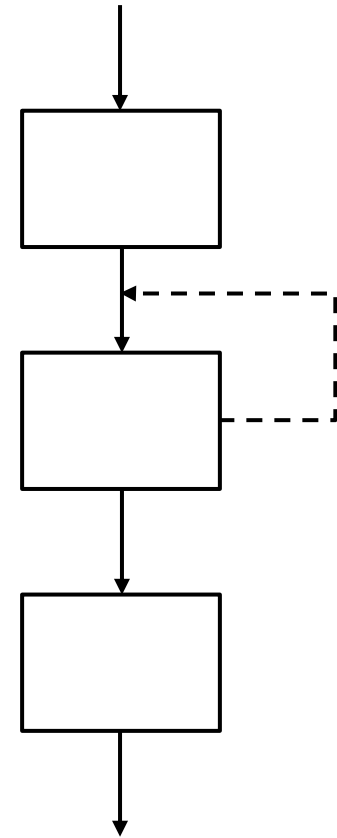
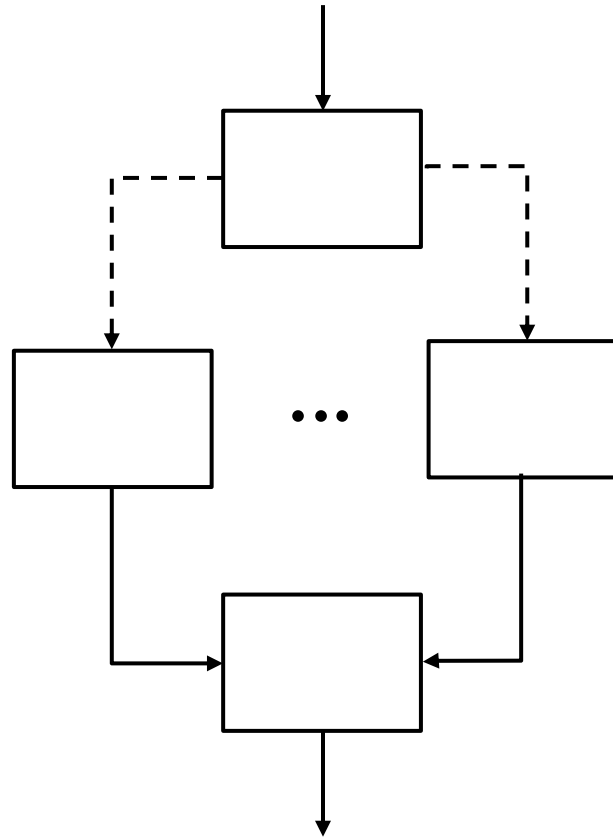
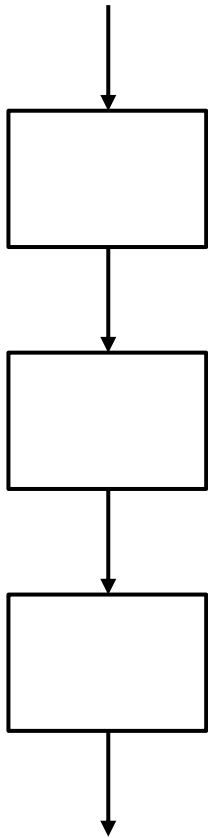


ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εντολές Επανάληψης: Η εντολή `for` - Μέρος Α

Τρόποι εκτέλεσης

- Σειριακή εκτέλεση
- Εντολές διακλάδωσης (π.χ., `if`, `if..else`, `if..elseif..else`)
- Εντολές επανάληψης (π.χ., `for`, `while`)



Εντολές Επανάληψεων

- Οι εντολές επανάληψης χρησιμοποιούνται ώστε να εκτελούμε εντολές πολλές φορές.
 - Ένα σύνολο εντολών (βρόγχος – loop) εκτελείται επαναληπτικά για ένα συγκεκριμένο εύρος τιμών ή για όσο ικανοποιείται μία συνθήκη.
- Η MATLAB/Octave διαθέτει τις παρακάτω εντολές επανάληψης:
 - **for** – επανάληψη για μία συγκεκριμένη ακολουθία τιμών,
 - **while** – επανάληψη για όσο ικανοποιείται μία συνθήκη.

