



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Εισαγωγή στους Υπολογιστές

Ενότητα 2: Αλγόριθμοι, Διαγράμματα Ροής,
Εισαγωγή στην Python μεταβλητές, βασικές δομές
δεδομένων, έλεγχος ροής

Αβούρης Νικόλαος

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

Σκοποί ενότητας

- Αλγόριθμοι, Διαγράμματα Ροής, Εισαγωγή στην Python μεταβλητές, βασικές δομές δεδομένων, έλεγχος ροής



Περιεχόμενα ενότητας

- Αλγόριθμοι, Διαγράμματα Ροής, Εισαγωγή στην Python μεταβλητές, βασικές δομές δεδομένων, έλεγχος ροής



Αλγόριθμοι, Διαγράμματα Ροής,
Εισαγωγή στην Python μεταβλητές,
βασικές δομές δεδομένων, έλεγχος
ροής

Αλγόριθμοι, Δομές Δεδομένων

- Αλγόριθμος: διαδικασία επίλυσης κάποιου προβλήματος – έκφραση ενός αλγορίθμου με (α) ψευδογλώσσα, (β) διάγραμμα ροής και (γ) γλώσσα προγραμματισμού
- Κρίσιμο θέμα: η οργάνωση δεδομένων
- Προβλήματα πολυπλοκότητας : καθορισμός προβλημάτων που μπορούν να λυθούν και εκτιμώμενος χρόνος επίλυσης



Αλγόριθμος

- Ακολουθία εντολών, μέσω της οποίας είναι δυνατή η ακριβής έκφραση μιας βήμα-προς-βήμα διαδικασίας που τερματίζει την εκτέλεσή της μετά την ολοκλήρωση ενός πεπερασμένου αριθμού βημάτων και η οποία επιτρέπει την επίλυση ενός δεδομένου προβλήματος.



Αλγόριθμος

Σειριακή εκτέλεση στοιχειωδών βημάτων

- εισαγωγής δεδομένων (από τον έξω κόσμο)
- επεξεργασίας των δεδομένων
- παραγωγής δεδομένων εξόδου



Χρήση βασικών εντολών

- Για να περιγράψουμε ένα αλγόριθμο χρειαζόμαστε ένα ορισμένο αριθμό στοιχειωδών **αλγοριθμικών δομών**, **(αρχετύπων ή εντολών)** που να είναι κοντά στη λειτουργία ενός υπολογιστή.



Αλγόριθμος – ένα παράδειγμα

(στο αμφιθέατρο, συμμετέχουμε όλοι)

ΒΗΜΑ 1. Σήκω όρθιος και σκέψου τον αριθμό 1

ΒΗΜΑ 2. Βρες κάποιον άλλο που είναι όρθιος και προσθέστε τους αριθμούς σας, κράτα το άθροισμα

ΒΗΜΑ 3. Από το ζευγάρι του ΒΗΜΑΤΟΣ 2, ο ένας κάθεται, ενώ ο άλλος πηγαίνει πίσω στο ΒΗΜΑ 2.

Ερώτηση: πώς τερματίζει ο αλγόριθμος; τι αποτέλεσμα δίνει; πώς αλλιώς μπορούμε να βρούμε το ίδιο αποτέλεσμα; ποιος τρόπος συγκλίνει πιο γρήγορα;



