6. Εντολές επανάληψης

Τμήμα Μηχανικών Σχεδίασης Προϊόντων και Συστημάτων

Εντολή επανάληψης for

Ένα από τα μεγάλα πλεονεκτήματα των υπολογιστών είναι ότι μπορούν να επαναλαμβάνουν μια διαδικασία πολλές φορές. Οι εντολές αυτές ονομάζονται **εντολές βρόχων** ή **βρόχοι(loops).**

Οι βασικές εντολές επανάληψης είναι η **for** και η **while**. Η διαφορά τους είναι ότι η **for** προκαλεί εκτέλεση μιας ομάδας εντολών για συγκεκριμένο αριθμό επαναλήψεων, ενώ με τη **while** ο αριθμός των επαναλήψεων δεν είναι συγκεκριμένος, αλλά η ομάδα εντολών εκτελείται επαναληπτικά συνεχώς όσο ικανοποιείται μια συγκεκριμένη συνθήκη.

Σύνταξη εντολής επανάληψης for

for δείκτης = αρχική_τιμή : τελική_τιμή

<εντολές>

end (Ο βρόχος κλείνει με end)

Όπου δείκτης είναι ένας μετρητής (μεταβλητή) που παίρνει τιμές από την αρχική τιμή έως την τελική τιμή. Κάθε φορά αυξάνεται κατά μία μονάδα.

Οι εντολές μεταξύ του **for** και του **end** εκτελούνται τόσες φορές έως ότου ο δείκτης πάρει την τελική τιμή.

Παράδειγμα

Να γραφούν οι εντολές που να εμφανίζουν στην οθόνη 100 φορές τη λέξη ‘ΚΑΛΗΜΕΡΑ’

```
for i=1:5
disp(‘ΚΑΛΗΜΕΡΑ’)
end
```

Λειτουργία της for

1. Ο μετρητής στην πρώτη επανάληψη παίρνει την πρώτη τιμή από το range (i παίρνει την τιμή 1)
2. Εκτελούνται οι loopactions (τυπώνεται η τιμή του μετρητή και ο χαρακτήρας αλλαγής γραμμής)
3. Ο μετρητής παίρνει την επόμενη τιμή του range (ι παίρνει την τιμή 2)
4. ….
5. Αυτό γίνεται μέχρι να πάρει όλες τις τιμές
6. Η μεταβλητή i έχει την τελευταία τιμή

```
>> for i = 1:5
fprintf('%d\n',i)
end
1
2
3
4
5
```

Λειτουργία της for

Ο βασικός σκοπός του μετρητή είναι να μετράει τον αριθμό των επαναλήψεων

```
>> for i=1:3
fprintf('Helloo...\n');
end
```

```
Helloo...
Helloo...
Helloo...
```

Είσοδος πολλών στοιχείων

Μια χαρακτηριστική περίπτωση χρήσης της for είναι όταν έχουμε την εισαγωγή πολλών στοιχείων

Στην ουσία η εντολή input τοποθετείται μέσα στο loop action

Και στην συνέχεια τυπώνονται οι αριθμοί που έδωσε ο χρήστης

```
>>Δώσε μου έναν αριθμό 3
Έδωσες τον αριθμό 3
Δώσε μου έναν αριθμό 4
Έδωσες τον αριθμό 4
Δώσε μου έναν αριθμό 5
Έδωσες τον αριθμό 5
>>
```

Άθροισμα αριθμών

Μια συχνή διαδικασία είναι η εύρεση του αθροίσματος αριθμών

Στην ουσία χρειαζόμαστε μια μεταβλητή ***metabliti***, εκτός από τον μετρητή, που θα κρατάει το τρέχων άθροισμα

Αρχικά ***metabliti=0***

Σε κάθε βήμα

metabliti = metabliti + nea timi

```
clc
format compact
n=randi([3 7])
metabliti=0;
for i=1:n
inputnum=input('Δώσε έναν αριθμό ');
metabliti=metabliti+inputnum;
end
fprintf('Το άθροισμα είναι
%.2f\n',metabliti)
n =
     3
Δώσε έναν αριθμό 1
Δώσε έναν αριθμό 2
Δώσε έναν αριθμό 3
Το άθροισμα είναι 6.00
>>
```

Γινόμενο αριθμών

Άλλη συχνή διαδικασία είναι η εύρεση του γινομένου αριθμών

Στην ουσία χρειαζόμαστε μια μεταβλητή ***metabliti***, εκτός από τον μετρητή, που θα κρατάει το τρέχων γινόμενο.