Η εντολή for – Σύνταξη

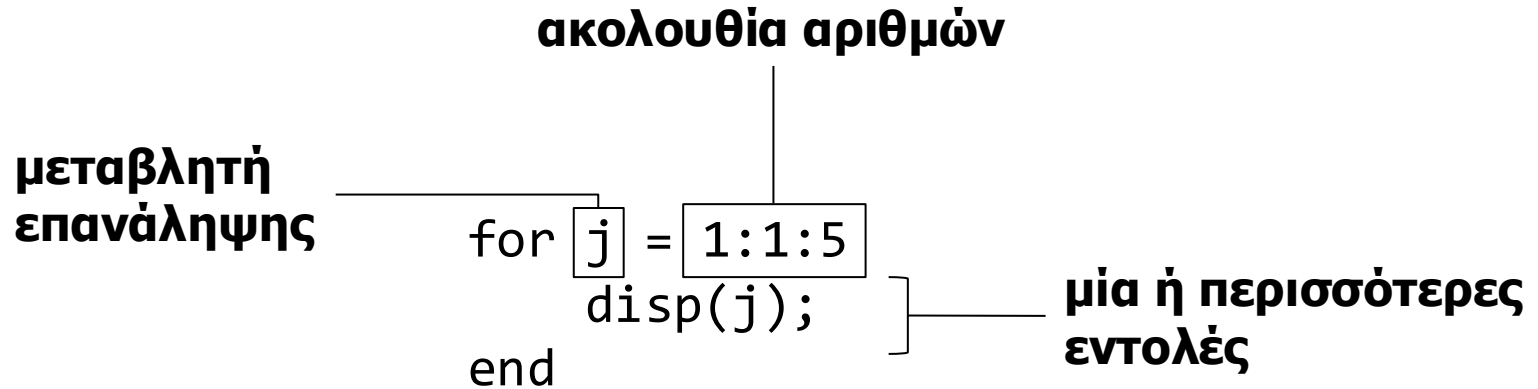
```
for var = AT:B:TT
    εντολές;
end
```

- *AT*: αρχική τιμή, *B*: βήμα, *TT*: τελική τιμή
- Δεν γίνεται *καμία* επανάληψη όταν:
 - $AT > TT$ και $B > 0$ ή
 - $AT < TT$ και $B < 0$
- Για μοναδιαίο *B* ($B=1$), η *for* μπορεί να γραφεί και ως:

```
for var = AT:TT
    εντολές;
end
```

Η εντολή for – Παράδειγμα

- Η εντολή for χρησιμοποιείται για να εκτελέσουμε ένα σύνολο εντολών πολλές φορές.



Η εντολή for – Ανάλυση

- Η **μεταβλητή επανάληψης** παίρνει την πρώτη τιμή της **ακολουθίας αριθμών**.
- Εκτελούνται όλες οι **εντολές** μία προς μία.

