



1ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ

ΧΡΗΣΤΟΣ ΕΥΣΤΑΘΙΟΥ

Καθηγητής Πληροφορικής
1^ο ΕΠΑΛ Τρικάλων

Αρχές Προγραμματισμού Υπολογιστών Σε γλώσσα Python

Για την Β' τάξη των ΕΠΑΛ



Τρίκαλα, 2021

ΣΥΓΓΡΑΦΕΑΣ:

Χρήστος Ευσταθίου, καθηγητής πληροφορικής –ΠΕ86

ΕΠΙΜΕΛΕΙΑ:

Νικόλαος Ντόγκας, Διευθυντής 1^{ου} ΕΠΑΛ Τρικάλων

1^η ΕΚΔΟΣΗ:

1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ, Σεπτέμβριος 2021

ISBN: 978-618-85383-2-0

Copyright[©]: 1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ, ΧΡΗΣΤΟΣ ΕΥΣΤΑΘΙΟΥ

Αναφορά Δημιουργού-Μη Εμπορική Χρήση 4.0 Διεθνές



Μπορείτε να:

Μοιραστείτε — αντιγράψετε και αναδιανέμετε το υλικό με κάθε μέσο και τρόπο

Προσαρμόστε — αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο υλικό

Υπό τους ακόλουθους όρους:

Αναφορά Δημιουργού — Θα πρέπει να καταχωρίσετε αναφορά στο δημιουργό, με σύνδεσμο της άδειας, και με αναφορά αν έχουν γίνει αλλαγές. Μπορείτε να το κάνετε αυτό με οποιονδήποτε εύλογο τρόπο, αλλά όχι με τρόπο που να υπονοεί ότι ο δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.

Μη Εμπορική Χρήση — Δε μπορείτε να χρησιμοποιήσετε το υλικό για εμπορικούς σκοπούς.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	5
Η ΓΛΩΣΣΑ PYTHON	7
1. ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ PYTHON	9
1.1 Στοιχεία θεωρίας	9
1.2 Ερωτήσεις κατανόησης	15
1.3 Ασκήσεις	21
2. ΑΛΓΟΡΙΘΜΙΚΕΣ ΔΟΜΕΣ	23
2.1 Στοιχεία θεωρίας	23
2.2 Ερωτήσεις κατανόησης	26
2.3 Ασκήσεις στη δομή επιλογής	33
2.4 Ασκήσεις στη δομή for	36
2.5 Ασκήσεις στη δομή while	39
3. ΣΥΝΑΡΤΗΣΕΙΣ	43
3.1 Στοιχεία θεωρίας	43
3.2 Ερωτήσεις κατανόησης	45
3.3 Ασκήσεις	51
4. ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ	55
4.1 Στοιχεία θεωρίας	55
4.2 Ερωτήσεις κατανόησης στις συμβολοσειρές	59
4.3 Ασκήσεις στις συμβολοσειρές	63
4.4 Ερωτήσεις κατανόησης στις λίστες	65
4.5 Ασκήσεις στις λίστες	74
5. ΚΛΑΣΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	81
5.1 Αλγόριθμος	81
5.2 Σειριακή Αναζήτηση	82
5.3 Ταξινόμηση	84
5.4 Ασκήσεις	86
6. ΕΠΑΝΑΛΗΨΗ	87
6.1 Ασκήσεις	87
ΠΑΡΑΡΤΗΜΑ: ΕΝΔΕΙΚΤΙΚΕΣ ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ	91
ΒΙΒΛΙΟΓΡΑΦΙΑ	147

Πρόλογος

Στο 1^ο ΕΠΑΛ Τρικάλων συνεχίζουμε την εκδοτική προσπάθεια, αξιοποιώντας την αξιόλογη εκπαιδευτική εμπειρία των καθηγητών του, εμπλουτίζοντας την διδασκαλία διαφόρων γνωστικών αντικειμένων και πέρα από το σχολικό βιβλίο, δίνοντας τη δυνατότητα στους μαθητές να έχουν πρόσβαση σε εναλλακτικές πηγές γνώσης και μάθησης.

Το βιβλίο που κρατάτε στα χέρια σας ή ξεφυλλίζετε ηλεκτρονικά στον υπολογιστή σας, αποτελεί τη τρίτη εκδοτική προσπάθεια στο 1^ο ΕΠΑΛ Τρικάλων και βασίστηκε σε ασκήσεις και εργασίες που έγιναν στο εργαστήριο από τον καθηγητή πληροφορικής και διευθυντή του 1^{ου} Εργαστηριακού Κέντρου Τρικάλων κ. Χρήστου Ευσταθίου.

Λειτουργεί συμπληρωματικά στο σχολικό βιβλίο για το μάθημα «Αρχές Προγραμματισμού Υπολογιστών σε γλώσσα Python» του τομέα Πληροφορικής της Β' τάξης των ΕΠΑΛ και γράφτηκε με σκοπό να βοηθήσει τους καθηγητές και τους μαθητές παραθέτοντας πλήθος ασκήσεων και ερωτήσεων.

Είναι οργανωμένο σύμφωνα με τη δομή του σχολικού βιβλίου από το οποίο αντλεί στοιχεία θεωρίας και ασκήσεις. Σε κάθε κεφάλαιο παρατίθεται σύντομα η θεωρία και ακολουθούν ερωτήσεις για την εμπέδωσή της και ασκήσεις. Στο τέλος του βιβλίου υπάρχει παράρτημα όπου παρατίθενται και οι ενδεικτικές λύσεις στις ασκήσεις του κάθε κεφαλαίου.

Αυτό το βιβλίο, όπως και όλες οι εκδόσεις του 1^{ου} ΕΠΑΛ Τρικάλων, κυκλοφορεί παράλληλα και σε ηλεκτρονική μορφή, έτσι ώστε να είναι προσιτό σε όλη την εκπαιδευτική κοινότητα και σε όποιον θέλει να ασχοληθεί με τον Προγραμματισμό Υπολογιστών σε γλώσσα Python.

Η εκδοτική προσπάθεια του 1^{ου} ΕΠΑΛ Τρικάλων θα συνεχιστεί και σε άλλους γνωστικούς τομείς, στηρίζοντας τους εκπαιδευτικούς του σχολείου μας στην επαγγελματική τους εξέλιξη και δίνοντας στους μαθητές μας τα εφόδια εκείνα που θα τους βοηθήσουν στην εξεύρεση εργασίας μετά την αποφοίτησή τους από το σχολείο μας.

ΝΙΚΟΛΑΟΣ Κ. ΝΤΟΓΚΑΣ

Διευθυντής 1^{ου} ΕΠΑΛ ΤΡΙΚΑΛΩΝ

Η γλώσσα Python

Η Python δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 σαν γλώσσα σεναρίων (script language). Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου του 2000.

Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η έκδοση 3.0 που είναι η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις ώστε να διορθωθούν κάποια λάθη που υπήρχαν σε προγενέστερες εκδόσεις. Η πιο πρόσφατη έκδοση είναι η 3.9.2 και δημοσιεύτηκε στις 26 Φεβρουαρίου 2021.

