

Προγραμματισμός I (HY120)

Διάλεξη 1:
Εισαγωγή



Ποιος είμαι εγώ!



2



Ναύπλιο, 4/1976-9/1993



Πάτρα, 9/1993-6/2004



Williamsburg, VA,
USA, 7/2004-7/2006
Χρήστος Δ. Αντωνόπουλος
24/9/2018



Μυτιλήνη, 10/2006-2/2007



Βόλος, 2/2007 - ...

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ο Υπεύθυνος των Εργαστηρίων

- Γιώργος Θάνος



3



Αθήνα (1976-1994)



Αθήνα (2000-2009)



Θεσσαλονίκη (1994-2000)



Βόλος (2009-σήμερα)

Η 2η Υπεύθυνη των Εργαστηρίων

- Βάνα Ντουφεξή



4



Πάτρα (1972-1995)



Urbana-Champaign (UIUC) (1995-2000)



Chicago (Northwestern) (2000-2007)



Βόλος, 10/2007 - ...

Οι Βοηθοί των Εργαστηρίων



5





Ποιοι είστε εσείς;

- Υπόβαθρο;
 - Έχετε δει υπολογιστή;
 - Έχετε ανοίξει υπολογιστή;
 - Έχετε «**προγραμματίσει**» υπολογιστή;
- Τι φαντάζεστε για το μάθημα;
 - Τι περιμένετε να μάθετε;
 - Τι απαιτήσεις φαντάζεστε ότι υπάρχουν;



Υπολογιστές, Υπολογιστές ... και Υπολογιστές

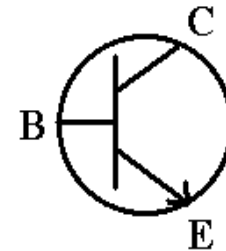
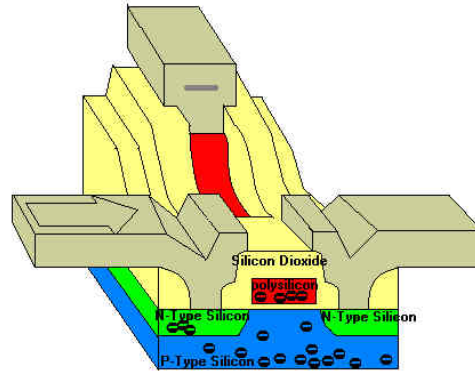
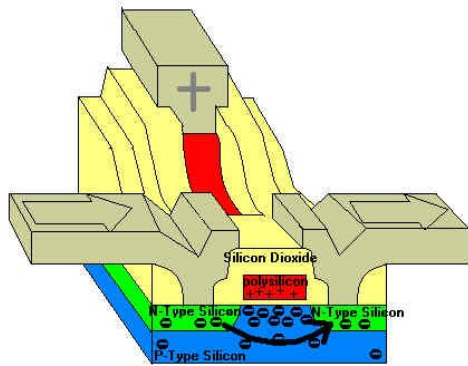


7

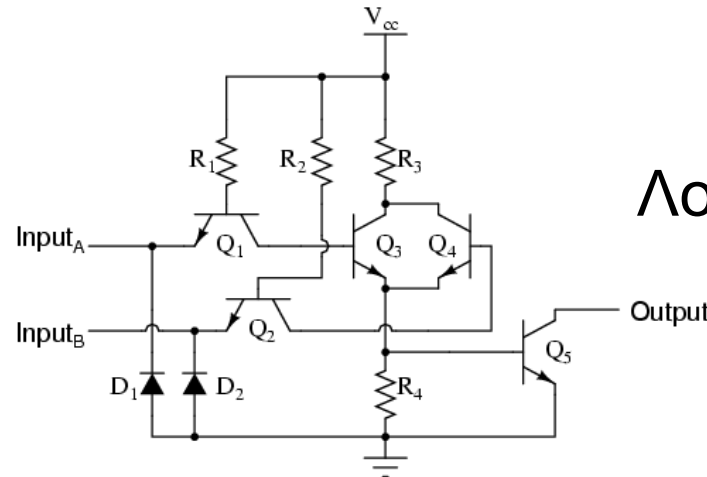
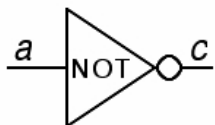
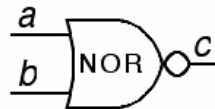
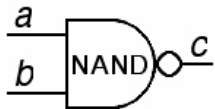
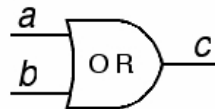
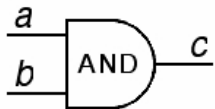




