ΠΛΗ10

ΕΝΟΤΗΤΑ 2: Αλγόριθμοι και Ψευδογλώσσα

Μάθημα 2.4:
Εντολές Επανάληψης

Δημήτρης Ψούνης

www.psounis.gr

Περιεχόμενα Μαθήματος

Α. Θεωρία

1. Εντολές Επανάληψης

1. Γενικά

- Η δομή επανάληψης είναι η σημαντικότερη δομή σε ένα πρόγραμμα.
  - ...διότι μας επιτρέπει να εκτελέσουμε ένα τμήμα προγράμματος πολλές φορές, το οποίο είναι το κύριο χαρακτηριστικό του προγραμματισμου.
- Στην **ψευδογλώσσα** που μαθαίνουμε, υπάρχουν δύο τρόποι για να κάνουμε επανάληψη της εκτέλεσης ενός τμήματος κώδικα:
  - Η δομή **ΓΙΑ...ΕΩΣ...ΕΠΑΝΑΛΑΒΕ** στην οποία ρητά αναφέρουμε πόσες φορές θέλουμε να εκτελεστεί ένα τμήμα κώδικα.
  - Η δομή **ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ(Συνθήκη);**
  - Η δομή **ΕΝΟΣΩ (συνθηκη) ΕΠΑΝΑΛΑΒΕ...ΕΝΟΣΩ-ΤΕΛΟΣ**
- Θα αναλύσουμε τους τρεις τρόπους επανάληψης και πότε χρησιμοποιούμε τον καθένα

Α. Θεωρία

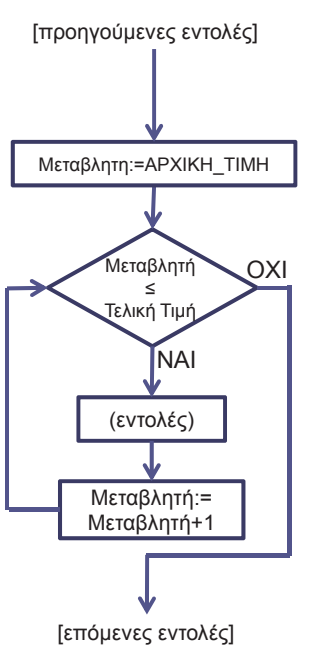
1. Εντολές Επανάληψης

2. Εντολή ΓΙΑ…ΕΩΣ…ΕΠΑΝΑΛΑΒΕ

- Το συντακτικό της δομής ΓΙΑ...ΕΩΣ...ΕΠΑΝΑΛΑΒΕ είναι:

```
ΓΙΑ (Μεταβλητη:=ΑΡΧΙΚΗ_ΤΙΜΗ) ΕΩΣ (ΤΕΛΙΚΗ_ΤΙΜΗ)
ΕΠΑΝΑΛΑΒΕ
   (Εντολή ή εντολές)
ΓΙΑ-ΤΕΛΟΣ
```

- Όπου η Μεταβλητή (πάντα ακέραια) θα πάρει διαδοχικά όλες τις τιμές από ΑΡΧΙΚΗ_ΤΙΜΗ έως ΤΕΛΙΚΗ_ΤΙΜΗ και για κάθε τιμή από αυτές θα εκτελεστεί η εντολή (ή οι εντολές)
- Για παράδειγμα αν ΑΡΧΙΚΗ_ΤΙΜΗ=1 και ΤΕΛΙΚΗ_ΤΙΜΗ=5 θα εκτελεστούν οι εντολές για:
  - Μεταβλητή=1
  - Μεταβλητή=2
  - Μεταβλητή=3
  - Μεταβλητή=4
  - Μεταβλητή=5



Α. Θεωρία
1. Εντολές Επανάληψης
2. Εντολή ΓΙΑ…ΕΩΣ…ΕΠΑΝΑΛΑΒΕ

- Παράδειγμα 1:
  - Τι κάνει το ακόλουθο τμήμα κώδικα;

```
ΓΙΑ I:=1 ΕΩΣ 5 ΕΠΑΝΑΛΑΒΕ
   ΤΥΠΩΣΕ ("Καλημέρα");
ΓΙΑ-ΤΕΛΟΣ
```