Η Python ενδείκνυται ως γλώσσα εισαγωγική στον προγραμματισμό και στην επιστήμη των υπολογιστών γενικότερα, με δεδομένη την απλότητα στη σύνταξή της. Διεθνώς πολλά πανεπιστήμια αλλά και η Δευτεροβάθμια Εκπαίδευση σε πολλές χώρες την υιοθετούν σαν γλώσσα εισαγωγής στον προγραμματισμό. Στη Δευτεροβάθμια εκπαίδευση της χώρας μας το πρώτο βήμα έγινε με τα Επαγγελματικά Λύκεια στα οποία αντικατέστησε την Pascal.

Η Python χρησιμοποιείται επίσης ως εργαλείο για εισαγωγή σε βασικές έννοιες των υπολογιστών και έχει με επιτυχία υποστηρίξει οικοδομιστικές προσεγγίσεις στη μάθηση, σε επί μέρους αντικείμενα.

Άλλες προτεινόμενες προσεγγίσεις είναι η χρήση της Python για διαχείριση δεδομένων, για μελέτες σε ανθρωπιστικές σπουδές, για γεωγραφικά συστήματα πληροφοριών, κλπ.

Επιγραμματικά, τα κυριότερα πλεονεκτήματά της είναι:

- Απλή στη σύνταξη
- Εύκολη στην εκμάθηση
- Ελεύθερη και Ανοικτού Κώδικα
- Γλώσσα υψηλού επιπέδου
- Φορητή
- Επεκτάσιμη
- Ενσωματώσιμη
- Διαθέτει εκτεταμένες βιβλιοθήκες
- Έχει πολλές δομές δεδομένων
- Συνδυάζει αντικειμενοστραφή, διαδικασιακό και συναρτησιακό τρόπο προγραμματισμού
- Απαιτεί υποχρεωτικά στοίχιση
- Είναι δωρεάν
- Διαθέτει μεγάλη κοινότητα υποστήριξης
- Προσφέρεται άφθονο εκπαιδευτικό υλικό

Σημείωση: Χρησιμοποιείται η έκδοση 2.7 της γλώσσας Python επειδή αυτή αναφέρεται στο σχολικό βιβλίο.

Τα κεφάλαια Διαχείριση αρχείων, Εφαρμογές σε γλώσσα προγραμματισμού με χρήση API δεν περιλαμβάνονται αφού διδάσκονται αναλυτικά στην Γ' τάξη του τομέα. Επίσης τα δυο πρώτα κεφάλαια παραλείπονται σαν καθαρώς θεωρητικά.

Επικουρικά προτείνεται η χρήση εφαρμογής εικονικής εκτέλεσης του κώδικα για την παρακολούθηση της ροής του προγράμματος και για τον έλεγχο των τιμών των μεταβλητών όπως το Python tutor (<http://pythontutor.com>)

1. Βασικά στοιχεία της Python

1.1 Στοιχεία θεωρίας

1.1.1 Τύποι δεδομένων

Αριθμοί

- **Ακέραιοι** (integers)
- **Κινητής υποδιαστολής** (floating point)
Παραδείγματα: 3.14 28.2E-5
- **Μιγαδικοί** (complex)
Παράδειγμα: (-2+3j)

Λογικοί (Boolean)

Παραδείγματα: True False

Συμβολοσειρές (Strings)

Παραδείγματα: “Καλημέρα σε όλους”
‘Σήμερα είναι μία ηλιόλουστη μέρα!’
‘a’ ‘4.5’

1.1.2 Μεταβλητές

Η μεταβλητή είναι ένα όνομα που αναφέρεται σε μία τιμή. Η τιμή αυτή μπορεί να αλλάξει όσες φορές θέλουμε ή όσες απαιτείται κατά την εκτέλεση του προγράμματός μας, εξ ου και ο όρος μεταβλητή. Οι μεταβλητές δεν είναι παρά τμήματα της μνήμης του υπολογιστή μας όπου μπορούμε να αποθηκεύσουμε πληροφορίες.

Τύποι μεταβλητών

- Ακέραιοι (integers)
- Λογικοί (Boolean)
- Συμβολοσειρές (Strings)

Η εντολή εκχώρησης δημιουργεί νέες μεταβλητές και τους δίνει τιμές.

Παράδειγμα:

```
>>> day= 'Δευτέρα'
```

Μία εντολή εκχώρησης αποτελείται από τρία μέρη:

Το αριστερό που υποχρεωτικά πρέπει να είναι μία μεταβλητή, ο τελεστής εκχώρησης `=` και το δεξί μέρος που μπορεί να είναι μία τιμή, μία άλλη μεταβλητή ή μία έκφραση που αποτιμάται σε μία τιμή.

Στην Python δεν δηλώνουμε ρητά τους τύπους δεδομένων που χρησιμοποιούμε. Η ιδιότητα αυτή ονομάζεται δυναμική απόδοση τύπων. Ο έλεγχος γίνεται κατά την εκτέλεση του προγράμματος.

1.1.3 Αριθμητικοί τελεστές

Τελεστής	Λειτουργία	Παράδειγμα	Αποτέλεσμα
-	Άρνηση	-5	-5
+	Πρόσθεση	11+3.1	14.1
-	Αφαίρεση	5-19	-14
*	Πολλαπλασιασμός	8.5*4	34.0
/	Ακέραια διαίρεση	11/2	5
%	Υπόλοιπο	8%3	2
**	Δύναμη	2**5	32

Προτεραιότητα αριθμητικών τελεστών

**	Δύναμη
-	Άρνηση
* / %	Πολλαπλασιασμός, Ακέραια Διαίρεση, Υπόλοιπο
+ -	Πρόσθεση, Αφαίρεση

1.1.4 Τελεστές μεταβολής τιμής

Τελεστής	Παράδειγμα		Αποτέλεσμα
+=	x=7	x+=2	9
-=	x=8	x-=2	6
=	x=4	x=3	12
/=	x=9	x/=5	1
%=	x=5	x%=3	2
=	x=2	x=4	16

1.1.5 Τελεστές συσχέτισης και τελεστές ισότητας

Τελεστής	Λειτουργία
>	Μεγαλύτερο από
<	Μικρότερο από
>=	Μεγαλύτερο ή ίσο με
<=	Μικρότερο ή ίσο με
==	Ίσο με
!=	Διαφορετικό από

1.1.6 Λογικοί τελεστές

or **Λογικό Η**

Το x or y επιστρέφει True αν το x ή το y είναι True

and **Λογικό ΚΑΙ**

Το x and y επιστρέφει True μόνο αν το x και το y είναι True

not **Άρνηση**

Αντιστρέφει την τιμή του τελεστέου

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΛΟΓΙΚΩΝ ΤΕΛΕΣΤΩΝ

P	Q	P and Q	P or Q	Not P
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

1.1.7 Προτεραιότητα τελεστών

1. Αριθμητικοί τελεστές
2. Τελεστές συσχέτισης
3. Λογικοί τελεστές

1.1.8 Σχόλια

Χρησιμεύουν για την τεκμηρίωση του κώδικά μας. Διακρίνονται σε:

- Σχόλια μιας γραμμής που εισάγονται με τον χαρακτήρα **#**
Παράδειγμα:
(f - 32) * 5 / 9 # μετατρέπει βαθμούς Fahrenheit σε Celsius
- Σχόλια περισσότερων γραμμών που περικλείονται σε τρία απλά εισαγωγικά.