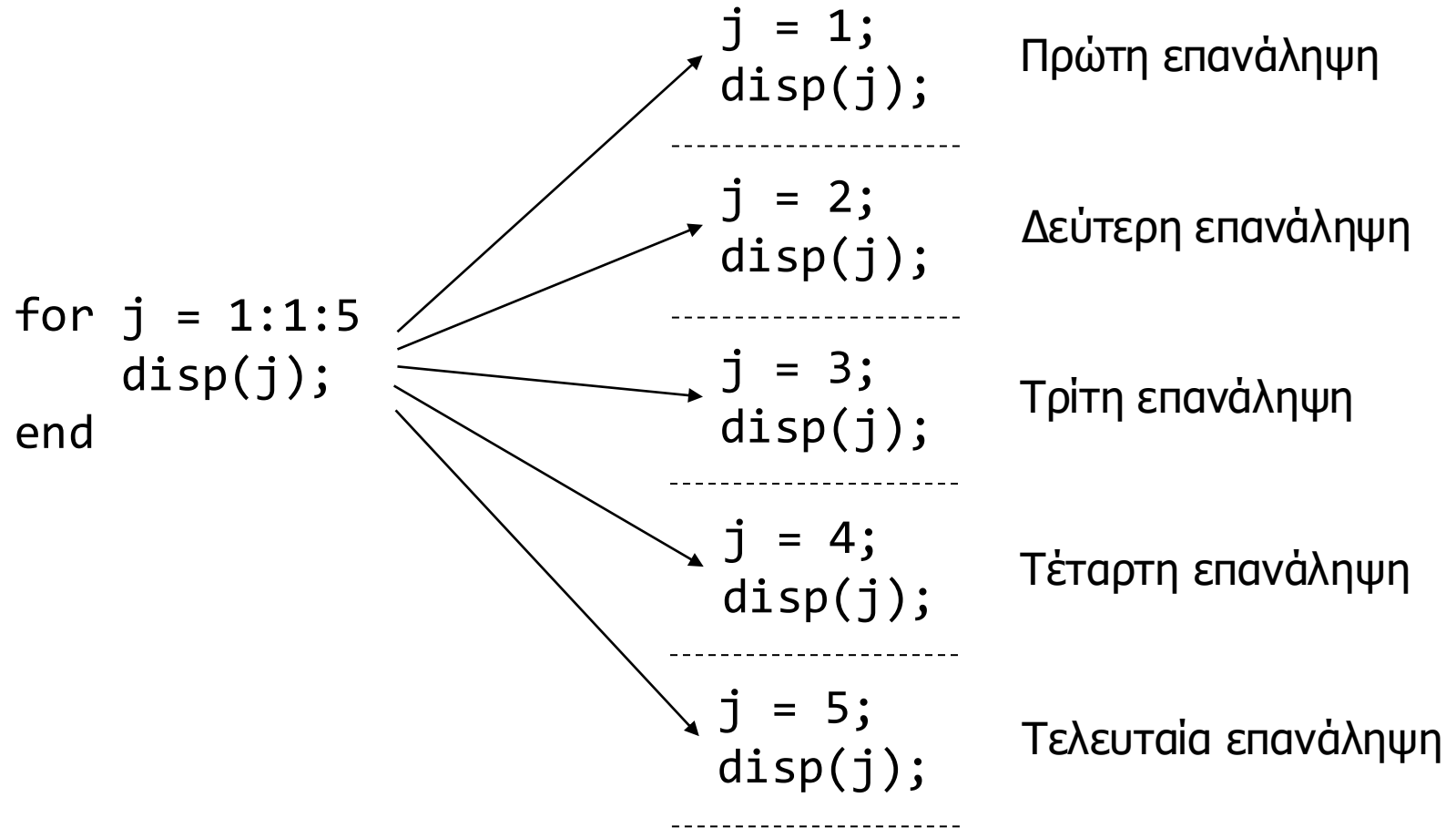
- Η **μεταβλητή επανάληψης** παίρνει την επόμενη τιμή της **ακολουθίας αριθμών**.
- Εκτελούνται όλες οι **εντολές** μία προς μία.

- ...ΚΟΚ...

- Η **μεταβλητή επανάληψης** παίρνει την τελευταία τιμή της **ακολουθίας αριθμών**.
- Εκτελούνται όλες οι **εντολές** μία προς μία.

Η εντολή for – Βήμα-Βήμα

- Αν «ξεδιπλώναμε» την εκτέλεση της προηγούμενης for θα είχαμε κάτι τέτοιο:



Ειδικές Περιπτώσεις

- Αντί για αύξηση της μεταβλητής ανά 1, μπορούμε να έχουμε μία οποιαδήποτε αύξηση ή μείωση.

- **Παράδειγμα 1:** Από το 1 έως το 20 με βήμα 2

```
for i = 1:2:20
    disp(i);
end
```

- **Παράδειγμα 2:** Από το 1000 έως το 0 με βήμα -100

```
for i = 1000:-100:0
    disp(i);
end
```

Η εντολή for – Παραδείγματα (1/2)

```
for i = 1:1:10  
    disp(i);  
end
```

Η επανάληψη θα γίνει 10 φορές. Το i θα αρχίσει από το 1 και σε κάθε επανάληψη θα αυξάνεται κατά 1, μέχρι να γίνει 10.

```
for j = 1:10  
    disp(j);  
end
```

Ισοδύναμο με την πρώτη περίπτωση: Η επανάληψη θα γίνει 10 φορές. Εφόσον δεν έχουμε ορίσει βήμα, αυτό θα είναι 1 (προεπιλογή). Άρα, το j θα αρχίσει από το 1 και θα αυξάνεται κατά 1, μέχρι να γίνει 10.

```
for m = 2:3:10  
    disp(m);  
end
```

Η επανάληψη θα γίνει 3 φορές. Το m θα αρχίσει από το 2 και σε κάθε επανάληψη θα αυξάνεται κατά 3, μέχρι να γίνει μεγαλύτερο από 10. Άρα το m θα πάρει διαδοχικά τις τιμές 2, 5 και 8.

```
for k = 1:2:100  
    disp(k);  
end
```

Η επανάληψη θα γίνει 50 φορές. Το k θα αρχίσει από το 1 και σε κάθε επανάληψη θα αυξάνεται κατά 2, μέχρι να γίνει μεγαλύτερο από 100. Άρα το k θα πάρει διαδοχικά τις τιμές 1, 3, 5, .. 95, 97, 99.