  - Απάντηση: Οι εντολές που περικλείονται ανάμεσα στο ΓΙΑ και το ΓΙΑ-ΤΕΛΟΣ εκτελούνται για κάθε τιμή του Ι. Άρα:
    - Για Ι=1 τυπώνεται «Καλημέρα»
    - Για Ι=2 τυπώνεται «Καλημέρα»
    - Για Ι=3 τυπώνεται «Καλημέρα»
    - Για Ι=4 τυπώνεται «Καλημέρα»
    - Για Ι=5 τυπώνεται «Καλημέρα»
  - Και το πρόγραμμα τερματίζει
  - Άρα το πρόγραμμα τυπώνει 5 φορές στην οθόνη τη λέξη ΚΑΛΗΜΕΡΑ

Α. Θεωρία
1. Εντολές Επανάληψης
2. Εντολή ΓΙΑ…ΕΩΣ…ΕΠΑΝΑΛΑΒΕ

- Παράδειγμα 2:
  - Τι κάνει το ακόλουθο τμήμα κώδικα;

```
ΓΙΑ I:=1 ΕΩΣ 5 ΕΠΑΝΑΛΑΒΕ
   ΤΥΠΩΣΕ (I);
ΓΙΑ-ΤΕΛΟΣ
```

  - Απάντηση: Οι εντολές που περικλείονται ανάμεσα στο ΓΙΑ και το ΓΙΑ-ΤΕΛΟΣ εκτελούνται για κάθε τιμή του Ι. Άρα:
    - Για Ι=1 τυπώνεται «1»
    - Για Ι=2 τυπώνεται «2»
    - Για Ι=3 τυπώνεται «3»
    - Για Ι=4 τυπώνεται «4»
    - Για Ι=5 τυπώνεται «5»
  - Και το πρόγραμμα τερματίζει

Συνήθως στις εφαρμογές χρησιμοποιείται η τιμή της μεταβλητής. Είναι σημαντικό ότι σε κάθε επανάληψη η τιμή της μεταβλητής είναι διαφορετική

Α. Θεωρία
1. Εντολές Επανάληψης
2. Εντολή ΓΙΑ…ΕΩΣ…ΕΠΑΝΑΛΑΒΕ

- Παράδειγμα 3:
  - Τι κάνει το ακόλουθο τμήμα κώδικα;

```
ΓΙΑ I:=1 ΕΩΣ 5 ΕΠΑΝΑΛΑΒΕ
   X:=I*I;
   ΤΥΠΩΣΕ(X);
ΓΙΑ-ΤΕΛΟΣ
```

  - Απάντηση: Οι εντολές που περικλείονται ανάμεσα στο ΓΙΑ και το ΓΙΑ-ΤΕΛΟΣ εκτελούνται για κάθε τιμή του Ι. Άρα:
    - Για Ι=1 υπολογίζεται το Χ=1*1=1 και τυπώνεται «1»
    - Για Ι=2 υπολογίζεται το Χ=2*2=4 και τυπώνεται «4»
    - Για Ι=3 υπολογίζεται το Χ=3*3=9 και τυπώνεται «9»
    - Για Ι=4 υπολογίζεται το Χ=4*4=16 και τυπώνεται «16»
    - Για Ι=5 υπολογίζεται το Χ=5*5=25 και τυπώνεται «25»
  - Και το πρόγραμμα τερματίζει

Συνήθως βέβαια, κάνουμε τον υπολογισμό που μας ενδιαφέρει στις εντολές μεταξύ των ΓΙΑ...ΓΙΑ-ΤΕΛΟΣ χρησιμοποιώντας την τιμή της μεταβλητής του βρόχου.