Παράδειγμα:
""" Αυτό είναι
Το πρώτο μας
Πρόγραμμα """

1.1.9 Βασικές εντολές

Για εμφάνιση στην οθόνη χρησιμοποιούμε την εντολή **print** με σύνταξη:

```
print "μήνυμα" ή  
print τιμή μεταβλητής
```

Για είσοδο αριθμητικών δεδομένων από το πληκτρολόγιο χρησιμοποιούμε την εντολή **input** με σύνταξη:

```
μεταβλητή=input("μήνυμα")
```

Για είσοδο δεδομένων συμβολοσειράς από το πληκτρολόγιο χρησιμοποιούμε την εντολή **raw_input** με σύνταξη:

```
μεταβλητή=raw_input("μήνυμα")
```

1.1.10 Βασικές συναρτήσεις

Η Python διαθέτει ένα σύνολο ενσωματωμένων συναρτήσεων που μπορούμε να καλούμε άμεσα, όπως οι παρακάτω:

float()

Μετατρέπει ακεραίους και συμβολοσειρές σε δεκαδικούς αριθμούς.

int()

Δέχεται οποιαδήποτε τιμή και τη μετατρέπει σε ακέραιο, κόβοντας τα δεκαδικά ψηφία, αν υπάρχουν.

abs()

Επιστρέφει την απόλυτη τιμή ενός αριθμού.

pow(a,b)

Επιστρέφει τη δύναμη του a υψωμένη στο b.

divmod(x,y)

Επιστρέφει το πηλίκο και το υπόλοιπο της διαίρεσης x/y.

1.2 Ερωτήσεις κατανόησης

1. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=1
>>> b=15
>>> c=a+b*3
>>> print c
```

Επίλεξε ένα:

- 47
- 48
- 46

2. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=4
>>> b=15
>>> c=(a/2-b)*2
>>> print c
```

Επίλεξε ένα:

- 26
- -28
- -26

3. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=7
>>> b=4
>>> c=a/b
>>> print c
```

Επίλεξε ένα:

- 1.7
- 1
- 1.75

4. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=12
>>> b=3
>>> c=a%b
>>> print c
```

5. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=11.0
>>> b=5
>>> c=a/b
>>> print c
```

Επίλεξε ένα:

- 2
- 2.2
- 2.0

6. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=19
>>> b=2
>>> a+=5
>>> b=a+3
>>> c=a+b
>>> print a,b,c
```

7. Διάλεξε τι θα εμφανίσει στη οθόνη η παρακάτω εντολή:

```
>>> print 14-3**2+7/2
```


Επίλεξε ένα:

- 8
- 8.5
- 8.0

8. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=12
>>> b=2
>>> c=a%b
>>> print c
```

Επίλεξε ένα:

- 0.0
- 0
- 2

9. Γράψε τι θα εμφανίσουν στην οθόνη οι παρακάτω εντολές:

```
>>> a=12
>>> b=5
>>> c=float(a%b)
>>> print c
```

10. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a=12
>>> b=2
>>> c=float(a)%b
>>> print c
```

11. Διάλεξε τι θα εμφανίσουν στην οθόνη οι παρακάτω εντολές:

```
>>> x="Maria"
>>> Print x
```

Επίλεξε ένα:

- *Maria*
- *x*
- *error*

12. Διάλεξε τι θα εμφανίσει στη οθόνη η παρακάτω εντολή:

```
>>> x="Maria"  
>>> print X
```

Επίλεξε ένα:

- *Maria*
- *x*
- *error*

13. Διάλεξε τι θα εμφανίσει στη οθόνη η παρακάτω εντολή:

```
>>> x="Maria"  
>>> y="Lena"  
>>> print x+y
```

Επίλεξε ένα:

- *MariaLena*
- *Maria Lena*
- *error*

14. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές αν σαν όνομα δώσω την τιμή Petros:

```
>>> name=input("Dose to onoma sou:")  
>>> print name
```

Επίλεξε ένα:

- *Petros*
- *error*
- *name*

15. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές αν σαν όνομα δώσω την τιμή 4:

```
>>> name=raw_input("Dose to onoma sou:")  
>>> print name
```

Επίλεξε ένα:

- 4
- *error*
- *name*

16. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές αν σαν όνομα δώσω την τιμή 4:

```
>>> name=raw_input("Dose to onoma sou:")  
>>> print name+4
```

Επίλεξε ένα:

- 4
- *error*
- 8

17. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x=3  
>>> y=12  
>>> x<1 and y!=x
```

Επίλεξε ένα:

- *True*
- *False*
- *true*

18. Διάλεξε τι θα εμφανίσει στη οθόνη η παρακάτω εντολή:

```
>>> 6==4*2 or 5!=5
```

Επίλεξε ένα:

- *False*
- *True*
- *false*

19. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x=9  
>>> y=3  
>>> not (x!=10) or (y>=1 or 4>10)
```

Επίλεξε ένα:

- *True*
- *False*
- *Error*

20. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x=6  
>>> z=15  
>>> print abs(x-z)
```

Επίλεξε ένα:

- *9*
- *-9*
- *0*

1.3 Ασκήσεις

- 1.** Να γράψετε πρόγραμμα που υπολογίζει το εμβαδό ενός τετραγώνου πλευράς a . Το πρόγραμμα διαβάζει από το πληκτρολόγιο το μήκος της πλευράς a , ζητώντας από το χρήστη να το πληκτρολογήσει. Στη συνέχεια υπολογίζει το εμβαδόν του τετραγώνου και εμφανίζει το αποτέλεσμα στην οθόνη με το ανάλογο μήνυμα.
- 2.** Να γράψετε πρόγραμμα που υπολογίζει τον μέσο όρο δυο αριθμών. Το πρόγραμμα διαβάζει από το πληκτρολόγιο τους αριθμούς. Στη συνέχεια υπολογίζει τον μέσο όρο και εμφανίζει το αποτέλεσμα στην οθόνη με το ανάλογο μήνυμα.
- 3.** Να γράψετε πρόγραμμα που κάνει χρηματική ανάλυση ενός ποσού. Το πρόγραμμα διαβάζει από το πληκτρολόγιο ένα ποσό. Στη συνέχεια υπολογίζει και εμφανίζει πόσα χαρτονομίσματα των 50 ευρώ περιλαμβάνει, στη συνέχεια πόσα των 20, κατόπιν πόσα των 10, έπειτα πόσα των 5 και τέλος ποιο είναι το υπόλοιπο σε κέρματα.
- 4.** Να γράψετε πρόγραμμα που ζητάει το όνομα και το έτος της γέννησής σας. Στη συνέχεια εμφανίζει με μια εντολή το όνομα και την ηλικία σας.
- 5.** Να γράψετε πρόγραμμα που διαβάζει από το πληκτρολόγιο 2 αριθμούς `num1` και `num2`. Στη συνέχεια υπολογίζει τη δύναμη του `num1` στη `num2` και εμφανίζει το αποτέλεσμα στην οθόνη με το ανάλογο μήνυμα.

2. Αλγοριθμικές δομές

2.1 Στοιχεία θεωρίας

2.1.1 Δομή Ακολουθίας

Οι εντολές εκτελούνται με τη σειρά που γράφονται. Η τελευταία εντολή σηματοδοτεί το τέλος του προγράμματος.

1.1.11 Δομή Επιλογής

Δομή Επιλογής if else

Σύνταξη:

