΟΜΗ ΤΕΛΟΥΣ

Πρόβλημα: Να υπολογιστεί ο μέγιστος κοινός διαιρέτης δυο αριθμών.



ΑΛΓΟΡΙΘΜΟΣ ΣΕ ΕΛΕΥΘΕΡΟ ΚΕΙΜΕΝΟ

Ο μέγιστος κοινός διαιρέτης δυο αριθμών υπολογίζεται με συνεχείς ακέραιες διαιρέσεις.

Στην πρώτη διαίρεση διαιρούμε τους δυο αριθμούς και υπολογίζουμε το υπόλοιπο.

Στη συνέχεια διαιρούμε τον δεύτερο αριθμό με το υπόλοιπο της προηγούμενης διαίρεσης.

Αυτό συνεχίζεται μέχρι να πάρουμε υπόλοιπο 0. Τότε ο δεύτερος αριθμός της διαίρεσης που έδωσε υπόλοιπο 0 είναι αυτό που ψάχνουμε.



ΑΠΑΝΤΗΣΗ

Μέγιστος Κοινός Διαιρέτης δυο ακεραίων αριθμών

Είσοδος: Δυο αριθμοί a, b

Έξοδος: Μέγιστος κοινός διαιρέτης: mkd

1. Επέστρεψε a, b
2. if ($b=0$)
 then
 - 2.1 $mkd=a$
 - 2.2 $mkd=$ Μέγιστος Κοινός Διαιρέτης δυο αριθμών με είσοδο b , υπόλοιπο ακέραιας διαίρεσης a με b
- Τέλοςif
3. Επέστρεψε Μέγιστος Κοινός Διαιρέτης: mkd



ΤΥΧΑΙΑ ΕΞΟΔΟΣ

a: 20

b: 12

20 12

12 8

8 4

4 0

Μέγιστος Κοινός Διαιρέτης : 4



ΑΝΑΔΡΟΜΗ ΚΑΙ ΑΝΤΙΣΤΡΟΦΗ

Πρόβλημα: Να γραφεί αλγόριθμος που θα ζητά να του δίνονται ονόματα ζώων μέχρι να ανιχνεύσει ένα κενό αλφαριθμητικό. Στη συνέχεια να τυπώνει τα ονόματα που του δόθηκαν με αντίστροφη σειρά.



ΑΠΑΝΤΗΣΗ

Ονόματα Ζώων

Είσοδος: Όνομα ζώου animal

Έξοδος: Όνομα ζώου animal

1. Επέστρεψε Select an animal:
2. Διάβασε όνομα ζώου animal
3. if (πρώτος χαρακτήρας animal<>κενό)
then
 - 3.1 Ονόματα Ζώων
Τέλοςif
4. Επέστρεψε animal
Τέλος



ΤΥΧΑΙΑ ΕΞΟΔΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Select an animal: Bear

Select an animal: Cat

Select an animal: Dog

Select an animal:

Dog

Cat

Bear