Χτίζοντας έναν Υπολογιστή



Transistor

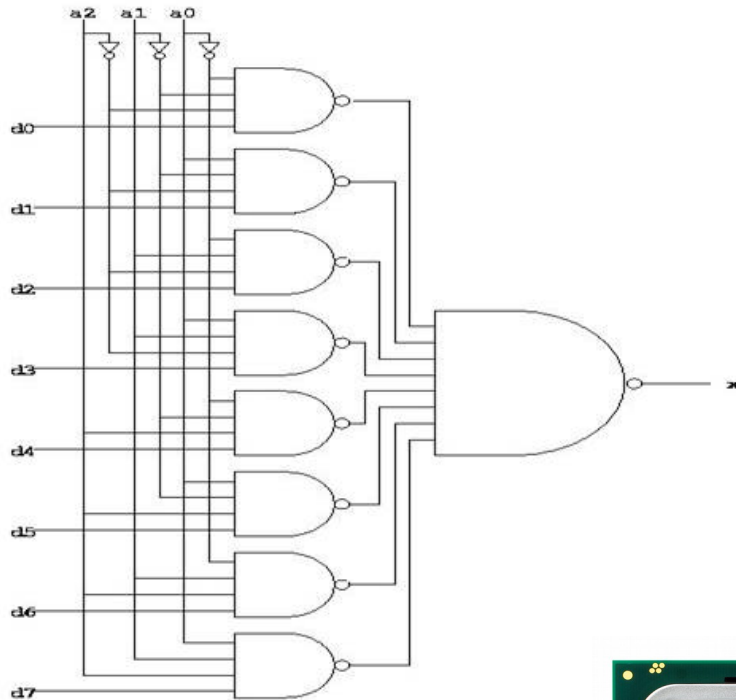


Λογικές Πύλες

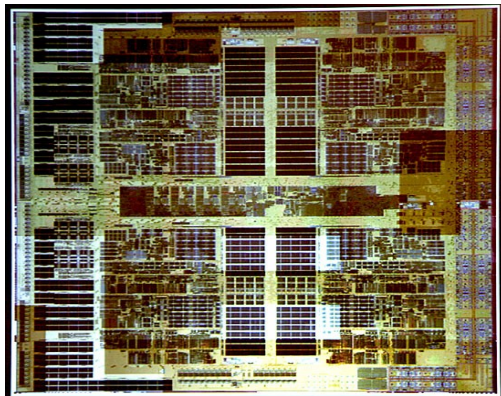
Χτίζοντας έναν Υπολογιστή



9



Λογικά Κυκλώματα

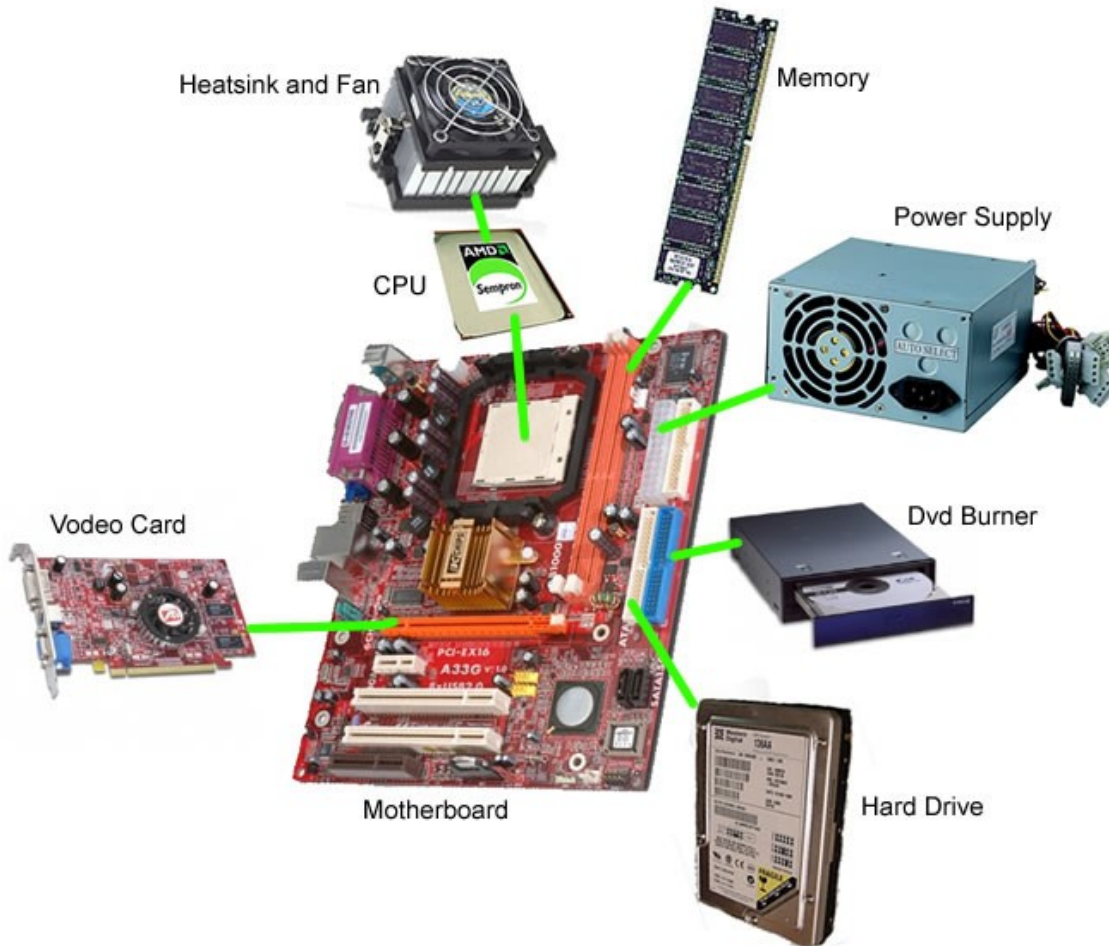


Ολοκληρωμένα
Κυκλώματα (π.χ.
Επεξεργαστές)