```
if <συνθήκη ελέγχου>:
```

```
    εντολές
```

```
else:
```

```
    εντολές
```

Δομή Επιλογής if elif else

Σύνταξη:

```
if <συνθήκη ελέγχου 1>:
```

```
    εντολές
```

```
elif <συνθήκη ελέγχου 2>:
```

```
    εντολές
```

```
else:
```

```
    εντολές
```

1.1.12 Δομές Επανάληψης

Βρόχος for

Οι βρόχοι **for** εκτελούνται για **συγκεκριμένο πλήθος επαναλήψεων**. Για τη δημιουργία τους χρησιμοποιείται η συνάρτηση **range()**.

Σύνταξη:

for όνομα μεταβλητής in range (αρχή, μέχρι, βήμα):

Εντολή 1

Εντολή 2

.....

Εντολή ν

Η δομή της συνάρτησης **range** είναι της μορφής (αρχή, μέχρι, βήμα), όπου αρχή, μέχρι, βήμα, είναι ακέραιοι αριθμοί.

Οι ενδείξεις της αρχής και του βήματος δεν είναι υποχρεωτικές, αλλά μπορεί να προστεθούν. Αντίθετα η ένδειξη «μέχρι», πρέπει πάντα να αναφέρεται.

Βρόχος while

Ο βρόχος **while** χρησιμοποιείται για **μη προκαθορισμένο αριθμό επαναλήψεων**, όπου υπάρχει περίπτωση να μην εκτελεστούν οι εντολές του βρόχου, με τον έλεγχο της συνθήκης να πραγματοποιείται πριν από την εκτέλεση των εντολών του βρόχου.

Μια συνήθης εφαρμογή του βρόχου **while** είναι να ελέγχει την εγκυρότητα των δεδομένων εισόδου από τον χρήστη.

Σύνταξη:

```
Αρχική τιμή μεταβλητής  
while ονομα_μεταβλητής <συνθήκη>:  
    Εντολή 1  
    Εντολή 2  
    ....  
    Εντολή ν
```

Σημείωση1:

Θα πρέπει μέσα στο μπλοκ εντολών να υπάρχει κατάλληλη εντολή, ώστε να εξασφαλίζεται ότι κάποια στιγμή η συνθήκη θα γίνει ψευδής και θα διακοπεί ο βρόχος. Διαφορετικά, ο βρόχος δεν θα τερματίζει.

Σημείωση2:

Θα πρέπει πριν το βρόχο *while*, αρχικά να δώσουμε μια τιμή στη μεταβλητή που ελέγχει τη συνθήκη του βρόχου, ώστε αυτός να εκτελεστεί ή όχι.

2.2 Ερωτήσεις κατανόησης

1. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
x=2015
if x%2==0 :
    print '###'
else:
    print '$$$'
```

Επίλεξε ένα:

- 0
- \$\$\$
- ###

2. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών αν σαν είσοδο δοθούν οι τιμές 4 και 4.0;

```
x=input('Give a number: ')
y=input('Give a number: ')
z=x+y
if x>2*y:
    z=x-2*y
    if z>2:
        p=x+2*y-z
    else:
        p=z*(x-y)
else:
    if x!=0:
        z=(3.0*y)/x
    else:
```

```
z=3*y
p=z*(y-x)
print x,y,z,p
```

Επίλεξε ένα:

- *4 4.0 3.0 0.0*
- *4 4 3 0*
- *4 4.0 3.0 0*

3. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών αν σαν είσοδο δοθεί η τιμή 2013;

```
x=input('Give a year ')
if x>2013:
    print 'Future'
elif x<2013:
    print 'Past'
else:
    print 'This year'
```

Επίλεξε ένα:

- *Past*
- *Future*
- *This year*

4. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
x=5
If x>5*y:
    print '*x'
else:
    print 'y'
```

Επίλεξε ένα:

- *x*
- *SyntaxError*
- *y*

5. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
for i in range(3):  
    print '*',
```

Επίλεξε ένα:

- * * *
- * * * * *
- * * * * * *

6. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0  
for i in range(4):  
    s+=i  
print s,i
```

Επίλεξε ένα:

- 10 4
- 6 4
- 6 3

7. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0  
for i in range(2,4):  
    s+=i  
print s
```

Επίλεξε ένα:

- 5
- 2
- 9

8. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
```

```
for i in range(3,13,4):
```

```
    s+=i
```

```
print s
```

Επίλεξε ένα:

- 44
- 20
- 21

9. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
```

```
for i in range(12,3,-5):
```

```
    s+=i
```

```
print i
```

Επίλεξε ένα:

- 19
- -5
- 7

10. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
x=12
while x<20:
    s=s+x
    x=x+1
print s
```

Επίλεξε ένα:

- 124
- 123
- 0

11. Τι θα συμβεί μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
x=12
while x<20:
    s=s+x
    x=x-1
print s
```

12. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
c=0
x=100
while c<6:
    if x%5==0 and x%10!=0:
        print x
        c+=1
    x+=1
```

Επίλεξε ένα:

- *Εμφανίζει τους άρτιους αριθμούς από 100 μέχρι 116*
- *Εμφανίζει τους περιττούς αριθμούς από 100 μέχρι 116*
- *Εμφανίζει τους άρτιους αριθμούς από 100 μέχρι 118*

13. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
for k in range(6,6):
    s+=k
print s
```

Επίλεξε ένα:

- *20*
- *0*
- *10*

14. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
s=0
for k in range(5,7,-1):
    s+=k
print s
```

15. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
counter=0  
athroisma=0  
for i in range(5,10,2):  
    counter+=1  
print counter, athroisma
```

Επίλεξε ένα:

- 3 4
- 4 0
- 3 0
- 2 11

2.3 Ασκήσεις στη δομή επιλογής

1. Άρτιος λέγεται ένας ακέραιος αριθμός όταν διαιρείται ακριβώς με το 2. Όλοι οι άλλοι είναι περιττοί. (Το 0 είναι άρτιος). Να γραφεί πρόγραμμα που δέχεται από το πληκτρολόγιο έναν ακέραιο και στη συνέχεια τον χαρακτηρίζει σαν άρτιο ή περιττό.
2. Να γραφεί πρόγραμμα που δέχεται από το πληκτρολόγιο δυο αριθμούς και στη συνέχεια εμφανίζει το αποτέλεσμα της σύγκρισής τους.
3. Γράψτε ένα πρόγραμμα στο οποίο θα εισάγετε ένα κεφαλαίο γράμμα (A, B, C, D, και E) και θα σας δίνει ένα όνομα (Alice, Beatrice, Chris, κλπ) το οποίο αρχίζει από αυτό το γράμμα.
4. Να γραφεί πρόγραμμα που δέχεται από το πληκτρολόγιο ένα γράμμα και ελέγχει αν είναι φωνήεν (a, e, i, o, u) ή σύμφωνο, τυπώνοντας ταυτόχρονα και ένα κατάλληλο μήνυμα.
5. Γράψτε ένα πρόγραμμα το οποίο θα βρίσκει και θα τυπώνει τον μέγιστο μεταξύ τριών ακεραίων αριθμών που δίνετε από το πληκτρολόγιο.
6. Τροποποιήστε το πρόγραμμα ώστε να τυπώνει τον ελάχιστο μεταξύ των τριών ακεραίων αριθμών.
7. Ένα έτος είναι δίσεκτο αν διαιρείται με το τέσσερα αλλά όχι με το 100, με εξαίρεση τα έτη που διαιρούνται με το 400 τα οποία είναι δίσεκτα. Γράψτε ένα πρόγραμμα το οποίο να ελέγχει αν ένα έτος που δίνετε από το πληκτρολόγιο είναι δίσεκτο ή όχι.