Α. Θεωρία
1. Εντολές Επανάληψης
2. Εντολή ΓΙΑ…ΕΩΣ…ΕΠΑΝΑΛΑΒΕ

- Επέκταση της εντολής για διαφορετικό βήμα αύξησης:

```
ΓΙΑ (Μεταβλητη:=ΑΡΧΙΚΗ_ΤΙΜΗ) ΕΩΣ (ΤΕΛΙΚΗ_ΤΙΜΗ) ΜΕ-ΒΗΜΑ (βήμα)
ΕΠΑΝΑΛΑΒΕ
   (Εντολή ή εντολές)
ΓΙΑ-ΤΕΛΟΣ
```

  - Η δομή εκτελείται όπως πριν, αλλά το βήμα αύξησης είναι
    Μεταβλητή:=Μεταβλητή+Βήμα

- Παράδειγμα:

```
ΓΙΑ I:=1 ΕΩΣ 5 ΜΕ-ΒΗΜΑ 2 ΕΠΑΝΑΛΑΒΕ
   X:=I*I;
   ΤΥΠΩΣΕ(X);
ΓΙΑ-ΤΕΛΟΣ
```

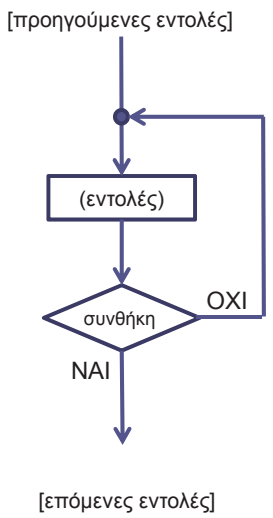
  - Απάντηση: Για Ι=1 υπολογίζεται το Χ=1*1=1 και τυπώνεται «1»
    - Για Ι=3 υπολογίζεται το Χ=3*3=9 και τυπώνεται «9»
    - Για Ι=5 υπολογίζεται το Χ=5*5=25 και τυπώνεται «25»

Α. Θεωρία
1. Εντολές Επανάληψης
3. Εντολή ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ

- Η δομή ΕΠΑΝΑΛΑΒΕ ... ΜΕΧΡΙ συντάσσεται ως ακολούθως

```
ΕΠΑΝΑΛΑΒΕ
   (Εντολή ή Εντολές)
ΜΕΧΡΙ (Συνθήκη)
```

- Στην δομή αυτή:
  - Εκτελούνται οι εντολές
  - Ελέγχεται η συνθήκη (την οποία έχουμε συντάξει με λογικούς και σχεσιακούς τελεστές).
    - Αν η απάντηση είναι false (δεν ισχύει η συνθήκη), τότε ξαναρχίζουμε από την αρχή (πρώτη εντολή μετά από το ΕΠΑΝΑΛΑΒΕ)
    - Αν η απάντηση είναι true (ισχύει η συνθήκη), τότε βγαίνουμε από την επανάληψη και τρέχουμε την αμέσως επόμενη εντολή μετά τη δομή επανάληψης



Α. Θεωρία
1. Εντολές Επανάληψης
3. Εντολή ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ

- Παράδειγμα 1:
- Τι κάνει το ακόλουθο τμήμα κώδικα;

```
Ι:=0;
ΕΠΑΝΑΛΑΒΕ
   Ι:=Ι+1;
   ΤΥΠΩΣΕ(Ι);
ΜΕΧΡΙ (Ι=3)
```

- Αρχικά Ι=0
  - Γίνεται επανάληψη:
    - Τίθεται Ι:=0+1=1 και τυπώνεται «1»
    - Γίνεται έλεγχος συνθήκης. 1=3 είναι false, άρα θα επαναλάβουμε
  - Γίνεται επανάληψη:
    - Τίθεται Ι:=1+1=2 και τυπώνεται «2»
    - Γίνεται έλεγχος συνθήκης. 2=3 είναι false, άρα θα επαναλάβουμε
  - Γίνεται επανάληψη:
    - Τίθεται Ι:=2+1=3 και τυπώνεται «3»
    - Γίνεται έλεγχος συνθήκης. 3=3 είναι true, άρα δεν θα επαναλάβουμε