Βασικές αλγοριθμικές δομές

1. Εκχώρηση τιμής σε μεταβλητή

`a = 5`

2. Εντολές εισόδου/εξόδου

`-a = raw_input()`

`-print b`

3. Επιλογή

`- if` συνθήκη : εντολή, `else` εντολή

4. Επανάληψη

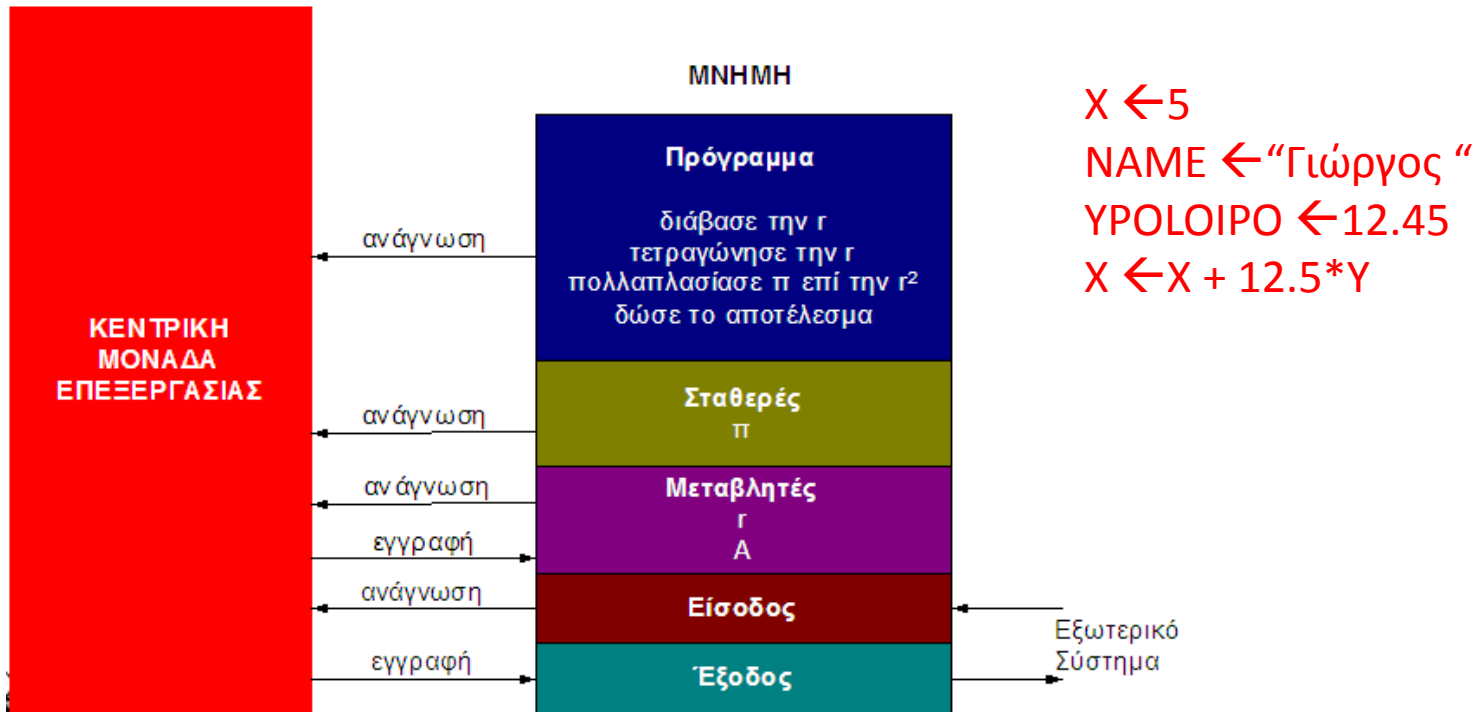
`- while` συνθήκη : εντολή

`- for i in (0,10) : εντολή`



Μεταβλητές

- Σύμβολα που αντιστοιχούν σε περιοχές της μνήμης και στις οποίες αποθηκεύονται τιμές



Πώς αντιμετωπίζει η Python τους διαφορετικούς τύπους μεταβλητών: dynamically typed, strongly typed

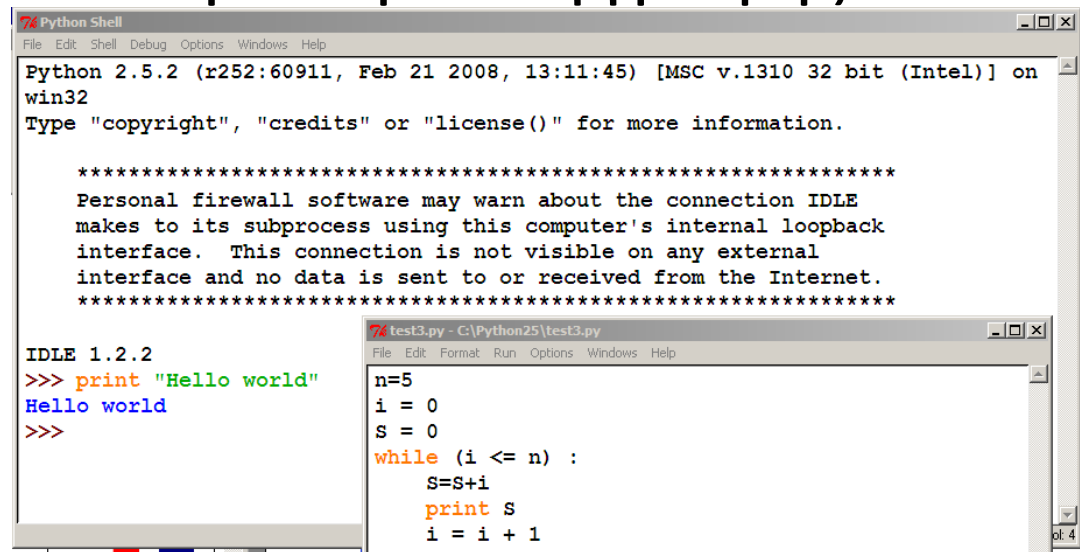
- (1) **statically typed language** :Μια γλώσσα που οι τύποι των μεταβλητών καθορίζονται κατά την ώρα της μεταγλώττισης. Είναι απαραίτητη η δήλωση των μεταβλητών. Η Java και C είναι γλώσσες με στατικούς τύπους
- (2) **dynamically typed language**: Γλώσσες στις οποίες ο τύπος των μεταβλητών αποφασίζεται κατά τη διάρκεια της εκτέλεσης. Οι VBScript και Python είναι τέτοιες γλώσσες
- (3) **strongly typed language**: Γλώσσες στις οποίες οι τύποι των μεταβλητών επιβάλλονται. Οι Java και Python έχουν ισχυρούς τύπους, ένας ακέραιος δεν μπορεί να χρησιμοποιηθεί σαν συμβολοσειρά χωρίς πριν να αλλάξει τύπο.
- (4) **weakly typed language**: Γλώσσες στις οποίες οι τύποι μπορεί να αγνοηθούν. Εκδόσεις της VBScript έχουν αδύνατους τύπους. Πχ.VBScript, συνεννόηση της συμβολοσειράς '12' με τον ακέραιο 3 έχει αποτέλεσμα '123', το οποίο μπορεί να χρησιμοποιηθεί ως 123, χωρίς μετατροπές.