- 8.** Γράψτε ένα πρόγραμμα το οποίο θα υπολογίζει και θα εμφανίζει στην οθόνη τις πραγματικές ρίζες μιας δευτεροβάθμιας εξίσωσης $ax^2+bx+c=0$. Επίσης, θα διακρίνει αν υπάρχουν δυο ρίζες ή μια (διπλή) δίνοντας τα αντίστοιχα μηνύματα. Τέλος, αν οι ρίζες είναι μιγαδικές θα βγαίνει το κατάλληλο μήνυμα. Από το πληκτρολόγιο δίνετε τους συντελεστες a , b , c .
- 9.** Γράψτε ένα πρόγραμμα στο οποίο δίνονται από το πληκτρολόγιο 3 αριθμοί. Αν υπάρχουν ανάμεσά τους ίδιοι, εμφανίζεται το μήνυμα «Υπάρχουν ίδιοι αριθμοί». Αν όχι, εμφανίζεται το μήνυμα «Όλοι οι αριθμοί είναι διαφορετικοί».
- 10.** Τροποποιήστε το προηγούμενο πρόγραμμα ώστε όταν υπάρχουν ίδιοι αριθμοί να εμφανίζει ποιοι είναι αυτοί. Αν όχι, εμφανίζεται το μήνυμα «Όλοι οι αριθμοί είναι διαφορετικοί».
- 11.** Γράψτε ένα πρόγραμμα στο οποίο δίνονται από το πληκτρολόγιο 4 αριθμοί. Στη συνέχεια βρίσκεται και εμφανίζεται ο μεγαλύτερος από αυτούς.
- 12.** Να γραφεί πρόγραμμα που θα δέχεται σαν είσοδο δύο ακέραιους αριθμούς. Αν είναι και οι δύο άρτιοι ή και οι δύο περιττοί, τότε να εμφανίζει τον μέσο όρο τους, σε διαφορετική περίπτωση να εμφανίζει την απόλυτη τιμή της διαφοράς τους.
- 13.** Να γραφεί πρόγραμμα που θα δέχεται σαν είσοδο δύο ακέραιους αριθμούς. Αν είναι ομόσημοι, τότε να εμφανίζει το δεκαδικό ημίγινόμενο τους, σε διαφορετική περίπτωση να εμφανίζει τη διαφορά τους.
- 14.** Το σούπερ μάρκετ της γειτονιάς σας, προκειμένου να προσελκύσει περισσότερους πελάτες αποφάσισε να χορηγεί

κουπόνια ανταπόδοσης που αντιστοιχούν σε δωροεπιταγές, όπως φαίνεται στον παρακάτω πίνακα.

Αριθμός κουπονιών	Αξία δωροεπιταγής
Από 500 μέχρι 1000	5€
Από 1001 μέχρι 2000	18€
Πάνω από 2000	54€

Να γράψετε πρόγραμμα που θα ζητάει από τον πελάτη τον αριθμό των κουπονιών του και να τον ενημερώνει για τη δωροεπιταγή που δικαιούται.

15. Το οικονομικό πακέτο μιας εταιρείας κινητής τηλεφωνίας παρέχει 200' ομιλίας προς όλους και 50MB internet, με πάγιο 22€. Κάθε μήνυμα χρεώνεται με 0,12€, ενώ κάθε επιπλέον λεπτό ομιλίας με 0,045€. Στην περίπτωση που ο συνδρομητής χρειαστεί επιπλέον MB, προσφέρονται 150MB με 5€.

Να γράψετε πρόγραμμα που θα ζητάει από τον συνδρομητή τα λεπτά ομιλίας του, τον αριθμό των SMS, καθώς και αν χρειάστηκε επιπλέον MB και θα υπολογίζει την τελική χρέωση. Η χρέωση αυτή θα επιβαρύνεται με 24% ΦΠΑ και θα εκτυπώνεται ο λογαριασμός προς τον πελάτη.

2.4 Ασκήσεις στη δομή for

1. Γράψτε κώδικα ο οποίος να περιέχει έναν βρόχο for για εκτύπωση των αριθμών από το 1 έως και το 10.
2. Γράψτε κώδικα ο οποίος να περιέχει έναν βρόχο for για αντίστροφη εκτύπωση των προηγούμενων αριθμών (από το 10 έως και το 1).
3. Γράψτε κώδικα ο οποίος να διαβάζει την ηλικία σας και στη συνέχεια να τυπώνει τους άρτιους αριθμούς μέχρι να φτάσει την ηλικία σας.
4. Να γράψετε πρόγραμμα το οποίο θα εκτυπώνει όλα τα πολλαπλάσια του 8 από το 0 μέχρι και το 50.
5. Να γράψετε πρόγραμμα το οποίο θα διαβάζει έναν ακέραιο αριθμό και θα εκτυπώνει την προπαίδεια αυτού του αριθμού.
6. Να γράψετε πρόγραμμα το οποίο θα εκτυπώνει όλους τους άρτιους από το 50 μέχρι το -10.
7. Να γράψετε πρόγραμμα το οποίο θα διαβάζει 10 ακεραίους από το πληκτρολόγιο και θα υπολογίζει και θα εκτυπώνει το μέσο όρο μόνο των αρτίων αριθμών που διαβάστηκαν.
8. Να γράψετε πρόγραμμα το οποίο διαβάζει 50 τυχαίους ακέραιους αριθμούς και θα εκτυπώνει τα ποσοστά των μονοψήφιων, διψήφιων και τριψήφιων αριθμών που διαβάστηκαν.