Α. Θεωρία
1. Εντολές Επανάληψης
3. Εντολή ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ

- Παράδειγμα 2:
- Τι κάνει το ακόλουθο τμήμα κώδικα;

```
Ι:=5;
ΕΠΑΝΑΛΑΒΕ
   Ι:=Ι+1
ΜΕΧΡΙ (Ι=3)
```

- Αρχικά Ι=5
  - Γίνεται επανάληψη:
    - Τίθεται Ι:=5+1=6
    - Γίνεται έλεγχος συνθήκης. 6=3 είναι false, άρα θα επαναλαβουμε
  - Γίνεται επανάληψη:
    - Τίθεται Ι:=6+1=7
    - Γίνεται έλεγχος συνθήκης. 7=3 είναι false, άρα θα επαναλαβουμε....
  - ..... Θα τερματίσει ποτέ το πρόγραμμα?
    - Η απάντηση είναι ΟΧΙ! Σε κάθε επανάληψη το Ι θα αυξάνεται κατά 1, άρα ποτέ δεν θα γίνει ίσο με 3!

Αυτό είναι παράδειγμα κακού προγραμματισμού, αποτελεί ένα από τα σημαντικότερα προγραμματιστικά λάθη και καλείται **ατέρμων βρόχος** (επανάληψη που δεν ολοκληρώνεται ποτέ)

Α. Θεωρία
1. Εντολές Επανάληψης
3. Εντολή ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ

- Παρατηρήσεις για την δομή ΕΠΑΝΑΛΑΒΕ-ΜΕΧΡΙ
- Η σύνταξη της δομής είναι αρκετά απλή, αλλά θα πρέπει εμείς, ως προγραμματιστές να συντάξουμε σωστά τις υπόλοιπες εντολές. Συγκεκριμένα:
  - Πρέπει να αρχικοποιήσουμε σωστά την μεταβλητή που θα έχουμε στην εντολή συνθήκης. Έτσι πριν την εντολή ΕΠΑΝΑΛΑΒΕ θα πρέπει να αρχικοποιήσουμε την μεταβλητή που θα χρησιμοποιήσουμε (εντολή αρχικοποίησης)
  - Πρέπει η μεταβλητή που έχουμε στην εντολή αρχικοποίησης να επηρεάζεται στο σώμα της επανάληψης (εντολή αύξησης μεταβλητής)
  - Σχηματικά:

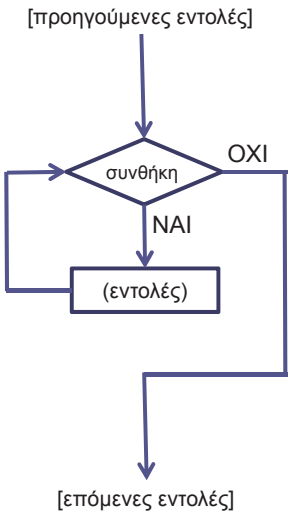
```
Ι:=0;                  <- Εντολή Αρχικοποίησης
ΕΠΑΝΑΛΑΒΕ
   (εντολή ή εντολές)
   Ι:=Ι+1              <- Εντολή αύξησης μεταβλητής
ΜΕΧΡΙ (Ι=3)            <- Συνθήκη
```

Α. Θεωρία
1. Εντολές Επανάληψης
4. Εντολή ΕΝΟΣΩ…ΕΝΟΣΩ ΤΕΛΟΣ

- Η δομή ΕΝΟΣΩ...ΕΝΟΣΩ-ΤΕΛΟΣ είναι ίδια με την ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ με την διαφορά ότι ο έλεγχος γίνεται στην αρχή της επανάληψης και όχι στο τέλος της επανάληψης

```
ΕΝΟΣΩ (συνθήκη) ΕΠΑΝΑΛΑΒΕ
        ... (ακολουθία εντολών)
ΕΝΟΣΩ-ΤΕΛΟΣ
```