Αρχικά ***metabliti=1***

Σε κάθε βήμα

metabliti = metabliti + nea timi

```
clc
format compact
n=randi([2 4])
metabliti=1;
for i=1:n
inputnum=input('Δώσε έναν αριθμό ');
metabliti=metabliti*inputnum;
end
fprintf('Το γινόμενο είναι %.2f\n',metabliti)
n =
    3
Δώσε έναν αριθμό 1
Δώσε έναν αριθμό 2
Δώσε έναν αριθμό 3
Το γινόμενο είναι 6.00
>>
```

Χρήσεις της for σε vectors

Συνήθως για άθροισμα χρησιμοποιείται η μεταβλητή sum, και για γινόμενο η μεταβλητή prod

- Υπολογισμός αθροίσματος στοιχείων

  ΠΡΟΣΟΧΗ αρχικοποίηση της μεταβλητής sum σε **μηδέν**
- Υπολογισμός γινομένου στοιχείων

  ΠΡΟΣΟΧΗ αρχικοποίηση της μεταβλητής prod σε **ένα**

- Υπάρχουν και οι συναρτήσεις που κάνουν την ίδια δουλειά….
- (***prod, sum, min, max, cumsum, cumprod***)

Εντολή for με βήμα

```
for δείκτης = αρχική_τιμή:βήμα:τελική_τιμή
<εντολές>
end
```

Όταν υπάρχει βήμα η μεταβολή του δείκτη είναι ανάλογη του βήματος

Παράδειγμα (Θετικό βήμα)

Να γραφτούν οι εντολές που εμφανίζουν στην οθόνη τους ζυγούς αριθμούς από 2 μέχρι 30.

```
for n=2:2:20
    disp(n)
end
```

Παράδειγμα (Αρνητικό βήμα)

```
format compact
for n=20:-2:2
        disp(n)
end
```

```
>>
kef6_for_vima
    20
    18
    16
    14
    12
    10
     8
     6
     4
     2
```

Παράδειγμα (Αρνητικό βήμα

kef6_for_vima_eisodos_for

Να γραφτούν οι εντολές που υπολογίζουν το άθροισμα 1+2+…Ν και το εμφανίζουν στην οθόνη.

```
x= input('ΠΑΡΑΚΑΛΩ ΔΩΣΤΕ ΤΟ Χ  ');
s=0;
for n=1:x
      s=s+n;
end
fprintf('ΤΟ ΑΘΡΟΙΣΜΑ ΕΙΝΑΙ =%d \n',s)
```

Πολλαπλά for

Συχνά στις εφαρμογές απαιτείται μέσα σ’ ένα βρόχο να δημιουργήσουμε ένα άλλο (εσωτερικό) βρόχο και σ’ αυτόν ένα άλλο βρόχο κ.ο.κ. Έχουμε έτσι τους λεγόμενους **πολλαπλούς** ή (**nested loops**).

Για παράδειγμα αν πρέπει να επαναλάβουμε μια διαδικασία για κάθε στοιχείο ενός m×n πίνακα, μπορούμε να σαρώσουμε τα στοιχεία του πίνακα ως εξής:

Κώδικας Πολλαπλών for

```
for i=1:m
  for j=1:n
    διαδικασία για i και j
  end
end
```