Χτίζοντας έναν Υπολογιστή



10



Δομικές Μονάδες -
Σύστημα

Μερικά Μυστικά των Υπολογιστών



11

- Οι υπολογιστές είναι ισχυρά εργαλεία
 - Μπορούν να κάνουν πράγματα
 - Που πριν λίγα χρόνια δεν μπορούσαμε καν να φανταστούμε
 - Μας κάνουν τη ζωή εύκολη
- Μας κάνουν τη ζωή δύσκολη
 - Οι υπολογιστές είναι **χαζοί**... Πολύ χαζοί...
 - Δεν μπορούν να κάνουν τίποτα από μόνοι τους
 - Χρειάζονται **οδηγίες** για το παραμικρό!
 - Μιλάνε μόνο μια **περίεργη γλώσσα**: 0 .. 1 .. 0 .. 0 .. 1..





12



0 vs NULL

Οδηγίες στους Υπολογιστές



13

```
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000 .ELF.....
00000010: 0300 3e00 0100 0000 d005 0000 0000 0000 ..>.....
00000020: 4000 0000 0000 0000 6022 0000 0000 0000 @....."....
00000030: 0000 0000 4000 3800 0900 4000 2200 2100 ...@.8...@.!.
00000040: 0600 0000 0400 0000 4000 0000 0000 0000 .....@.....
00000050: 4000 0000 0000 0000 4000 0000 0000 0000 @.....@.....
00000060: f801 0000 0000 0000 f801 0000 0000 0000 .....
00000070: 0800 0000 0000 0000 0300 0000 0400 0000 .....
00000080: 3802 0000 0000 0000 3802 0000 0000 0000 8.....8.....
00000090: 3802 0000 0000 0000 1c00 0000 0000 0000 8.....
000000a0: 1c00 0000 0000 0000 0100 0000 0000 0000 .....
000000b0: 0100 0000 0500 0000 0000 0000 0000 0000 .....
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000d0: 6809 0000 0000 0000 6809 0000 0000 0000 h.....h.....
000000e0: 0000 2000 0000 0000 0100 0000 0600 0000 .. .....
000000f0: a80d 0000 0000 0000 a80d 2000 0000 0000 .....
00000100: a80d 2000 0000 0000 6802 0000 0000 0000 .....h.....
00000110: 7002 0000 0000 0000 0000 0000 2000 0000 p.....
00000120: 0200 0000 0600 0000 b80d 0000 0000 0000 .....
00000130: b80d 2000 0000 0000 b80d 2000 0000 0000 .. .....
00000140: f001 0000 0000 0000 f001 0000 0000 0000 .....
00000150: 0800 0000 0000 0000 0400 0000 0400 0000 .....
00000160: 5402 0000 0000 0000 5402 0000 0000 0000 T.....T.....
00000170: 5402 0000 0000 0000 4400 0000 0000 0000 T.....D.....
```

```
219: 00 73 74          add  %dh,0x74(%rbx)
21c: 64 69 6e 00 5f 66 6c  imul $0x616c665f,%fs:0x0(%rsi),%ebp
223: 61
224: 67 73 32          addr32 jae 259 <_init-0x307>
227: 00 5f 6d          add  %bl,0x6d(%rdi)
22a: 6f              outsl %ds:(%rsi),(%dx)
22b: 64 65 00 5f 49    fs add %bl,%gs:0x49(%rdi)
230: 4f 5f            rex.WRXB pop %r15
232: 72 65            jb   299 <_init-0x2c7>
234: 61              (bad)
235: 64 5f            fs pop %rdi
237: 62 61            (bad)
239: 73 65            jae  2a0 <_init-0x2c0>
23b: 00 5f 76          add  %bl,0x76(%rdi)
23e: 74 61            je   2a1 <_init-0x2bf>
240: 62              (bad)
241: 6c              insb (%dx),%es:(%rdi)
242: 65 5f            gs pop %rdi
244: 6f              outsl %ds:(%rsi),(%dx)
245: 66 66 73 65      data16 data16 jae 2ae <_init-0x2b2>
249: 74 00            je   24b <_init-0x315>
24b: 5f              pop  %rdi
```

- Γλώσσα Μηχανής
- Η μητρική γλώσσα του υπολογιστή
- Ού μπλέξεις...

- Συμβολική γλώσσα
- Πολύ κοντά στη γλώσσα μηχανής...
- ... Αλλά κάτι μπορείς να καταλάβεις
 - Και πάλι ού μπλέξεις
 - Εκτός αν χρειαστεί ;-)

Οδηγίες στους Υπολογιστές



14

```
#include <stdio.h>
```

```
int main (int argc, char *argv[]) {  
    char input;  
    int i, count;  
  
    printf("Hello world");  
    printf("Enter character:");  
    input = getchar();  
  
    printf("Enter number of repetitions:");  
    count = getchar();  
  
    for (i = 0; i < count; i++) {  
        putchar(input);  
    }  
    printf("\n");  
  
    return(0);  
}
```

- Γλώσσες υψηλού επιπέδου
- Τώρα μάλιστα! Κάτι γίνεται!
 - Πιο κοντά στον άνθρωπο
- Ναι, αλλά ο υπολογιστής θα καταλάβει;