- 9.** Να γράψετε πρόγραμμα το οποίο εκτυπώνει όλους τους θετικούς διψήφιους ακραίους. Τροποποιήστε το πρόγραμμα ώστε να υπολογίζει και να εμφανίζει το άθροισμα των αριθμών αυτών καθώς και τον μέσο όρο τους.
- 10.** Να γράψετε πρόγραμμα το οποίο θα διαβάζει 20 ακραίους από τον πληκτρολόγιο και θα υπολογίζει και θα εκτυπώνει το πλήθος των άρτιων αριθμών που διαβάστηκαν καθώς και το πλήθος των αριθμών που είναι μεγαλύτεροι από 15.
- 11.** Να γράψετε πρόγραμμα που θα εκτυπώνει τους τριψήφιους αριθμούς που είναι πολλαπλάσια του 7 καθώς και το πόσοι είναι οι αριθμοί αυτοί.
- 12.** Να γράψετε πρόγραμμα που θα εκτυπώνει το άθροισμα των τετραγώνων των μονοψήφιων άρτιων αριθμών.
- 13.** Σε ένα αγώνα ρίψης ακοντίου, διεξάγεται ο προκριματικός γύρος με τη συμμετοχή 14 αθλητών. Στην τελική φάση προκρίνονται όσοι αθλητές πετύχουν επίδοση άνω των 80 μέτρων. Να γράψετε πρόγραμμα που να διαβάζει την επίδοση κάθε αθλητή, να υπολογίζει και να εμφανίζει πόσοι αθλητές πέρασαν το όριο.
- 14.** Να γράψετε ένα πρόγραμμα που με χρήση ένθετων βρόχων for τυπώνει το παρακάτω:
- ```
*
* *
* * *
* * * *
```

**15.** Συμπληρώστε το προηγούμενο πρόγραμμα ώστε να τυπώνει το παρακάτω:

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

## 2.5 Ασκήσεις στη δομή while

1. Να γράψετε πρόγραμμα το οποίο θα τυπώνει τους 20 πρώτους θετικούς ακεραίους αριθμούς με χρήση πρώτα της for και στη συνέχεια της while.
2. Να γράψετε πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι να δοθεί ο αριθμός 0. Το πρόγραμμα θα υπολογίζει και εκτυπώνει:
  - α) Πόσοι αριθμοί διαβάστηκαν (εκτός του 0).
  - β) Το πλήθος των ζυγών αριθμών.
  - γ) Το άθροισμα και τον μέσο όρο των μονών αριθμών.
3. Να γράψετε πρόγραμμα το οποίο θα διαβάζει το όνομα και τους βαθμούς ενός μαθητή της Α΄ Λυκείου σε όλα τα μαθήματα στο Α΄ Τετράμηνο και στη συνέχεια θα υπολογίζει και θα εκτυπώνει:
  - α) τον μέσο όρο του τετραμήνου (15 μαθήματα).
  - β) το πλήθος των μαθημάτων που είχε βαθμό πάνω από 16
  - γ) τον μεγαλύτερο και τον μικρότερο βαθμό
  - δ) το ποσοστό των μαθημάτων που πήρε βαθμό από 18-20.
4. Να γράψετε πρόγραμμα που να βρίσκει σε ποιον όρο το άθροισμα  $1+2+3+4+\dots$  γίνεται μεγαλύτερο του 2000.
5. Να γράψετε πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο συνεχώς και θα σταματάει μόλις διαβάσει έναν αρνητικό αριθμό. Για κάθε αριθμό που διαβάζει θα εμφανίζει το τετράγωνο του. Επίσης το πρόγραμμα θα υπολογίζει πόσοι αριθμοί διαβάστηκαν (χωρίς τον αρνητικό).

**6.** Να γράψετε πρόγραμμα που:

- α) θα διαβάζει επαναληπτικά το όνομα και τον βαθμό στο μάθημα του Προγραμματισμού των μαθητών μιας τάξης μέχρι να διαβάσει τον χαρακτήρα \$ ως όνομα μαθητή.
  - β) θα υπολογίζει και εκτυπώνει τον μέσο όρο του τμήματος
  - γ) Το όνομα του μαθητή με τον μεγαλύτερο βαθμό (υπάρχει μόνο ένας)
  - δ) Το ποσοστό των μαθητών που πήραν βαθμό κάτω από 10
- Σημείωση: Υπάρχει τουλάχιστον ένας μαθητής στο τμήμα.

**7.** Το παραγοντικό ενός αριθμού  $n$  δίνεται από τον τύπο  $n!=1*2*3*...*n$ . Να γράψετε ένα πρόγραμμα που ζητά από το πληκτρολόγιο έναν ακέραιο και υπολογίζει το παραγοντικό του. Το παραγοντικό του 0 είναι 1.

Συμπληρώστε το πρόγραμμα ώστε να ζητά συνέχεια αριθμό και να υπολογίζει το παραγοντικό του, μέχρι να δώσουμε αριθμό αρνητικό.

**8.** Από τις παρατηρήσεις ενός μελισσοκόμου προκύπτει ότι μια κοινότητα μελισσών υπό κανονικές συνθήκες αναπτύσσεται με ρυθμό 3.8% ετησίως. Αν ένας μελισσοκόμος διαθέτει μελίσσια με συνολικό πληθυσμό 1200 μέλισσες, να γράψετε πρόγραμμα που υπολογίζει και εμφανίζει σε πόσα έτη θα ξεπεράσει τη χωρητικότητα των κυψελών του που είναι 2000 μέλισσες.

**9.** Να συμπληρώσετε το προηγούμενο πρόγραμμα ώστε να δίνουμε από το πληκτρολόγιο τον αρχικό αριθμό των μελισσών καθώς και τον αριθμό των μελισσών που δεν πρέπει να ξεπεράσει.

**10.** Να γράψετε πρόγραμμα το οποίο θα διαβάζει τυχαίους αριθμούς από το πληκτρολόγιο συνεχώς μέχρις ότου το



άθροισμα τους να ξεπεράσει το 150 ή μόλις το πλήθος των αριθμών που διαβάστηκαν φτάσει τους 10. Τότε σταματά και εμφανίζει πόσοι αριθμοί διαβάστηκαν καθώς και το άθροισμά τους.

**11.** Να γράψετε πρόγραμμα που εμφανίζει τους 10 πρώτους αριθμούς της ακολουθίας Fibonacci.

*Στα Μαθηματικά, οι αριθμοί Fibonacci είναι οι αριθμοί της ακέραιας ακολουθίας: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...*

*Εξ ορισμού, οι πρώτοι δύο αριθμοί Fibonacci είναι το 0 και το 1, και κάθε επόμενος αριθμός είναι το άθροισμα των δύο προηγούμενων.*



## 3. Συναρτήσεις

### 3.1 Στοιχεία θεωρίας

Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα μέρη προγραμμάτων. Μας επιτρέπουν να δίνουμε ένα όνομα σε ένα σύνολο εντολών και να το εκτελούμε καλώντας το όνομά τους, οπουδήποτε στο πρόγραμμα και όσες φορές θέλουμε.

Αυτή η διαδικασία ονομάζεται κλήση (call) της συνάρτησης.

Υπάρχουν δυο είδη συναρτήσεων:

**Ενσωματωμένες συναρτήσεις** όπως π.χ. `int()`, `abs()`, `float()`, `sqrt()`

**Συναρτήσεις του χρήστη**, δηλαδή αυτές που φτιάχνουμε εμείς για τις ανάγκες του προγράμματός μας.

Οι συναρτήσεις του χρήστη ορίζονται χρησιμοποιώντας τη δεσμευμένη λέξη **def** (από το *define* που σημαίνει *ορίζω*). Στη συνέχεια βάζουμε το όνομα της συνάρτησης. Μετά προσθέτουμε ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν μερικά ονόματα μεταβλητών και η γραμμή τελειώνει με άνω και κάτω τελεία σύμφωνα με τη σύνταξη:

```
def <όνομα συνάρτησης> ([{λίστα παραμέτρων}]):
 εντολές
 [return <αποτέλεσμα>]
```

Η λίστα των παραμέτρων μπορεί να είναι κενή. Μια συνάρτηση δεν είναι υποχρεωτικό να επιστρέφει κάποια τιμή.