Το αποθηκευμένο πρόγραμμα

Εκκίνηση της Python (IDLE (Python GUI) στο Start menu)

- Εμφανίζεται το περιβάλλον python **shell**
- CNTRL-N για ένα νέο παράθυρο συγγραφής προγράμματος
- Αποθηκεύεται ως `αρχείο.py`
- Run Module για την εκτέλεσή του



The image shows two overlapping windows from the Python IDLE environment. The top window is titled 'Python Shell' and displays the Python 2.5.2 startup screen, including version information, date, and a warning about firewall software. The bottom window is titled 'test3.py - C:\Python25\test3.py' and shows a Python script with a while loop that calculates the sum of integers from 1 to 5.

```
Python Shell
Python 2.5.2 (r252:60911, Feb 21 2008, 13:11:45) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.2.2
>>> print "Hello world"
Hello world
>>>
```

```
test3.py - C:\Python25\test3.py
File Edit Format Run Options Windows Help
n=5
i = 0
S = 0
while (i <= n) :
    S=S+i
    print S
    i = i + 1
. . .
ot.4
```



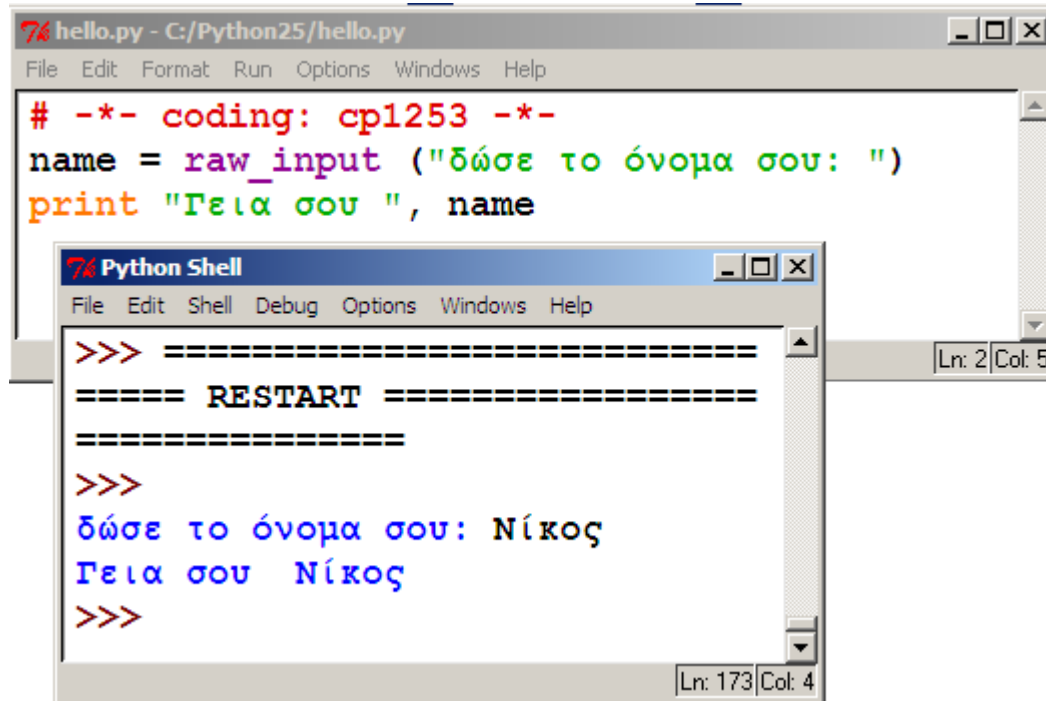
Αποθήκευση-εκτέλεση

- Ανοίγουμε νέο παράθυρο(File | New Window, ήCtrl + N)
- Έστω ότι θέλουμε να δημιουργήσουμε ένα πρόγραμμα που ρωτάει τον χρήστη το όνομα του και μετά τον χαιρετάει
- Γράφουμε τις εντολές του προγράμματος (εντολές εισόδου-εξόδου, έχουμε εντολές επεξεργασίας;) και στη συνέχεια..
- Εκτελούμε το πρόγραμμα(Run | Run Module, ήF5)
 - you'll be asked to save the program; Θα σώσουμε το πρόγραμμα σαν hello.py
 - Αν χρησιμοποιούμε ελληνικούς χαρακτήρες θα χρειαστεί να δηλώσουμε τη σχετική κωδικοποίηση
- `–# -*-coding: cp1253 -*-`

Η είσοδος/ έξοδος του προγράμματος θα εμφανιστεί στο python shell



Εντολές εισόδου/εξόδου Python



The image shows a screenshot of a Python IDE. The top window is titled 'hello.py - C:/Python25/hello.py' and contains the following code:

```
# -*- coding: cp1253 -*-
name = raw_input("δώσε το όνομα σου: ")
print "Γεια σου ", name
```

The bottom window is titled 'Python Shell' and shows the execution of the script. It displays the prompt '>>>' followed by a separator line '==== RESTART ====' and another prompt '>>>'. The user has entered 'Νίκος' in response to the prompt, and the shell has printed 'Γεια σου Νίκος'.

`name= raw_input("δώσε το όνομα σου: ")`

`print"Γεια σου ", name`

-Ποια μεταβλητή χρησιμοποιήσαμε στο πρόγραμμα αυτό;