Η εντολή for – Παραδείγματα (2/2)

```
for i = 0:2:100  
    disp(i);  
end
```

Η επανάληψη θα αρχίσει από το 0 και σε κάθε επανάληψη θα αυξάνεται κατά 2, μέχρι να γίνει 100. Οπότε θα τυπώσει όλους τους άρτιους αριθμούς από το 0 έως το 100.

```
for j = 100:10:1  
    disp(j);  
end
```

Δεν θα εκτελεστεί καμία επανάληψη, γιατί το 100 είναι μεγαλύτερο από το 1, και το βήμα είναι θετικός αριθμός.

```
for k = 100:-50:10000  
    disp(k);  
end
```

Δεν θα εκτελεστεί καμία επανάληψη, γιατί το 100 είναι μικρότερο από το 10000 και το βήμα είναι αρνητικό.

```
for n = 10:-1:1  
    disp(n);  
end
```

Η επανάληψη θα γίνει 10 φορές. Το n θα αρχίσει από το 10 και σε κάθε επανάληψη θα μειώνεται κατά 1, μέχρι να γίνει 1.

Άσκηση

- Να γραφεί πρόγραμμα που να εμφανίζει όλους τους ακέραιους αριθμούς από το 0 έως το 40 οι οποίοι είναι πολλαπλάσια του 4.
- Λύση:

```
for i = 0:4:40
    disp(i);
end
```

Παραδείγματα χρήσης (1/2)

- Ζητάμε από το χρήστη να εισάγει 5 αριθμούς:

```
for j = 1:1:5
    data = input('Type a number: ');
end
```

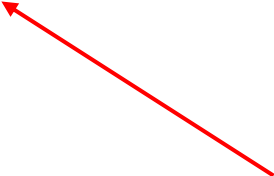
- Κάνουμε κάτι απλό με τους αριθμούς, π.χ. τους τυπώνουμε:

```
for j = 1:1:5
    data = input('Type a number: ');
    disp(data);
end
```

Παραδείγματα χρήσης (2/2)

- Κάνουμε κάτι σύνθετο με τους αριθμούς, π.χ, τους προσθέτουμε:

```
for j = 1:1:5
    data = input('Type a number: ');
    ...
end
```



- Έχουμε μόνο έναν αριθμό τη φορά: πως θα τους προσθέσουμε?
- **Λύση:** χρησιμοποιούμε μια μεταβλητή για το τρέχων άθροισμα:

```
sum = 0;
for j = 1:1:5
    data = input('Type a number: ');
    sum = sum + data;
end
```

Εύρεση πλήθους, αθροίσματος και γινομένου

- Υπολογισμός πλήθους στοιχείων (μέτρημα):
 - Ορίζω μία μεταβλητή, πχ. count = 0 ή plithos = 0
 - Όταν πρέπει να μετρήσω ένα στοιχείο, αυξάνω την μεταβλητή, πχ. count = count + 1
- Υπολογισμός αθροίσματος:
 - Ορίζω μια μεταβλητή, πχ. sum = 0
 - Προσθέτω έναν αριθμό num στην μεταβλητή sum, πχ. sum = sum + num
- Υπολογισμός γινομένου:
 - Ορίζω μια μεταβλητή, πχ. prod = 1
 - Πολλαπλασιάζω έναν αριθμό num στην μεταβλητή prod, πχ. prod = prod * num

Παράδειγμα: Πρόσθεση

- Σύνολο μηνιαίων μισθών ενός έτους:

```
total = 0;
```

```
for i = 1:1:12
```

```
    monthly = input('Μισθός: ');
```

```
    total = total + monthly;
```

```
end
```

```
fprintf('Το σύνολο των μισθών είναι %d\n', total);
```

Επεξήγηση: Συνολικά χρειάζονται μόνο δύο μεταβλητές: μία για τον εκάστοτε προσθετέο (`monthly`) και μία για το άθροισμα (`total`). Το άθροισμα αρχικά παίρνει την τιμή 0. Στη συνέχεια, για κάθε επανάληψη, γίνεται ανάγνωση μιας τιμής και αποθηκεύεται στην `monthly`. Μετά, η `monthly` προσθέτετε στο τρέχον άθροισμα `total`.