- Στην δομή αυτή:
  - Ελέγχεται η συνθήκη (την οποία έχουμε συντάξει με λογικούς και σχεσιακούς τελεστές).
    - Αν η απάντηση είναι true (ισχύει η συνθήκη), εκτελούνται οι εντολές και επαναλαμβάνουμε από την αρχή.
    - Αν η απάντηση είναι false (δεν ισχύει η συνθήκη), τότε βγαίνουμε από την επανάληψη και τρέχουμε την αμέσως επόμενη εντολή μετά τη δομή επανάληψης.



Α. Θεωρία
1. Εντολές Επανάληψης
4. Εντολή ΕΝΟΣΩ…ΕΝΟΣΩ ΤΕΛΟΣ

- Παράδειγμα 1:
  - Τι κάνει το ακόλουθο τμήμα κώδικα;

```
K:=5;
ΕΝΟΣΩ (K<8) ΕΠΑΝΑΛΑΒΕ
   L:=2*K+1;
   K:=K+1;
   ΤΥΠΩΣΕ (L);
ΕΝΟΣΩ-ΤΕΛΟΣ
```

  - Απάντηση:
    - Αρχικοποιείται το Κ με 5
    - Γίνεται ο έλεγχος συνθήκης (5<8). Επιστρέφεται true, άρα γίνονται τα βηματα L=2*5+1=11 και K=5+1=6. Τυπώνεται «11».
    - Γίνεται ο έλεγχος συνθήκης (6<8). Επιστρέφεται true, άρα γίνονται τα βηματα L=2*6+1=13 και K=6+1=7. Τυπώνεται «13».
    - Γίνεται ο έλεγχος συνθήκης (7<8). Επιστρέφεται true, άρα γίνονται τα βηματα L=2*7+1=15 και K=7+1=8. Τυπώνεται «15».
    - Γίνεται ο έλεγχος συνθήκης (8<8). Επιστρέφεται false, άρα τερματίζει η επανάληψη.

Α. Θεωρία
1. Εντολές Επανάληψης
5. Χρήση των 3 Εντολών Επανάληψης

- Βλέπουμε ότι και οι 3 δομές επανάληψης με παρόμοιο τρόπο κάνουν τις ίδιες ενέργειες.
  - Η πιο συνηθισμένη δομή είναι η **ΓΙΑ...ΓΙΑ-ΤΕΛΟΣ** και την χρησιμοποιούμε όταν ξέρουμε ποιες τιμές θα πάρει η μεταβλητή.
  - Αν δεν ξέρουμε ακριβώς ποιες τιμές θα πάρει ή μεταβλητή ή πόσες φορές πρέπει να γίνει η επανάληψη, τότε χρησιμοποιούμε την δομή **ΕΝΟΣΩ...ΕΝΟΣΩ-ΤΕΛΟΣ**.
  - Η δομή **ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙ** χρησιμοποιείται πιο σπάνια.
    - Εμείς θα την χρησιμοποιούμε μόνο για λόγους αμυντικού προγραμματισμού (θα το δούμε στη συνέχεια)
- Ειδικά οι δομές ΕΝΟΣΩ..ΕΝΟΣΩ ΤΕΛΟΣ και ΕΠΑΝΑΛΑΒΕ..ΜΕΧΡΙ μοιάζουν αρκετά
  - Μία σημαντική διαφορά είναι ότι η ΕΠΑΝΑΛΑΒΕ…ΜΕΧΡΙ θα τρέξει σίγουρα τουλάχιστον μία φορά, το οποίο δεν ισχύει απαραίτητα για την ΕΝΟΣΩ…ΕΝΟΣΩ ΤΕΛΟΣ

Α. Θεωρία
1. Εντολές Επανάληψης
6. Σχέση των 3 Εντολών Επανάληψης

- Ενδιαφέρον επίσης έχει ότι η εντολή ΓΙΑ μπορεί να προσομοιωθεί από τις άλλες δύο ως ακολούθως:

```
ΓΙΑ I:=1 ΕΩΣ 10 ΕΠΑΝΑΛΑΒΕ
   (Εντολή ή εντολές)
ΓΙΑ-ΤΕΛΟΣ
```