Περιεχόμενο Μαθήματος

- Εισαγωγή στον προγραμματισμό – Αλγόριθμος / Πρόγραμμα / Διαδικασία ανάπτυξης
- Το μοντέλο μνήμης και η έννοια της μεταβλητής.
- Βασικοί τύποι δεδομένων και τελεστές.
- Συναρτήσεις
- Δομές ελέγχου, διακλαδώσεις, επαναλήψεις
- Αποσφαλμάτωση και εργαλεία
- Πίνακες
- Αναζήτηση / Ταξινόμηση.
- Συμβολοσειρές
- Δείκτες.
- Σύνθετοι τύποι δεδομένων

- **Εργαστήριο!**



Πληροφορίες

- Βασικά συγγράμματα:
 - «**C Προγραμματισμός**», 7^η έκδοση, Deitel & Deitel, Εκδόσεις Μ. Γκιούρδας.
 - «**C Από τη Θεωρία στην Εφαρμογή**», 3^η έκδοση, Γ.Σ. Τσελίκης & Ν.Δ. Τσελίκας
 - «**Η γλώσσα C σε βάθος**», 5^η έκδοση, Νίκος Μ. Χατζηγιαννάκης, Εκδόσεις Κλειδάριθμος ΕΠΕ
- Μπορείς να μάθεις ποδήλατο διαβάζοντας οδηγίες σε βιβλίο;
 - Ο προγραμματισμός δε μαθαίνεται με διάβασμα
 - Δουλειά στον υπολογιστή!!!
 - Πειραματισμός, λάθη, εύρεση λαθών κ.ο.κ.

Απαιτήσεις - Αξιολόγηση



17



ΒΑΛΤΕ ΜΟΥ ΕΝΑ 5

Απαιτήσεις – Αξιολόγηση - 1ετείς



18

- 1ετείς: Εργαστήριο + Quiz + Homeworks + Εξέταση (στο εργαστήριο)
 - Δικαίωμα εξέτασης:
 - Εργαστήριο + Quiz : Point System – Πρέπει να μαζέψετε 26 πόντους
 - Και 12 πόντους μετά το 6ο εργαστήριο και 2ο quiz
 - Τελικός Βαθμός: $0.5 * \text{Εξέταση} + 0.25 * \text{Quiz} + 0.25 * \text{Homeworks}$
 - Για να είναι προβιβάσιμος:
 - Τελικός ≥ 5 (κατόπιν κλιμακώνεται σε [6, 10])
 - Βαθμός εξέτασης ≥ 5

Απαιτήσεις – Αξιολόγηση – 1ετείς – Point System



19

● 12 Εργαστήρια

- 0 πόντοι για απουσία / ανεπαρκή ποιότητα παραδοτέου,
- 1 πόντος για οριακά επαρκή ποιότητα παραδοτέου,
- 3 πόντοι για επαρκή / καλή ποιότητα παραδοτέου

● 4 Quiz

- Ζυγισμένα βάρη
- 20 πόντοι σύνολο (και στα 4)
- Βαθμός quiz != πλήθος πόντων που πήρατε στο εκάστοτε quiz

Απαιτήσεις – Αξιολόγηση – Μεγαλύτεροι φοιτητές



20

- Δικαίωμα εξέτασης στο εργαστήριο: Εργασία εξαμήνου ≥ 5
 - Ή επιτυχής εργασία 2017-18
- Τελικός βαθμός:
 - Βαθμός εξέτασης, κλιμακωμένος στο [6, 10]
- Επιτυχία: Τελικός βαθμός ≥ 5

Απαιτήσεις – Αξιολόγηση - Όλοι



21

- Όλοι:
 - Γραπτή εξέταση (μόνο Σεπτέμβρη): Χωρίς project, εργαστήριο κλπ.
- Τελικός βαθμός:
 - Βαθμός γραπτού κλιμακωμένος στο [5, 6] (για γραπτή εξέταση)
- Επιτυχία: Τελικός βαθμός ≥ 5



Ώρες Εργαστηρίου

- Θα διαλέξετε κάποιο από τα τμήματα
- Παρακολουθείτε τη σελίδα του μαθήματος και τον πίνακα ανακοινώσεων
- Αρχίζουμε **από την επόμενη εβδομάδα!!!**

Οι καλοί λογαριασμοί...

- Δεν επιτρέπεται να μοιράζεστε κώδικα εκτός ομάδας. Το εννοούμε...



23

- Έχουμε τρόπο να βρούμε τις “εξυπνάδες”
 - Για όποιον κάνει “εξυπνάδες” (παίρνει ή δίνει κώδικα), απλά 0... Στη 2η φορά, ραντεβού του χρόνου και αναφορά στο Τμήμα...
 - Δικαιολογίες δεν ήξερα, δεν κατάλαβα δεν πιάνουν...