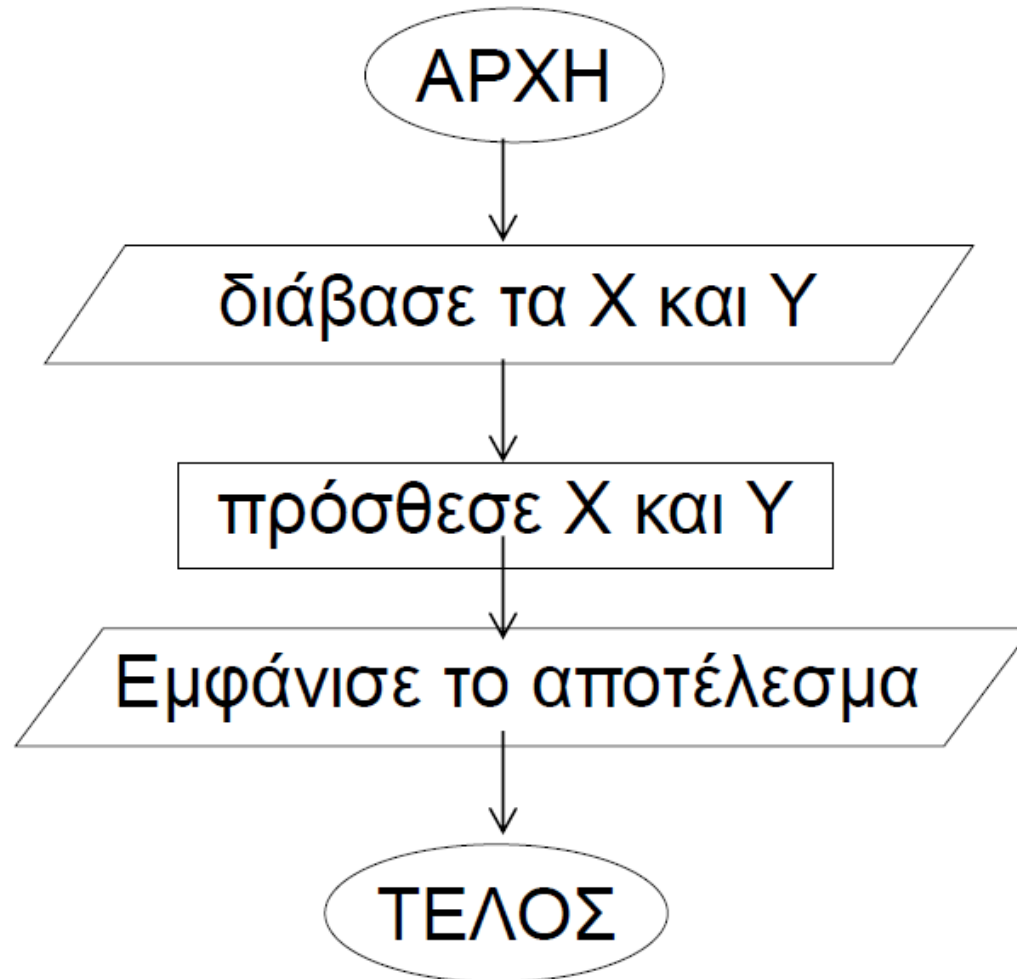


2^ο παράδειγμα προγράμματος

- Κατασκευάστε ένα πρόγραμμα που ζητάει από τον χρήστη δύο αριθμούς και τυπώνει το άθροισμα τους (αναπαράσταση του αλγορίθμου με διάγραμμα ροής και την python)



2^ο πρόγραμμα: το διάγραμμα ροής



2^ο πρόγραμμα: python

```
# -*-coding: cp1253 -*-  
print "*** Αθροιστής ***"  
x = raw_input("Δώσε τον 1ο αριθμό :")  
y = raw_input("Δώσε τον 2ο αριθμό :")  
sum = x + y  
print "Το άθροισμα είναι: " , sum
```

Μεταβλητές

x
y
sum

```
>>>  
*** Αθροιστής ***  
Δώσε τον 1ο αριθμό :8  
Δώσε τον 2ο αριθμό :2  
Το άθροισμα είναι: 82  
>>>
```

Κάτι δεν πήγε καλά στο πρόγραμμα αυτό (τι;)



Το πρόβλημα των τύπων δεδομένων

- String (**str** στη vPython) –μεταβλητές για αλφαριθμητικά (ακολουθίες χαρακτήρων)
- integer(**int** στην Python) –ακέραιοι
- floating point(**float** στην Python) –δεκαδικοί αριθμοί
- Boolean(**bool** στην Python) –λογικές μεταβλητές

Μετατροπή τύπου δεδομένων

- `x= float (y); Z= int(A); R= str(B)`



Το 2^ο πρόγραμμα -ξανά

```
# -*-coding: cp1253 -*-  
print "*** Αθροιστής ***"  
x = float (raw_input("Δώσε τον 1ο αριθμό :"))  
y = float (raw_input("Δώσε τον 2ο αριθμό :"))  
sum = x + y  
print "Το άθροισμα είναι:" , sum
```

```
>>>
```

```
*** Αθροιστής ***
```

```
Δώσε τον 1ο αριθμό :10
```

```
Δώσε τον 2ο αριθμό :20
```

```
Το άθροισμα είναι: 30.0
```



Εκχώρηση τιμής σε μεταβλητή- μεταβολή τιμής μεταβλητής

weight = 68 'σε κιλά

place = "Πάτρα" 'τόπος γέννησης

a = 3 'a έχει την τιμή 3

b = a - 5 'b έχει την τιμή -2

c = b * -3 'c έχει την τιμή 6

d = c / 2 'd έχει την τιμή 3

Στην Python δεν χρειάζεται δήλωση του τύπου μεταβλητής, αλλά ο τύπος καθορίζεται από την πρώτη φορά που θα πάρει η μεταβλητή τιμή



Μεταβλητές στην python

- Το όνομα των μεταβλητών πρέπει να αρχίζει με **γράμμα ή “_”έλεγχος τύπου με type(a)**

- Γίνονται πράξεις

Αριθμητικές (+, -, *, /, **, //, %, <<, >>, &, |, ^, ~, <, >, ==, <=, >=, !=)

Αλφαριθμητικές (+, <, >, ==, <=, >=, !=)

Λογικές (not, and, or)

- Επιτρέπονται πράξεις μεταξύ διαφορετικών τύπων μεταβλητών εκτός των μιγαδικών

```
>>> 7.0 / 2
```

```
3.5
```



Μεταβλητές

- Οι μιγαδικοί περιγράφονται είτε με τη μορφή “(real+imagj)” είτε με τη συνάρτηση “complex(real, imag)”
- Για να διαχωριστεί το πραγματικό και το φανταστικό μέρος ενός μιγαδικού z χρησιμοποιούμε `z.real` και `z.imag`

```
>>> a=1.5+0.5j
```

```
>>> a.real
```

```
1.5
```

```
>>> a.imag
```

```
0.5
```



Formating with % (%w.pf)

```
>>> print "%d" % 101.6
```

```
101
```

```
>>> x=20.0/3
```

```
>>> print "%.4f" % x
```

```
6.6667
```

```
>>> print "%3.2f" % x
```

```
6.67
```



Άσκηση

Αν ο χρήστης δώσει κατά σειρά:

(α) Κώστα/ 3/ 3

(β) 3/ 3/ Κώστα

Να προβλέψετε με ακρίβεια
την απάντηση, εξηγήστε

```
# -*- coding: cp1253 -*-  
  
##### χαιρετισμός #####  
name=raw_input ( "Πώς σε λένε; ")  
print "Γεια σου ", name  
print ""  
##### η ρίζα του αριθμού #####  
number = raw_input ( "Δώσε ένα αριθμό :")  
riza = ( float ( number) ) ** 0.5  
  
print "η ρίζα του αριθμού ", number, " είναι: ", riza  
  
print "η ρίζα του αριθμού ", number, " είναι: %5.3f " %riza  
  
##### εισαγωγή στοιχειων με input #####  
  
number = input ( "\nδώσε ένα ακόμη αριθμό : ")  
  
print "Το τετράγωνο του αριθμού %.2f είναι : %.2f. " % ( number, number**2)
```



Απάντηση

=====

>>>

Πώς σε λένε; Κώστα
Γεια σου Κώστα

Δώσε ένα αριθμό :3

η ρίζα του αριθμού 3 είναι: 1.73205080757
η ρίζα του αριθμού 3 είναι: 1.732

δώσε ένα ακόμη αριθμό : 3

Το τετράγωνο του αριθμού 3.00 είναι : 9.00.

>>> |

=====

>>>

Πώς σε λένε; 3
Γεια σου 3

Δώσε ένα αριθμό :3

η ρίζα του αριθμού 3 είναι: 1.73205080757
η ρίζα του αριθμού 3 είναι: 1.732

δώσε ένα ακόμη αριθμό : Κώστα

Traceback (most recent call last) :

File "C:/Users/user/Documents/2014/acad21
3/ex_input.py", line 17, in <module>
 number = input ("\nδώσε ένα ακόμη αριθμ
File "<string>", line 1
 Κώστα
 *



Διαγράμματα ροής –εντολές επιλογής

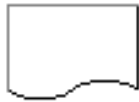
- Περιγράψτε τον αλγόριθμο με χρήση διαγράμματος ροής και στη συνέχεια δημιουργήσετε ένα πρόγραμμα Python του παιχνιδιού «μάντεψε ένα αριθμό από το 1 στο 100».
- Ο παίκτης προσπαθεί να βρει ένα αριθμό που ο υπολογιστής έχει βάλει στο μυαλό του με ερωτήσεις που παίρνουν απάντηση: σωστός, όχι είναι μεγαλύτερος, όχι είναι μικρότερος,
- Το παιχνίδι τελειώνει όταν το βρει, κερδίζει πόντους 10- n όπου n οι προσπάθειες και 0 αν οι προσπάθειες είναι μεγαλύτερες ή ίσες των 10. Εξετάστε το ενδεχόμενο το παιχνίδι να επαναλαμβάνεται αν το ζητήσει ο παίκτης.



Διάγραμμα Ροής : σύμβολα



The **Process Symbol** represents any process, function, or action and is the most frequently used symbol in flowcharting.



The **Document Symbol** is used to represent any type of hard copy input or output (i.e. reports).



Offpage Connector Symbols are used to indicate the flow chart continues on another page. Often, the page number is placed in the shape for easy reference.



The **Input/Output Symbol** represents data that is available for input or resulting from processing (i.e. customer database records).