- Με εντολή ΕΝΟΣΩ… ΕΝΟΣΩ ΤΕΛΟΣ

```
I:=1;
ΕΝΟΣΩ (I<=10) ΕΠΑΝΑΛΑΒΕ
   (Εντολή ή εντολές)
   I:=I+1;
ΕΝΟΣΩ-ΤΕΛΟΣ
```

- Με εντολή ΕΠΑΝΑΛΑΒΕ..ΩΣΠΟΥ

```
I:=0;
ΕΠΑΝΑΛΑΒΕ
   I:=I+1;
   (Εντολή ή εντολές)
ΜΕΧΡΙ (I=10)
```

Α. Θεωρία
1. Εντολές Επανάληψης
7. Αμυντικός Προγραμματισμός

- Μία καλή στρατηγική στον προγραμματισμό είναι πάντα να ελέγχουμε την είσοδο του χρήστη αν ικανοποιεί τις προδιαγραφές του προγράμματός μας.
- Π.χ. Αν θέλουμε να διαβάσουμε έναν ακέραιο μεταξύ 1 και 100 και πρέπει να αποφύγουμε ο χρήστης να εισάγει μία λανθασμένη τιμή τότε εφαρμόζουμε **αμυντικό προγραμματισμό** και κάνουμε έλεγχο αν η τιμή που εισήγαγε ο χρήστης είναι σωστή.
- Έτσι τον αναγκάζουμε να επαναπληκτρολογήσει μέχρι να βάλει την σωστή τιμή.
- Στο ακόλουθο πρόγραμμα κάνουμε αμυντικό προγραμματισμό για να διαβάσουμε έναν αριθμό από το 1 έως το 100.

```
ΑΛΓΟΡΙΘΜΟΣ defensive
ΔΕΔΟΜΕΝΑ
   x: INTEGER;
ΑΡΧΗ

   ΕΠΑΝΑΛΑΒΕ
      ΤΥΠΩΣΕ("Δώσε τον αριθμό: ");
      ΔΙΑΒΑΣΕ(x);
   ΜΕΧΡΙ (x>=0 AND x<=100)

ΤΕΛΟΣ
```


Α. Θεωρία
1. Εντολές Επανάληψης
8. Γενικά Σχόλια για το Συντακτικό των Εντολών Επανάληψης

- Στην ψευδογλώσσα του ΕΑΠ μια εντολή επανάληψης νοείται ως μια ακόμη εντολή. Έτσι σεβόμαστε το συντακτικό που έχουμε πει για τις εντολές, δηλαδή ότι κάθε εντολή πρέπει να τελειώνει με ερωτηματικό.
  - Όταν γράφουμε «στο χέρι» κώδικα θα πρέπει να τελειώνουμε μια εντολή με ερωτηματικό.
  - Όταν γράφουμε μια εντολή επανάληψης στον μεταγλωττιστή δεν θα την τελειώνουμε με ερωτηματικό.
- Επίσης για ένα μπλοκ κώδικα, ισχύει ότι όταν γράφουμε κώδικα με «το χέρι» η τελευταία εντολή δεν πρέπει να έχει ερωτηματικό. Μπλοκ κώδικα, δηλαδή εντολές που ομαδοποιούνται έχουμε δει για την ώρα στα εξής σημεία:
  - ΑΡΧΗ [μπλοκ] ΤΕΛΟΣ του αλγορίθμου
  - ΕΑΝ ..ΤΟΤΕ [μπλοκ] ΕΑΝ-ΤΕΛΟΣ
  - ΕΑΝ..ΤΟΤΕ [μπλοκ] ΑΛΛΙΩΣ [μπλοκ] ΕΑΝ-ΤΕΛΟΣ
  - ΓΙΑ..ΕΩΣ..ΕΠΑΝΑΛΑΒΕ [μπλοκ] ΓΙΑ-ΤΕΛΟΣ
  - ΕΠΑΝΑΛΑΒΕ [μπλοκ] ΜΕΧΡΙ ..
  - ΕΝΟΣΩ..ΕΠΑΝΑΛΑΒΕ [μπλοκ] ΕΝΟΣΩ-ΤΕΛΟΣ
- Στην ακόλουθη διαφάνεια φαίνεται ένα πρόγραμμα και οι μικροαλλαγές που γίνονται στη χρήση ερωτηματικών όταν το γράφουμε με το χέρι και όταν το γράφουμε στο μεταγλωττιστή.