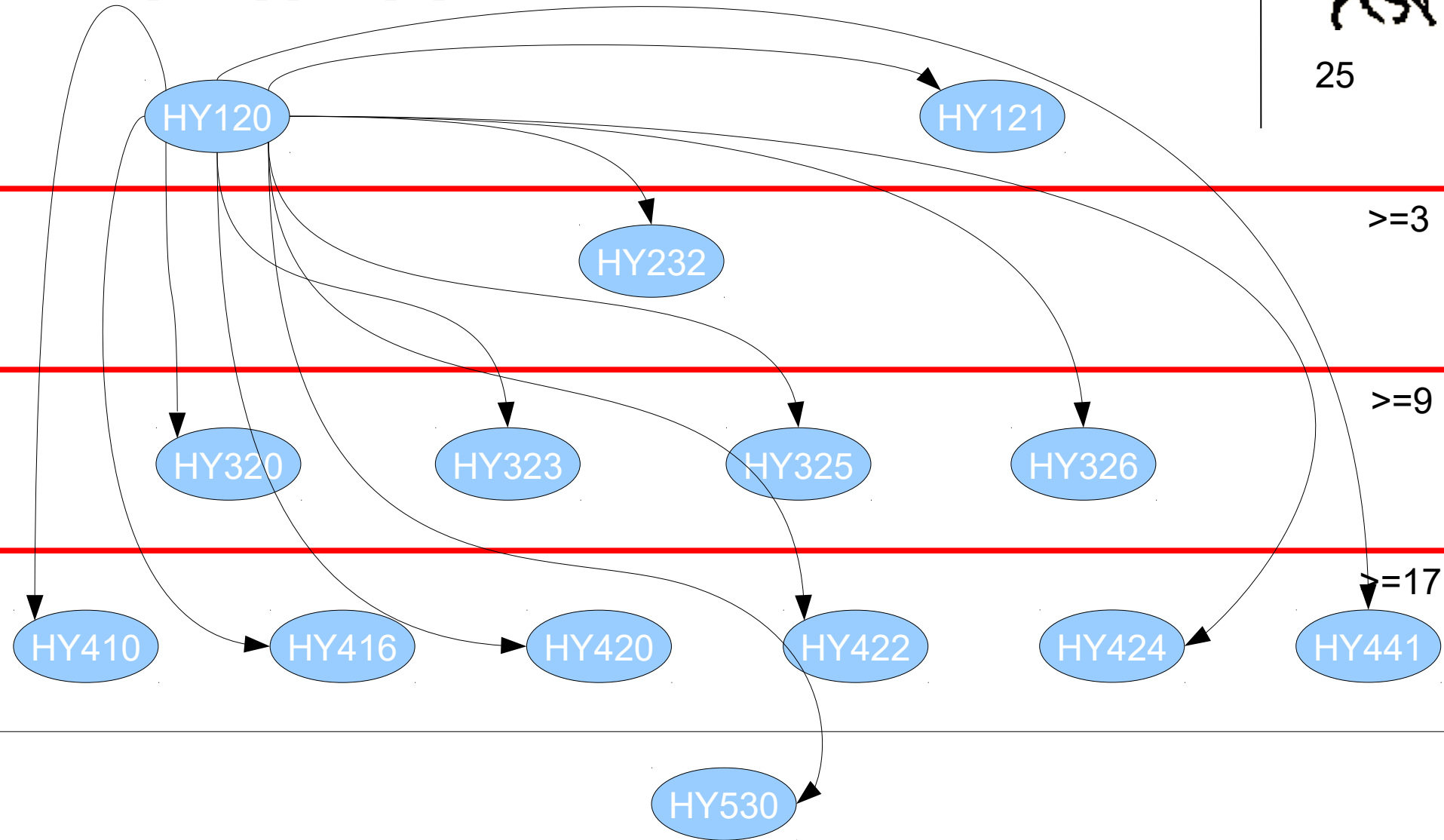
Οι καλοί λογαριασμοί ...

- Πολιτική Τμήματος για αντιγραφές:
 - 1^η φορά (σε οποιοδήποτε μάθημα): Απώλεια μαθήματος για 1 έτος
 - 2^η φορά (σε οποιοδήποτε μάθημα): Απώλεια εξαμήνου
 - 3^η φορά (σε οποιοδήποτε μάθημα): Απώλεια έτους και αναφορά στη Σύγκλητο

Πρόγραμμα Σπουδών



25



Και γιατί να προσπαθήσω;