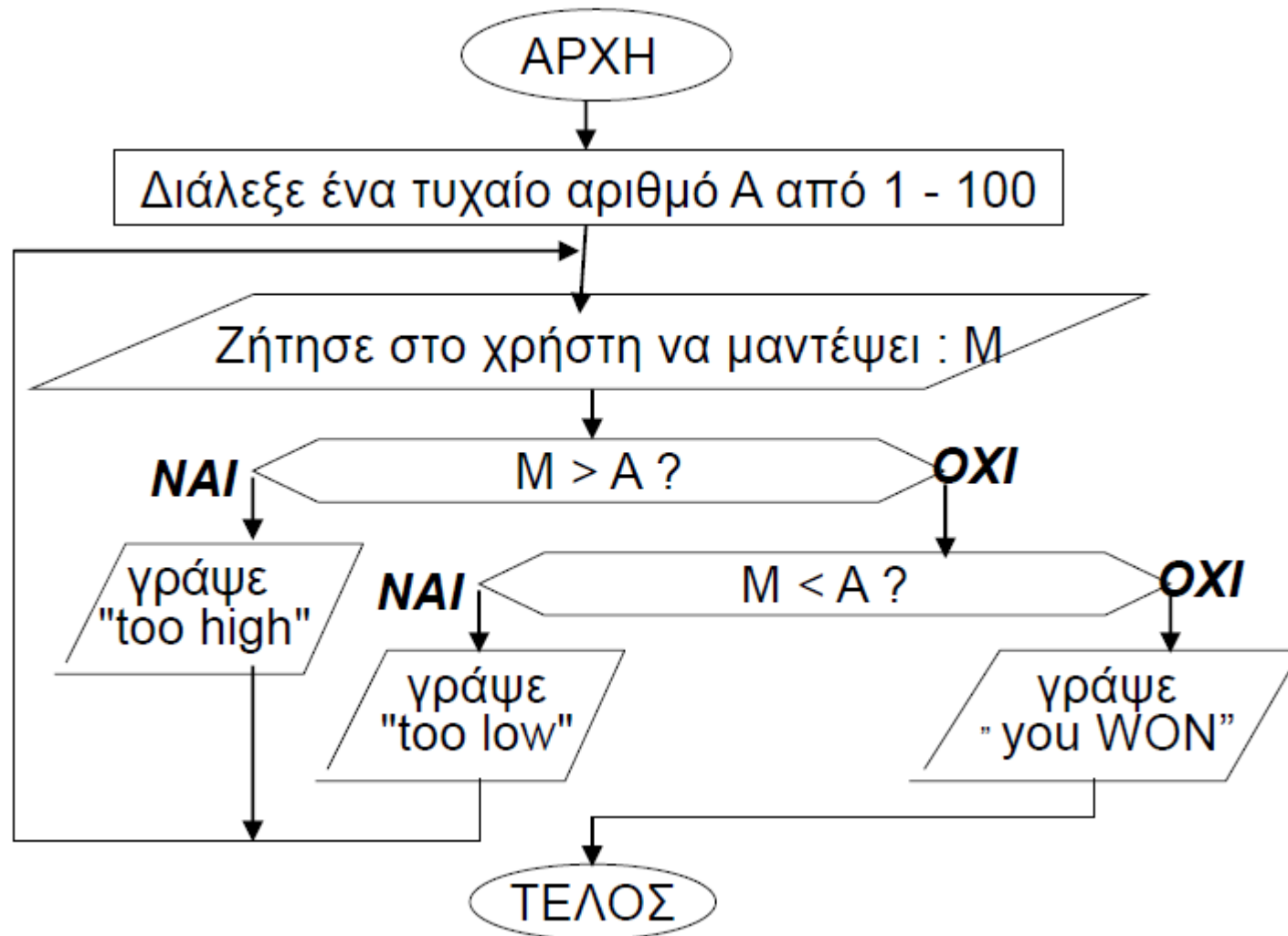
The **Decision Symbol** is a junction where a decision must be made. A single entry may have any number of alternative solutions, but only one can be chosen.



The **Connector Symbol** represents the exit to, or entry from, another part of the same flow chart. It is usually used to break a flow line that will be continued elsewhere. It's a good idea to reference page numbers for easy location of connectors.



Διάγραμμα ροής: παράδειγμα



Εντολή επιλογής (if)

- Δυνατότητα επιλογής τιμής

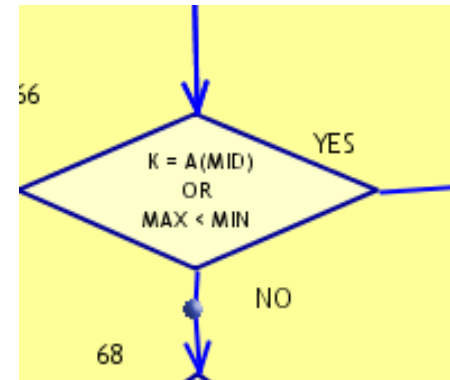
```
if(ΣΥΝΘΗΚΗ) thenΕΠΙΛΟΓΗ-1  
elseΕΠΙΛΟΓΗ-2
```

- Παράδειγμα: $f(n) = n$ όταν $n > 0$, διαφορετικά $f(n) = 0$

```
if( $n \leq 0$ )
```

```
then  $x \leftarrow 0$ 
```

```
else  $x \leftarrow n$ 
```



If στην PYTHON

if συνθήκη:
 εντολή1

elif συνθήκη:
 εντολή2

else:
 εντολή3



Άσκηση

Αν `epilogh==2` ποιες γραμμές του προγράμματος θα εκτελεστούν;

```
1 epilogh= int(raw_input("κάνε την επιλογή σου!"))
```

```
2if epilogh== 1:
```

```
3 message = "κέρδισες ένα αυτοκίνητο!"
```

```
4elif epilogh== 2:
```

```
5 message = «κέρδισες πλυντήριο πιάτων!"
```

```
6else:
```

```
7 message = «κέρδισες ένα σαμπουάν!"
```

```
8 print message
```