Η συνάρτηση δέχεται δεδομένα μέσω των παραμέτρων και επιστρέφει τα αποτελέσματα μέσω άλλων ή και των ίδιων παραμέτρων στο πρόγραμμα ή σε άλλη συνάρτηση.

Οι **παράμετροι** καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα. Όταν καλούμε τη συνάρτηση, δίνουμε και τις τιμές με τον ίδιο τρόπο, οι οποίες τιμές ονομάζονται **ορίσματα**.

Κάποιες από τις ενσωματωμένες συναρτήσεις, που έχουμε ήδη συναντήσει, δεν απαιτούν ορίσματα, όπως για παράδειγμα, όταν καλούμε την `math.pi()`, όπου δεν έχουμε κάποιο όρισμα. Σε άλλες όμως συναρτήσεις απαιτούνται ένα ή και περισσότερα ορίσματα, όπως στην `math.pow()` όπου απαιτούνται δύο, ένα για τη βάση και ένα για τον εκθέτη.

## 3.2 Ερωτήσεις κατανόησης

1. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
def func(a,b):
 c=a+b
 return c
print func('Nikos','Maria')
```

Επίλεξε ένα:

- Nikos Maria
- NikosMaria
- Nikos

2. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
def func(a,b,c):
 c=(a+b)*c
 return c
print func('Nikos','Maria',3)
```

Επίλεξε ένα:

- NikosMaria3
- Error
- NikosMariaNikosMariaNikosMaria

3. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
def func(a,b,c):
 a+=1
```

```
z=(a+b)/c
return z
a=12
b=3
c=1.0
d=func(a,b,c)
print (a,b,c,d)
```

Επίλεξε ένα:

- (12.0, 3.0, 1.0, 16.0)
- (12, 3, 1, 16)
- Error

4. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
def func2(x,y):
 if x>y:
 z=x-y
 else:
 z=x+y
 return z
z=func2(3,2)
print z
```

Επίλεξε ένα:

- 1
- 5
- error

5. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
def fun1(a,b,c):
 b+=a
 d=(a+b+c)/3
```

```
print fun1(1,2,3)
```

Επίλεξε ένα:

- 2
- 2.0
- None

6. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def func(x,y):
 z=x**y
 return z
a=2
b=5.0
```

```
print func (a,b)
```

Επίλεξε ένα:

- 32
- None
- 32.0

7. Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def func(x,y):
 z=x+y
 return z
a='Trikala'
```

```
b='2021'
print func(a,b)
```

Επίλεξε ένα:

- **Trikala 2021**
- **Error**
- **Trikala2021**

**8.** Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def func(a,b):
 if a>b:
 m=a
 else:
 m=b
 return m
m=3
k=-3
print func(k,m)
```

Επίλεξε ένα:

- **3**
- **None**
- **-3**

**9.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def func(a,b):
 if a>b:
 m=a
 else:
 m=b
```



```
a=3
b=-3
print func(a,b)
```

Επίλεξε ένα:

- 3
- 0
- None

**10.** Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def diafora(a,b):
 x=a-b
 return x
```

```
a=1
b=2
print x
```

Επίλεξε ένα:

- 1
- Error
- -1

**11.** Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def diafora(a,b):
 x=a-b
 return x
```

```
x=1
y=2
print x
```

Επίλεξε ένα:

- 1
- Error
- -1

**12.** Διάλεξε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def diaf(a,b):
```

```
 x=a-b
```

```
a=5
```

```
b=3
```

```
print diaf(a,b)
```

Επίλεξε ένα:

- 2
- -2
- None

## 3.3 Ασκήσεις

1. Γράψτε μια συνάρτηση που δέχεται σαν παράμετρο έναν αριθμό και επιστρέφει το τριπλάσιό του.
2. Γράψτε μια συνάρτηση που δέχεται σαν παραμέτρους δύο αριθμούς και επιστρέφει την απόλυτη τιμή της διαφοράς τους.
3. Γράψτε μια συνάρτηση που δέχεται σαν παράμετρο την απόσταση σε χιλιόμετρα και επιστρέφει την απόσταση σε μίλια.  
(1 μίλι = 1.6 χιλιόμετρα).
4. Γράψτε μια συνάρτηση που δέχεται σαν παράμετρο τρεις αριθμούς μεταξύ 0 και 100 και επιστρέφει τον μέσο όρο τους.
5. Γράψτε μια συνάρτηση η οποία δέχεται έναν χαρακτήρα και ελέγχει αν είναι φωνήεν ή όχι και εμφανίζει αυτή την πληροφορία.
6. Γράψτε ένα πρόγραμμα το οποίο δέχεται άγνωστο πλήθος χαρακτήρων μέχρι να πληκτρολογήσουμε τον χαρακτήρα τελεία και υπολογίζει:
  - α. Τον συνολικό αριθμό των φωνηέντων που πληκτρολογήθηκαν χρησιμοποιώντας τη συνάρτηση της προηγούμενης άσκησης.
  - β. Το ποσοστό των φωνηέντων σε σχέση με όλους τους άλλους χαρακτήρες που πληκτρολογήθηκαν.
7. Γράψτε μια συνάρτηση που προσομοιώνει το ρίξιμο ενός ζαριού και εμφανίζει τη ζαριά.
  - α. Γράψτε ένα πρόγραμμα που ρωτά πόσες φορές θα ρίξουμε το ζάρι και στη συνέχεια καλεί τη συνάρτηση.

- β. Τροποποιήστε το πρόγραμμα ώστε να ρωτάει συνέχεια αν θα ρίξουμε το ζάρι. Πατώντας «Υ» καλεί τη συνάρτηση που εμφανίζει τη ζαριά. Πατώντας οτιδήποτε άλλο, το πρόγραμμα τερματίζει.

8. Η εφορία κάθε τέλος έτους φορολογεί τα αυτοκίνητα ανάλογα με τον κυβισμό τους, σύμφωνα με τον παρακάτω πίνακα:

| Κυβισμός      | Φόρος |
|---------------|-------|
| 0 έως 999     | 110 € |
| 1000 έως 1399 | 150 € |
| 1400 έως 1999 | 225 € |
| 2000 και πάνω | 600 € |

- α. Να γράψετε μια συνάρτηση που θα δέχεται τον κυβισμό ενός αυτοκινήτου και θα επιστρέφει την αντίστοιχη φορολογία.
- β. Να γράψετε το κύριο πρόγραμμα το οποίο για μια εταιρεία που διαθέτει 10 αυτοκίνητα θα διαβάζει τα κυβικά για το κάθε ένα και θα υπολογίζει τη φορολογία για το καθένα.
- γ. Συμπληρώστε το πρόγραμμα ώστε να επιστρέφει το συνολικό ποσό που πρέπει να πληρώσει η εταιρεία για όλα τα αυτοκίνητά της.
9. Να γράψετε μια συνάρτηση η οποία θα δέχεται από το κυρίως πρόγραμμα τις ώρες εργασίας και το ωρομίσθιο ενός υπαλλήλου και θα επιστρέφει σε αυτό τα εξής:
- α. Τον ακαθάριστο μισθό του υπαλλήλου (γινόμενο των ωρών εργασίας και του ωρομισθίου).
- β. Τον φόρο εισοδήματος (20% του ακαθαρίστου μισθού).
- γ. Τις κοινωνικές ασφάλισεις (6% του ακαθαρίστου μισθού).
10. Να γράψετε ένα πρόγραμμα το οποίο θα δέχεται το όνομα, τις ώρες εργασίας και την ωριαία αμοιβή 20 υπαλλήλων και

κάνοντας χρήση της πιο πάνω συνάρτησης να υπολογίζει και να εμφανίζει το όνομα του, τον ακαθάριστο μισθό του, τον φόρο εισοδήματος, τις κοινωνικές ασφαλίσεις και τον καθαρό μισθό του κάθε υπαλλήλου.