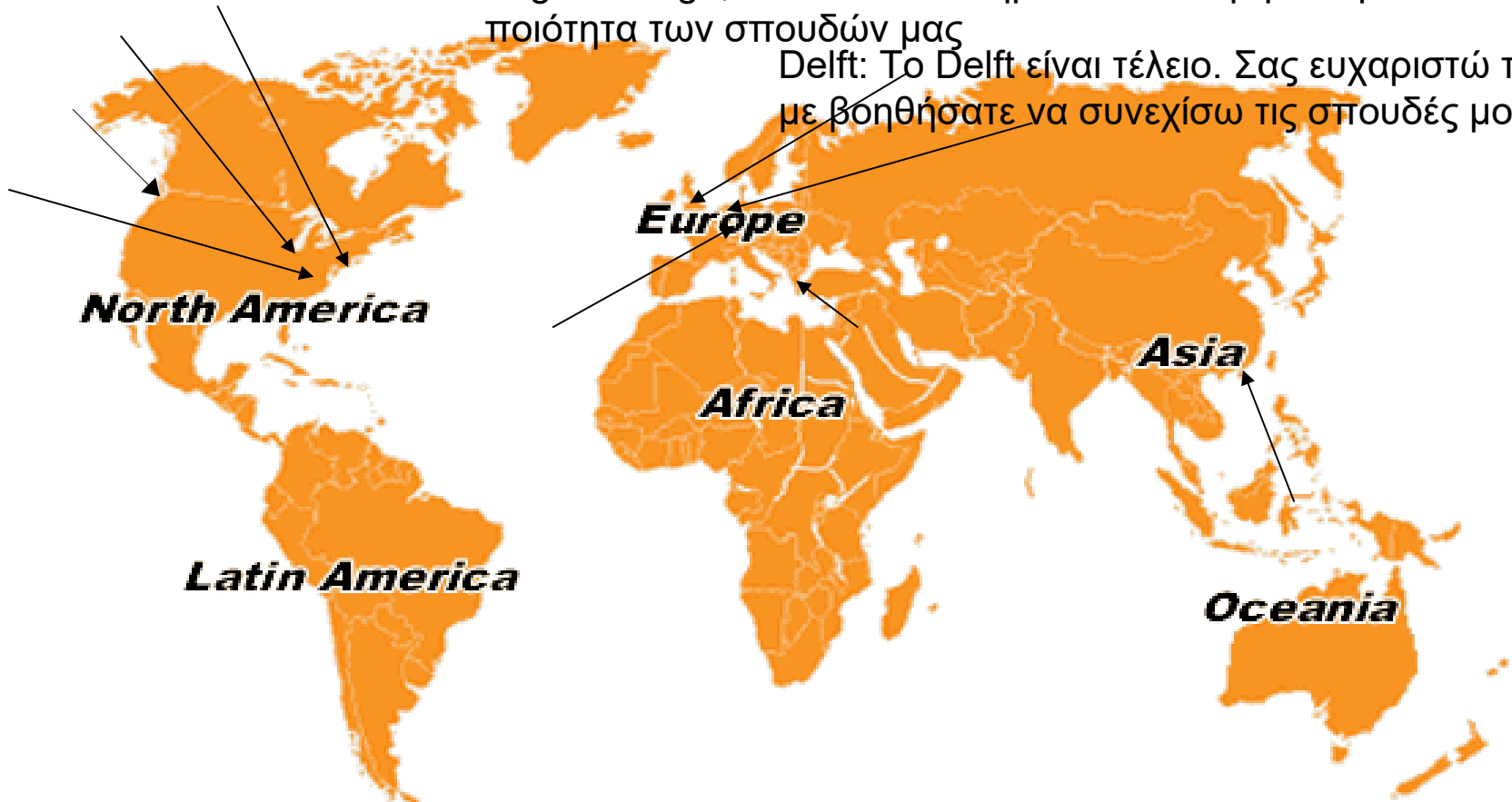


26

NY University: Εδώ στο Πανεπιστήμιο πάντως τα πράγματα πολύ καλά.
Δε θέλω να πω μεγάλες κουβέντες, αλλά νομίζω ότι άμα έχεις τελειώσει
το Τμήμα το επίπεδό σου είναι πολύ καλό. Το ίδιο και η ικανότητά σου
να αντιμετωπίζεις τις δυσκολίες.

King's College, London: Όταν ήρθα εδώ εκτίμησα την
ποιότητα των σπουδών μας

Delft: Το Delft είναι τέλειο. Σας ευχαριστώ που
με βοηθήσατε να συνεχίσω τις σπουδές μου εκεί



Επικοινωνία



27

- Ιστοσελίδα + forum μαθήματος

- <http://eclass.uth.gr/eclass/courses/MHX399/>

The screenshot shows the open eclass interface. On the left is a dark sidebar with navigation options: Ενεργά εργαλεία, Έγγραφα, Ανακοινώσεις, Εργασίες, Μηνύματα (1), Πληροφορίες, Συζητήσεις, Σύνδεσμοι, Ανενεργά εργαλεία, and Διαχείριση μαθήματος. The main content area displays the course title '2018-19 - Προγραμματισμός Ι (HY120)' and the instructor 'Χρήστος Αντωνόπουλος, Γιώργος Θάνος'. Below this is a 'Περιγραφή' section with a blue pencil icon. It features a code editor image with the text 'HY120 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι' overlaid. To the right of the image is a paragraph of text. At the bottom of the description box, it lists 'Κωδικός: ΜΗΧ399' and 'Κατηγορία: Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών » Προπτυχιακό'. At the bottom of the page, there are two buttons: 'Ημερολόγιο' and 'Ανακοινώσεις'.

Χαρτοφυλάκιο / 2018-19 - Προγραμματισμός Ι (HY120)

2018-19 - Προγραμματισμός Ι (HY120)

Χρήστος Αντωνόπουλος, Γιώργος Θάνος

Περιγραφή



Βασικός στόχος του μαθήματος είναι να παρέχει στους πρωτοετείς φοιτητές την πρώτη επαφή με τις βασικές αρχές του προγραμματισμού και γενικότερα της επίλυσης προβλημάτων με τη βοήθεια υπολογιστή. Ακολούθως, οι φοιτητές εισάγονται στη γλώσσα προγραμματισμού C, μια κλασική, ισχυρή γλώσσα διαδικαστικού (functional), δομημένου προγραμματισμού.

Το μάθημα συνοδεύεται από εργαστήριο, στο οποίο οι συμμετέχοντες εφαρμόζουν πρακτικά - σε σειρά εργασιών - όσα διδάχθηκαν στο μάθημα με στόχο τη βαθύτερη εμπέδωση των εννοιών και τεχνικών.