Παράδειγμα: Μέσος Όρος

- Μέσος όρος μηνιαίων μισθών ενός έτους:

```
total = 0;
```

```
count = 0;
```

```
for i = 1:1:12
```

```
    monthly = input('Μισθός: ');
```

```
    total = total + monthly;
```

```
    count = count + 1;
```

```
end
```

```
average = total / count;
```

```
fprintf('Ο μέσος όρος είναι %d\n', average);
```

Επεξήγηση: Αφού υπολογίσουμε το άθροισμα όλων των μισθών του έτους, στη συνέχεια διαιρούμε με το πλήθος των μισθών.

Ένθετες εντολές if

- Ένα for μπορεί να περιέχει μια οποιαδήποτε εντολή επομένως μπορεί να περιέχει και ένα if.
- **Παράδειγμα:** εύρεση αρνητικών αριθμών.

```
for j = 1:1:5
    data = input('Δώσε αριθμό: ');

    if data < 0
        fprintf('0 αριθμός είναι αρνητικός. ');
    end
end
```

Άσκηση 1

- Να γράψετε ένα πρόγραμμα που να ζητάει από το χρήστη να εισάγει 20 αριθμούς (έναν τη φορά) και να τυπώνει το άθροισμα μόνο των θετικών αριθμών.

- **Λύση:**

```
sum = 0;
for i = 1:1:20
    n = input('Δώσε αριθμό: ');

    if n > 0
        sum = sum + n;
    end
end

fprintf('Το άθροισμα των θετικών είναι %d\n', sum);
```

Άσκηση 2

- Να γράψετε ένα πρόγραμμα που να ζητάει από το χρήστη να εισάγει 20 αριθμούς (έναν τη φορά) και να τυπώνει το πλήθος των θετικών και το πλήθος των αρνητικών αριθμών.

- **Λύση:**

```
countPos = countNeg = 0;
```

```
for i = 1:1:20
```

```
    n = input('Δώσε αριθμό: ');
```

```
    if n > 0
```

```
        countPos = countPos + 1;
```

```
    elseif n < 0
```

```
        countNeg = countNeg + 1;
```

```
    end
```

```
end
```

```
fprintf('Το πλήθος των θετικών είναι %d\n', countPos);
```

```
fprintf('Το πλήθος των αρνητικών είναι %d\n', countNeg);
```

Η εντολή break

- Η εντολή break μέσα σε επανάληψη προκαλεί τερματισμό της εκτέλεσης της επανάληψης.
- Η εκτέλεση του προγράμματος μεταφέρεται στην εντολή που υπάρχει αμέσως το end της επανάληψης.
- Παράδειγμα:

```
for c = 1:1:20
    if c == 12
        break;
    end
    disp(c);
end
```

Η εντολή `continue`

- Η εντολή `continue` προκαλεί διακοπή των εντολών της επανάληψης που εκτελείται τώρα.
- Όμως η επανάληψη συνεχίζεται με την επόμενη επανάληψη που υπαγορεύεται από την επαναληπτική διαδικασία.
- Παράδειγμα:

```
for c = 1:1:20
    if c == 12
        continue;
    end
    disp(c);
end
```

Ένθετη for

- Ένα for μπορεί να περιέχει μία οποιαδήποτε εντολή επομένως μπορεί να περιέχει και ένα άλλο for.
- Στην περίπτωση αυτή έχουμε ένθετα for.

Άσκηση: Προπαίδεια του 10

- Να γράψετε ένα πρόγραμμα που να τυπώνει την προπαίδεια

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Συχνά Λάθη

- Λάθος όρια ή λάθος βηματισμός:

```
for i = 100:1
    disp(i);
end
```

```
for i = 1:-2:20
    disp(i);
end
```

- Λάθος χρήση της μεταβλητής επανάληψης:

```
i = 0;
for i = 1:1:100
    i = i - 1;
    disp(i);
end
```

Παρατηρήσεις

- Η στοίχιση του κώδικα βοηθάει πολύ στην κατανόηση και στην αποφυγή λαθών.

Κακογραμμένος κώδικας (δυσανάγνωστος):

```
sum = 0; for i=1:1:10,sum = sum + i; end  
disp(sum);
```

Καλά στοιχισμένος κώδικας (ευανάγνωστος):

```
sum = 0;  
  
for i = 1:1:10  
    sum = sum + i;  
end  
  
disp(sum);
```