Παράδειγμα Πολλαπλών for

```
clc
% Ορισμός διαστάσεων
rows =input('Δώσε αριθμό γραμμών '); % Αριθμός γραμμών
cols =input('Δώσε αριθνό στηλών '); % Αριθμός στηλών
clc
% Δημιουργία πίνακα
result = zeros(rows, cols); % Μηδενικός πίνακας για να αποθηκεύσουμε αποτελέσματα
disp('Πίνακας result Μηδενικός πίνακας για να αποθηκεύσουμε αποτελέσματα')
disp(result)
```

```
for i = 1:rows;
    for j = 1:cols;
        result(i, j) = i * j; % Υπολογισμός: Γινόμενο των δεικτών
    end
end
% Εμφάνιση αποτελέσματος
disp('Ο πίνακας result είναι :')
disp(result)
```

Εντολή επανάληψης: while

Η **while** είναι πολύ βασική εντολή επανάληψης. Με τη while ο αριθμός των επαναλήψεων δεν είναι συγκεκριμένος, όπως με τη for, αλλά ο βρόχος επαναλαμβάνεται συνεχώς όσο ικανοποιείται μια συγκεκριμένη συνθήκη.

```
while συνθήκη
εντολές
...
end
```

Λειτουργία της while

- Όταν πρόκειται να εκτελεστεί μία while γίνονται τα παρακάτω:

1. Υπολογίζεται το condition
   1. Αν είναι true εκτελείται το action (κάτι σαν if)
2. Ξανα-υπολογίζεται το condition
   1. Αν είναι true εκτελείται το action
3. …. και συνεχίζεται μέχρι κάποια φορά το condition να είναι false

- ΠΡΟΣΟΧΗ
  - Πρέπει με κάποιο τρόπο το action να αλλάζει την αποτίμηση του condition αλλιώς έχουμε Infinite loops (ctrl + c σταματά)
  - Μπορεί να μην εκτελεστεί και ΚΑΜΙΑ φορά το action

Παρατηρήσεις στην while

- Το condition στην while πρέπει να έχει μία μεταβλητή τουλάχιστον (η οποία πρέπει να έχει πάρει τιμή πριν την πρώτη αποτίμηση του condition)
- Το action πρέπει να μεταβάλει την μεταβλητή αυτή, και συνεπώς την αποτίμηση του condition, αλλιώς έχουμε Infinite loops (ctrl + c σταματά)
- Μπορεί να μην εκτελεστεί και ΚΑΜΙΑ φορά το action

ΧΡΗΣΗ ΤΗΣ WHILE (ΠΑΡΑΔΕΙΓΜΑ)

Μία συχνή χρήση της **while** είναι ο έλεγχος της εισόδου του χρήστη

Παράδειγμα μπορούμε να ζητάμε έναν θετικό αριθμό

Γίνεται επανάληψη της prompt όσο ο χρήστης δίνει θετικό αριθμό

```
clc
inputnumber=input('Δώσε ένα θετικό αριθμό ');
while inputnumber>=0
      fprintf("Εδωσες%d\n',inputnumber)
      inputnumber=input('Δώσε ένα θετικό αριθμό ');
end
fprintf('ok\n')
```

Πολλαπλά conditions σε ένα while loop

- Σε πολλές περιπτώσεις το condition μπορεί να είναι μια πολύπλοκη έκφραση που δημιουργείται με την χρήση των τελεστών && και ||
- Για παράδειγμα

while x>=0 && x<=100

- ένα άλλο παράδειγμα

while x<50 || y<100

Καταμέτρηση της εισόδου με while

Όταν δεχόμαστε είσοδο με **while** δεν γνωρίζουμε εκ των προτέρων πόσους αριθμούς θα δώσει ο χρήστης.

Μπορούμε όμως να μετρήσουμε πόσους έδωσε για να το γνωρίζουμε στην συνέχεια

Αυτό γίνεται με την χρήση ενός μετρητή που μετράει τις επαναλήψεις ενός while

```
clc
format compact
c=0;
inputnumber=input(Δώσε ένα θετικό αριθμό');
while inputnumber >= 0;
    fprintf('Έδωσες τον αριθμό%d\n',inputnumber);
    c=c+1;
    inputnumber=input(Δώσε ένα θετικό αριθμό');
end
fprintf ('Έδωσες θετικό αριθμό %d\n',c')
```