Α. Θεωρία
1. Εντολές Επανάληψης
8. Γενικά Σχόλια για το Συντακτικό των Εντολών Επανάληψης

Στον μεταγλωττιστή

```
ΑΛΓΟΡΙΘΜΟΣ example
ΔΕΔΟΜΕΝΑ
   I,N,X: INTEGER;
ΑΡΧΗ
   ΤΥΠΩΣΕ("ΔΩΣΕ N: ");
   ΔΙΑΒΑΣΕ(N);
   ΤΥΠΩΣΕ("ΑΡΧΗ",EOLN);
   ΓΙΑ I:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
      X:=I*I;
      ΤΥΠΩΣΕ(X,EOLN);
   ΓΙΑ-ΤΕΛΟΣ
   ΤΥΠΩΣΕ("ΤΕΛΟΣ");
ΤΕΛΟΣ
```

Με το χέρι

```
ΑΛΓΟΡΙΘΜΟΣ example
ΔΕΔΟΜΕΝΑ
   I,N,X: INTEGER;
ΑΡΧΗ
   ΤΥΠΩΣΕ("ΔΩΣΕ N: ");
   ΔΙΑΒΑΣΕ(N);
   ΤΥΠΩΣΕ("ΑΡΧΗ",EOLN);
   ΓΙΑ I:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
      X:=I*I;
      ΤΥΠΩΣΕ(X,EOLN)    <= ΠΡΟΣΟΧΗ!
   ΓΙΑ-ΤΕΛΟΣ;           <= ΠΡΟΣΟΧΗ!
   ΤΥΠΩΣΕ("ΤΕΛΟΣ")      <= ΠΡΟΣΟΧΗ!
ΤΕΛΟΣ
```

Σημείωση:

- Στον μεταγλωττιστή θα περάσουν και οι δύο τρόποι. Και με τους δύο τρόπους θα μεταγλωττιστεί το πρόγραμμα.
- Ωστόσο αποτελεί ισχυρή σύσταση να επιλέξετε να σκέφτεστε με το 1ο τρόπο. Στις συνήθεις γλώσσες προγραμματισμού (όπως η C που θα κάνουμε) ισχύει ο 1ος τρόπος. Οπότε είναι αχρείαστο να εργαζόμαστε με το 2ο τρόπο.

Β.Ασκήσεις
Εφαρμογή 1: Ελάχιστος Ν αριθμών

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό Ν μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει Ν αριθμούς.
- Το πρόγραμμα τυπώνει τον ελάχιστο από τους Ν αριθμούς που εισήγαγε ο χρήστης.

Σημείωση:

- Η άσκηση αυτή είναι αυξημένης δυσκολίας για αρχάριους προγραμματιστές διότι εισάγει και έναν προγραμματιστικό τρόπο σκέψης. Συμβουλευθείτε με μεγάλη προσοχή τη λύση, και προχωρήστε με τις επόμενες ασκήσεις

Β.Ασκήσεις

Εφαρμογή 2: Μέγιστος Ν αριθμών

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό Ν μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει Ν ακέραιους αριθμούς.
- Το πρόγραμμα υπολογίζει και τυπώνει το μέγιστο των αριθμών που εισήγαγε ο χρήστης.