Κωδικός: ΜΗΧ399

Κατηγορία: Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών » Προπτυχιακό

Ημερολόγιο

Ανακοινώσεις

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Χρήστος Δ. Αντωνόπουλος
24/9/2018



- E-mail για απορίες για το μάθημα: **ce120lab@gmail.com**
- ΕΠΙΚΟΙΝΩΝΙΑ με τον διδάσκοντα:
 - Γκλαβάνη 37, Γραφείο Β3/5
 - Ώρες γραφείου: στη σελίδα του μαθήματος.
 - E-mail: *cdantonop@gmail.com, cda@inf.uth.gr*
- ΕΠΙΚΟΙΝΩΝΙΑ με τον υπεύθυνο εργαστηρίων:
 - Γκλαβάνη 37, Γραφείο Γ5/8
 - E-mail: *gthanos@uth.gr*
- ΕΠΙΚΟΙΝΩΝΙΑ με την υπεύθυνη εργαστηρίων:
 - Γκλαβάνη 37, Γραφείο Γ5/10
 - E-mail: *doufexi@gmail.com, vdoufexi@inf.uth.gr*

Μεθοδολογία Επίλυσης Προβλήματος



29

- **Ανάλυση** του προβλήματος
 - Τι έχουμε;
 - Τι ακριβώς θέλουμε να βρούμε;
 - Πώς σπάει σε μικρότερα / απλούστερα προβλήματα;
 - Χονδρικά ποια στρατηγική πρέπει να ακολουθήσουμε;
- **Αλγόριθμος**
 - Ακριβής ακολουθία πεπερασμένων βημάτων που πρέπει να ακολουθηθούν για να λυθεί το πρόβλημα
 - Απαιτήσεις
 - Ορθότητα
 - Καλή επίδοση
 - «Οικονομία» πόρων
 - Για το ίδιο πρόβλημα πιθανόν να υπάρχουν >1 κατάλληλοι αλγόριθμοι

Παράδειγμα: Γραμμική Αναζήτηση



30

Βρες τη θέση που υπάρχει (αν υπάρχει) το 50

1ο βήμα

1	13	17	20	21	34	45	46	47	50	55	59	61	63	70
---	----	----	----	----	----	----	----	----	----	----	----	----	----	----



2ο βήμα

1	13	17	20	21	34	45	46	47	50	55	59	61	63	70
---	----	----	----	----	----	----	----	----	----	----	----	----	----	----



3ο βήμα

1	13	17	20	21	34	45	46	47	50	55	59	61	63	70
---	----	----	----	----	----	----	----	----	----	----	----	----	----	----



10ο βήμα

1	13	17	20	21	34	45	46	47	50	55	59	61	63	70
---	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Γραμμική αναζήτηση - Ψευδοκώδικας



32

```
binarySearch(array, value, upper)
  position = 0
  while position ≤ upper {
    if array[position] = value
      return position
    else
      position = position+1
  }
  return not found
```


Δυαδική αναζήτηση - Ψευδοκώδικας



33

```
binarySearch(array, value, lower, upper)
  while lower ≤ upper {
    mid := floor((upper-lower)/2)+lower
    if array[mid] = value
      return mid
    if value < array[mid]
      upper := mid-1
    else
      lower := mid+1
  }
  return not found
```

Μεθοδολογία Επίλυσης Προβλήματος

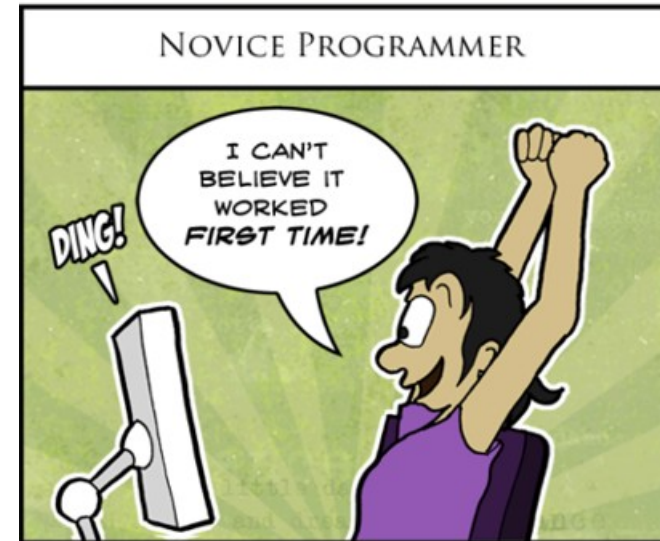


- **Κωδικοποίηση**

- «Μετάφραση» του αλγόριθμου σε γλώσσα που καταλαβαίνει ο υπολογιστής
 - C στην περίπτωση μας

- **Δοκιμή και Αποσφαλμάτωση**

- Ουδείς αλάνθαστος!
 - Το πιο «σπαστικό» κομμάτι...
 - ...Και το πιο χρονοβόρο
 - ...Και το πιο σημαντικό



Τι Είναι Πρόγραμμα και Γλώσσα Προγραμματισμού;



35

- **Πρόγραμμα**: Αλγόριθμος γραμμένος σε μια γλώσσα προγραμματισμού
 - Αυτή που «καταλαβαίνει» ο υπολογιστής
 - Με λίγη βοήθεια... Ο υπολογιστής ξέρει μόνο από 0 και 1
- **Γλώσσα προγραμματισμού**
 - Σαν τις κανονικές γλώσσες
 - Μόνο πιο σαφής και τυπική ...
 - Έχει **λεξιλόγιο / σημασιολογία**
 - Έχει **συντακτικό**