Άθροισμα και γινόμενο vector

Για να εκτελέσουμε πολλαπλασιασμό ενός vector με ένα αριθμό μπορούμε να επεξεργαστούμε ένα ένα τα στοιχεία με for

Το ίδιο κάνει και ο τελεστής *

```
>> vec=[3 5 3 2];
>> for i=1:length(vec)
vec(i)=vec(i)*3;
end
>> vec
vec =
     9    15     9     6
>>
```

Χρήσιμες συναρτήσεις για vectors / matrices

- ***sum*** και ***prod*** βρίσκουν το άθροισμα/γινόμενο για κάθε στοιχείο μιας γραμμής ή στήλης ενός πίνακα.
- ***min*** και ***max*** επιστρέφουν το μεγαλύτερο/ μικρότερο στοιχείο του vector ή ανά column στα matrices
- **cumsum** και **cumprod** . Βρίσκουν το αθροιστικό άθροισμα και το αθροιστικό γινόμενο ενός πίνακα

Παράδειγμα αθροιστικό (συνολικό) άθροισμα

Να γραφτούν οι εντολές που διαβάζουν ένα σύνολο αριθμών και υπολογίζουν το άθροισμα των τετραγώνων τους. Το τέλος του συνόλου των αριθμών να δηλώνεται με κάποιο συγκεκριμένο αριθμό πχ(100).

```
clc
format compact
x=input('Παρακαλώ δώσε έναν αριθμό ');
s=0;
while x~=100
   s=s+x^2;
   x=input('Παρακαλώ δώσε έναν αριθμό ');
end
fprintf('Το άθροισμα των τετραγώνων είναι %d',s)
```

Εντολή **break.** Τερματίζει ένα βρόχο πρόωρα

```
for i=1:10
   if i==5
      disp('Βρέθηκε το 5, τερματίζω το βρόχο ‘);
      break; % Τερματισμός του βρόχου όταν i==5
   end
   disp(['i=',num2str(i)]);
end
```

Εντολή **continue.** Παραλείπει την τρέχουσα επανάληψη του βρόχου και προχωρά στην επόμενη

```
for i= 1:5
   if i==3
      disp('Παράλειψη του 3.’);
      continue; % Παράληψη του υπόλοιπου κώδικα αυτής της επανάληψης
   end
   disp(['i= ',num2str(i)]);
end
```

Συνάρτηση menu

Με τη συνάρτηση menu δημιουργείται ένα παράθυρο γραφικών με κουμπιά επιλογής.

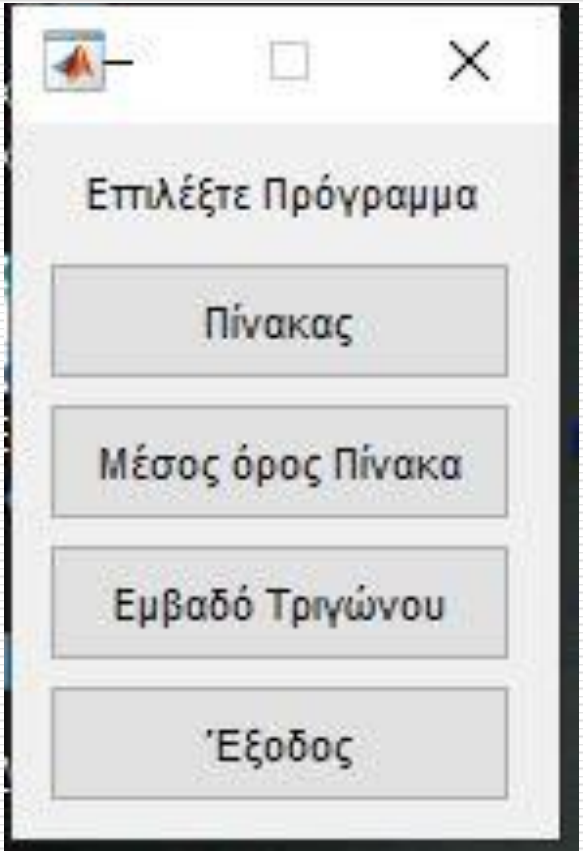
Γεν. τύπος: type=menu(header, item1, item2, item3,..)