Β.Ασκήσεις

Εφαρμογή 3: Άθροισμα Ν αριθμών

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό Ν μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει Ν ακέραιους αριθμούς.
- Το πρόγραμμα υπολογίζει και τυπώνει το άθροισμα των αριθμών που εισήγαγε ο χρήστης.

Β.Ασκήσεις

Εφαρμογή 4: Γινόμενο Ν αριθμών

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) που:

- Ζητάει από το χρήστη να εισάγει έναν αριθμό Ν μεταξύ του 5 και του 10 με αμυντικό προγραμματισμό
- Έπειτα ζητάει από το χρήστη να εισάγει Ν ακέραιους αριθμούς.
- Το πρόγραμμα υπολογίζει και τυπώνει το γινόμενο των αριθμών που εισήγαγε ο χρήστης.

.

Β.Ασκήσεις

Εφαρμογή 5: Εμφωλιασμένοι Βρόχοι 1

Τι τυπώνει το ακόλουθο πρόγραμμα; Μελετήστε το, τόσο με «χαρτί και μολύβι» όσο και στον μεταγλωττιστή. Τροποποιήστε την έξοδο του προγράμματος ώστε να είναι κατά το δυνατόν «κομψή», ή τουλάχιστον κομψότερη από την εκτύπωση που κάνει το πρόγραμμα αυτό.

```
ΑΛΓΟΡΙΘΜΟΣ efarmogi5
ΔΕΔΟΜΕΝΑ
   I,J: INTEGER;
ΑΡΧΗ
   ΓΙΑ I:=1 ΕΩΣ 4 ΕΠΑΝΑΛΑΒΕ
      ΓΙΑ J:=1 ΕΩΣ 5 ΕΠΑΝΑΛΑΒΕ
         ΤΥΠΩΣΕ(I+J);
      ΓΙΑ-ΤΕΛΟΣ
   ΓΙΑ-ΤΕΛΟΣ
ΤΕΛΟΣ
```

Σημείωση:

- Επειδή έχουμε επανάληψη μέσα στην επανάληψη, η παραπάνω δομή χαρακτηρίζεται προγραμματιστικά **«εμφωλιασμένοι βρόχοι»**

Β.Ασκήσεις

Εφαρμογή 6: Εμφωλιασμένοι Βρόχοι 2

(Α) Τι τυπώνει το ακόλουθο πρόγραμμα; Μελετήστε το, τόσο με «χαρτί και μολύβι» όσο και στον μεταγλωττιστή

```
ΑΛΓΟΡΙΘΜΟΣ efarmogi6
ΔΕΔΟΜΕΝΑ
   I,J,N: INTEGER;
ΑΡΧΗ
   N:=10;
   ΓΙΑ I:=1 ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
      ΓΙΑ J:=I ΕΩΣ N ΕΠΑΝΑΛΑΒΕ
         ΤΥΠΩΣΕ("*");
      ΓΙΑ-ΤΕΛΟΣ
      ΤΥΠΩΣΕ(EOLN);
   ΓΙΑ-ΤΕΛΟΣ
ΤΕΛΟΣ
```

(Β) Κάντε κατάλληλη μετατροπή στο πρόγραμμα ώστε το τρίγωνο να τυπώνεται ανάποδα (δηλαδή η βάση του τριγώνου να εμφανίζεται στο τέλος και όχι στην αρχή της εκτέλεσης.

Β.Ασκήσεις

Εφαρμογή 7: Αμυντικός Προγραμματισμός

Γράψτε ένα πρόγραμμα (χρησιμοποιώντας τον μεταγλωττιστή) για εξάσκηση στον αμυντικό προγραμματισμό που θα διαβάζει 5 ακεραίους αριθμούς X,Y,Z,W

- Ο X θα πρέπει να είναι μεταξύ του 5 και του 10
- Ο Y θα πρέπει να είναι θετικός
- Ο Z θα πρέπει να είναι μη αρνητικός
- Ο W θα πρέπει να είναι αρνητικός.