ΣΥΝΤΑΚΤΙΚΟ



- Το συντακτικό ορίζει το σύνολο των επιτρεπτών προτάσεων (ακολουθία συμβόλων).
- Επιτρεπτή πρόταση = συντακτικά ορθό πρόγραμμα
 - Μπορεί να μεταφραστεί σε γλώσσα μηχανής και να εκτελεστεί.
- Ειδικές «γλώσσες» περιγραφής των κανόνων σύνταξης γλωσσών προγραμματισμού
 - Extended Backus Naur Form (EBNF)
 - Συντακτικά διαγράμματα



Σύνταξη και Σημασία

- **Σύνταξη:** κανόνες για τον σχηματισμό προτάσεων.
- **Σημασία:** το νόημα που έχουν οι συντακτικά επιτρεπτές προτάσεις.
 - Η σύνταξη μιας γλώσσας δεν μπορεί να ορίσει ταυτόχρονα και την σημασία της.
 - Υπάρχουν όμως κανόνες!
- Υπάρχουν συντακτικά επιτρεπτές προτάσεις που δεν έχουν ορισμένη (μια μοναδική) σημασία;
 - Στις ανθρώπινες γλώσσες: Ναι
 - Μαζί μιλάμε και χώρια καταλαβαίνομαστε
 - Σε γλώσσες προγραμματισμού: Κατά κανόνα όχι



Για παράδειγμα ...

- Συντακτικά ορθές προτάσεις:

$$-2+13$$

$$15/5-2$$

- Όμως, σε τι τιμές αντιστοιχούν;

$$-2+13 : \quad (-2) + (13) \quad \text{ή} \quad -(2+13)$$

$$15/5-2 : \quad (15/5) - 2 \quad \text{ή} \quad 15 / (5-2)$$



Εκτέλεση Εντολών

- Οι εντολές ενός προγράμματος εκτελούνται από «αριστερά προς τα δεξιά» και «από πάνω προς τα κάτω»
 - Όπως διαβάζουμε ένα κείμενο.
 - Βαρετό...
- Μπορεί να γίνουν **άλματα**, είτε «προς τα εμπρός» είτε «προς τα πίσω»
 - Κάποιες εντολές μπορεί να παρακαμφθούν ή/και να εκτελεστούν παραπάνω από μια φορά.
 - Τα άλματα πραγματοποιούνται με ειδικές εντολές που ορίζει κάθε γλώσσα προγραμματισμού:
 - **Ελέγχου ροής** εκτέλεσης προγράμματος
 - **Κλήσης υποπρογραμμάτων**



Εκτέλεση Προγράμματος

επέλεξε την πρώτη εντολή

εκτέλεσε την εντολή

επέλεξε την «επόμενη»
εντολή

μνήμη

διάβασμα

γράψιμο

μεταβλητές
προγράμματος

Το σημειωματάριο
του προγράμματος...



Έλεγχος Ροής Εκτέλεσης

- Ο προγραμματιστής μπορεί να επηρεάσει την ροή εκτέλεσης του προγράμματος μέσω **εντολών ελέγχου**.
 - Εντολές ελέγχου: διαβάζουν την κατάσταση του προγράμματος και ανάλογα μεταφέρουν την εκτέλεση σε ένα διαφορετικό σημείο του κώδικα.
 - Π.χ `if <έκφραση> goto <n>`
 - Αν η <έκφραση> είναι αληθής τότε η εκτέλεση μεταφέρεται στην εντολή με αριθμό <n>.
- Μπορούμε εύκολα να φτιάξουμε προγράμματα με εντολές που εκτελούνται **υπό συνθήκη** ή/και **επανειλημμένα**.



Για Παράδειγμα...

παράμετροι

πρόγραμμα P(x, y)

```
1:s = 0
2:if (x==0) goto 6
3:s = s + y
4:x = x -1
5:if (1==1) goto 2
6:print s
```

εκτέλεση P(0, 5)

```
s = 0
if (x==0) goto 6
print s
```

εκτέλεση P(3, 5)

```
s = 0
if (x==0) goto 6
s = s + y
x = x -1
if (1==1) goto 2
if (x==0) goto 6
s = s + y
x = x -1
if (1==1) goto 2
if (x==0) goto 6
s = s + y
x = x -1
if (1==1) goto 2
if (x==0) goto 6
print s
```



Υποπρογράμματα

- Ομάδα εντολών (και δεδομένων) που είναι **συντακτικά** ή/και **εκτελεστικά ανεξάρτητη** (αυτόνομη) από τον υπόλοιπο κώδικα.
 - Ένα πρόγραμμα μπορεί να αποτελείται από (ή να χρησιμοποιεί) πολλά διαφορετικά υποπρογράμματα.
 - Όταν καλείται ένα υποπρόγραμμα, η εκτέλεση **μεταφέρεται** στο υποπρόγραμμα
 - Όταν τερματιστεί η εκτέλεση του, **επιστρέφει** μαγικά στο πρόγραμμα που πραγματοποίησε την κλήση.
 - Να τα χρησιμοποιείτε... please!!!



Εκτέλεση με Υποπρογράμματα