Header δημιουργεί επικεφαλίδα

Item1, item2, item3,… δημιουργούν κουμπιά με τα αντίστοιχα ονόματα.

Ο αριθμός επιλογής του χρήστη επιστρέφεται στη μεταβλητή choice.

ΓΡΑΦΙΚΟ ΜΕΝU ΕΠΙΛΟΓΗΣ



Επιλογή ‘Πίνακας’

Πατώντας το πλήκτρο πίνακας:

Να βρεθούν η ορίζουσα, ο ανάστροφος και ο αντίστροφος του πίνακα a=[2 3 4;9 6 4;2 9 3]

Δημιουργία 1ου προγράμματος

```
a=[2 3 4;9 6 4;2 9 3];
disp('h orizoyza toy pinaka=‘)
d=det(a)
disp('o anastrofos toy pinaka =‘)
at=a’
disp('o antistrofos toy pinaka=‘)
al=inv(a)
pause
main %epistrofh sto arxiko programma
```

Δημιουργία 2ου προγράμματος

```
a=[4 5 8 5 9]
s=0;
for i=1:5
    s=s+a(i);
end
mesos1=s/5
disp('mesos oros=')
disp(mesos1)
pause
main
```

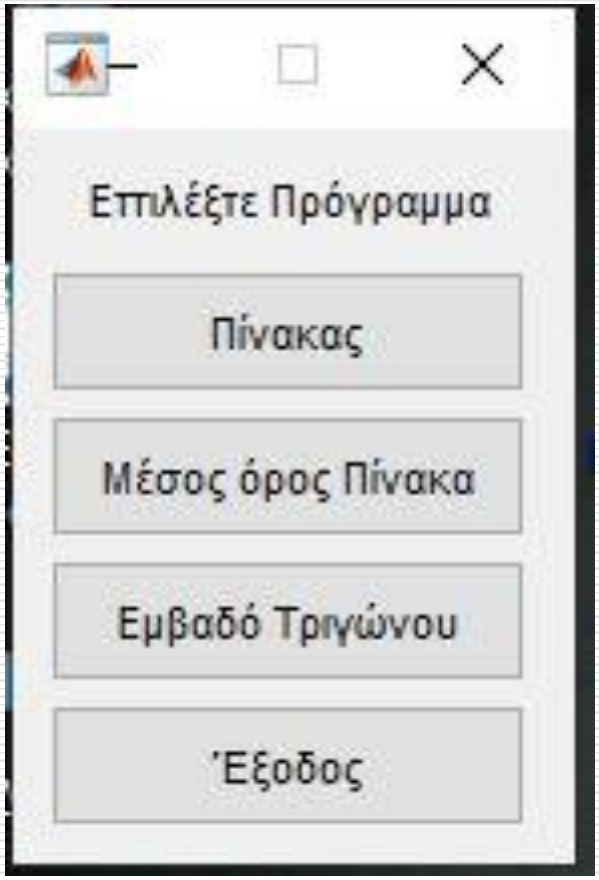
Δημιουργία 3ου προγράμματος

```
a=input('dose a=');
b=input('dose b=');
c=input('dose c=‘);
s=(a+b+c)/2;
e=sqrt(s*(s-a)*(s-b)*(s-c));
disp('embadon=‘)
disp(e)
pause
main
```

Δημιουργία menu

```
clc
type=menu('Επιλέξτε Πρόγραμμα','Πίνακας','Μέσος όρος Πίνακα','Εμβαδό Τριγώνου','Έξοδος')
if type==1
    clc
    proto_programma;
    pause
    main
elseif type==2
    clc
    deytero_programma;
    pause
    main
elseif type==3
    clc
    trito_programma;
    pause
    main
else
    clc
    disp('Exit')
    pause
end
```

Με την εκτέλεση του προγράμματος main θα εμφανισθεί ο παρακάτω πίνακας επιλογών



Όταν εκτελεστεί κάποια διαδικασία, επιστρέφουμε στο παράθυρο του Μενού μας πατώντας space-bar