Διάφορες δομές επιλογής

1. **if** συνθήκη:
2. εντολή1
3. εντολή2

A

1. **if** συνθήκη:
2. εντολή1
3. **else:**
4. εντολή2
5. εντολή3

B

D

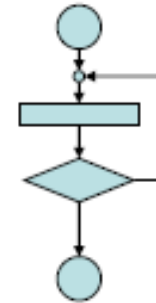
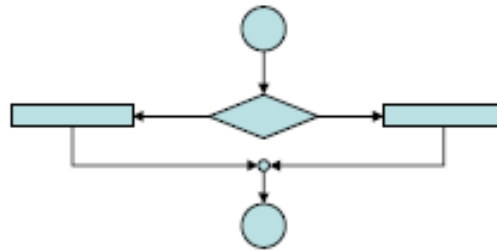
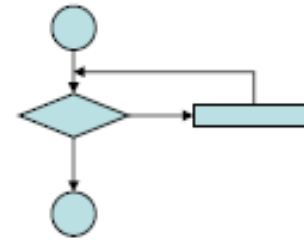
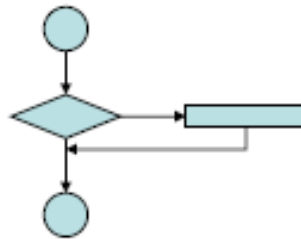
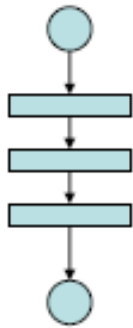
1. **if** συνθήκη1:
2. εντολή1
3. **elif** συνθήκη2 :
4. εντολή2
5. εντολή3

C

1. **if** συνθήκη1:
2. εντολή1
3. **elif** συνθήκη2:
4. εντολή2
5. **else:**
6. εντολή3
7. εντολή4



Σύνοψη δομών επιλογής



Άσκηση

- Να εκφράσετε σε Python τον αλγόριθμο «μάντεψε ένα αριθμό»
- Πρόσθεσε την περίπτωση υπολογισμού των πόντων και εκτύπωση του σκορ



Συνθήκη: τελεστές σύγκρισης

$== \rightarrow !=$

$!= \rightarrow ==$

$> \rightarrow <=$

$>= \rightarrow <$

$< \rightarrow >=$

$<= \rightarrow >$



Φροντιστηριακή Άσκηση Φ2: Μετατροπή Θερμοκρασιών

Γράψτε ένα πρόγραμμα που θα μετατρέπει τιμές θερμοκρασιών από βαθμούς Κελσίου (C) σε Φαρενάιτ (F) και αντίστροφα. Αν δεν γνωρίζετε τους τύπους μετατροπής ψάξτε τους στο Διαδίκτυο. Το πρόγραμμα:

- Πρώτα θα διαβάζει έναν ακέραιο αριθμό που αντιστοιχεί στο πλήθος των τιμών που θα διαβάσει στην είσοδο. Ο αριθμός θα είναι στο διάστημα [1, 100].
- Μετά θα διαβάζει τις τιμές προς μετατροπή ως αλφαριθμητικά που το καθένα τελειώνει με τον χαρακτήρα C ή F και δηλώνει σε ποιά κλίμακα είναι εκφρασμένη η τιμή (πχ. 12C σημαίνει 12 βαθμοί Κελσίου). Το αριθμητικό μέρος του αλφαριθμητικού θα είναι πάντα ακέραιος στο διάστημα [-100, 200].
- Το πρόγραμμα κάθε τιμή που διαβάζει θα πρέπει να την μετατρέπει στην άλλη κλίμακα και να τυπώνει το αποτέλεσμα ως πραγματικό αριθμό με ακρίβεια 2 δεκαδικών ψηφίων.

| Παράδειγμα εισόδου: | Παράδειγμα εξόδου: |
|---------------------|--------------------|
| 3 | 53.60 |
| 12C | 7.22 |
| 45F | -148.00 |
| -100C | |

**Υποβάλλεται
στην Αστράια
(10-17/10)**



Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

- Έκδοση **1.0** διαθέσιμη [εδώ](#).



Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Αβούρης Νικόλαος, Παλιουράς Βασίλειος, Κουκιάς Μιχαήλ, Σγάρμπας Κυριάκος. «Εισαγωγή στους Υπολογιστές Ι, Αλγόριθμοι και Δομές Δεδομένων, Εισαγωγή». Έκδοση: 1.0. Πάτρα 2014.
Διαθέσιμο από τη δικτυακή διεύθυνση:

https://eclass.upatras.gr/modules/course_metadata/opencourses.php?fc=15



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Διαφάνειες 13, 15, 25: χρήση του Python Editor IDLE