11. Μια εταιρεία έχει 48 υπαλλήλους και τους πληρώνει κλιμακωτά κάθε εβδομάδα σύμφωνα με τον παρακάτω πίνακα.

| ΩΡΕΣ ΕΡΓΑΣΙΑΣ     | €/ΩΡΑ |
|-------------------|-------|
| από 0 έως και 40  | 5     |
| από 41 έως και 50 | 7     |
| πάνω από 50       | 9     |

Για παράδειγμα, ένας υπάλληλος που δούλεψε 55 ώρες σε μία εβδομάδα θα πληρωθεί  $40 \cdot 5 + 10 \cdot 7 + 5 \cdot 9 = 315\text{€}$ .

Να γράψετε πρόγραμμα το οποίο:

- Να διαβάζει το όνομα του κάθε υπαλλήλου.
- Να διαβάζει τον αριθμό των ωρών εργασίας κάθε υπαλλήλου σε μία εβδομάδα, ελέγχοντας ότι είναι αριθμός μεγαλύτερος ή ίσος του μηδενός.
- Να υπολογίζει με τη χρήση συνάρτησης την εβδομαδιαία αμοιβή κάθε υπαλλήλου.
- Να εμφανίζει το όνομα και την εβδομαδιαία αμοιβή κάθε υπαλλήλου.
- Να υπολογίζει και να εμφανίζει το συνολικό κόστος μισθοδοσίας της εταιρείας.
- Να υπολογίζει και να εμφανίζει το ποσοστό των εργαζομένων που πήραν μισθό μεγαλύτερο από 700€.



## 4. Δομές Δεδομένων

### 4.1 Στοιχεία Θεωρίας

Δομή Δεδομένων ονομάζεται ένα σχήμα οργάνωσης των δεδομένων τα οποία χρησιμοποιεί το πρόγραμμά μας.

#### Στατικές Δομές Δεδομένων

Το μέγεθος μιας στατικής δομής καθορίζεται κατά τη συγγραφή του προγράμματος από τον προγραμματιστή, ώστε να είναι γνωστό κατά τη φάση της μεταγλώττισης. Όταν το πρόγραμμα ξεκινήσει να εκτελείται δεσμεύεται χώρος στη μνήμη για τις στατικές δομές. Το μέγεθος των στατικών δομών παραμένει σταθερό κατά την εκτέλεση του προγράμματος, αφού δεν μπορούμε να αφαιρέσουμε ούτε να προσθέσουμε αντικείμενα στις δομές αυτές.

#### Δυναμικές Δομές Δεδομένων

Στηρίζονται σε μια λειτουργία που λέγεται δυναμική εκχώρηση μνήμης. Κατά τη δυναμική εκχώρηση μνήμης, το πρόγραμμα μπορεί να ζητήσει από το λειτουργικό σύστημα όση μνήμη απαιτείται για τη δημιουργία της δομής δεδομένων κατά την εκτέλεση του προγράμματος. Οι δυναμικές δομές δεδομένων μπορούν να μεταβάλλουν το μέγεθός τους, προσθέτοντας ή αφαιρώντας αντικείμενα. Η πιο γνωστή δυναμική δομή δεδομένων είναι η λίστα, η οποία είναι και η βασική δομή δεδομένων της Python.

### 4.1.1 Συμβολοσειρά

Συμβολοσειρά ή αλφαριθμητικό ονομάζεται μια ακολουθία από χαρακτήρες και μπορεί να αποτελείται από περισσότερες από μία λέξεις, που μπορούν να είναι στην Ελληνική Γλώσσα, στην Αγγλική ή σε κάθε γλώσσα που υποστηρίζεται από το πρότυπο Unicode. Μια συμβολοσειρά ορίζεται με εισαγωγικά αμφίπλευρα, μονά ή διπλά.

Στην Python υπάρχει ο τύπος **str** στον οποίο οι συμβολοσειρές αποτελούν ακολουθίες χαρακτήρων. Η προσπέλαση σε κάθε χαρακτήρα της συμβολοσειράς γίνεται με τον αριθμό της θέσης που βρίσκεται, μέσα σε αγκύλες. Η αρίθμηση στην Python ξεκινάει από το 0.

Η συνάρτηση **len** μας επιστρέφει το μήκος της συμβολοσειράς.

Ένας σημαντικός τελεστής που χρησιμοποιείται και στις λίστες είναι ο τελεστής **in**, ο οποίος ελέγχει αν ένα αντικείμενο ανήκει σε ένα σύνολο αντικειμένων.

Επίσης, οι γνωστοί συγκριτικοί τελεστές, οι οποίοι συγκρίνουν με βάση τη λεξικογραφική διάταξη των χαρακτήρων, ισχύουν και στις συμβολοσειρές.

Μπορούμε να πάρουμε ένα τμήμα της συμβολοσειράς με χρήση του **τελεστή διαμέρισης :** (slice operator). Στη βιβλιογραφία το τμήμα αυτό της συμβολοσειράς αναφέρεται και ως **φέτα** (slice).

Τα αντικείμενα του τύπου **str** δεν είναι τροποποιήσιμα, είναι όπως λέμε αμετάβλητα (immutable). Δηλαδή δεν μπορούμε να αλλάξουμε κάποιο τμήμα της συμβολοσειράς.

Οποιαδήποτε επεξεργασία θέλουμε να κάνουμε σε συμβολοσειρές γίνεται με χρήση του τελεστή φέτας για αποκοπή του μέρους που θέλουμε και του τελεστή **+** για συνένωση.



### 4.1.2 Λίστα

Η λίστα είναι ουσιαστικά μια διατεταγμένη ακολουθία από αντικείμενα τα οποία είναι του ίδιου ή και διαφορετικού τύπου. Για κάθε αντικείμενο που εισάγεται στη λίστα δίνεται ένας αύξων αριθμός, που χρησιμοποιείται για την αναφορά του στο αντικείμενο. Μπορεί κανείς ανά πάσα στιγμή να προσθέσει, να διαγράψει ή να αλλάξει ένα αντικείμενο σε μία λίστα.

Η προσπέλαση στα στοιχεία της λίστας γίνεται ακριβώς όπως στις συμβολοσειρές. Η λίστα μπορεί και αυτή να θεωρηθεί ως ένα σύνολο από αντικείμενα, οπότε μπορούμε να χρησιμοποιήσουμε τον υπαρξιακό τελεστή **in** και τη συνάρτηση **len**.

Αν δεν θέλουμε να θέσουμε κάποια αρχική τιμή στα στοιχεία της λίστας μπορούμε να δώσουμε την τιμή **None**, που υποδηλώνει την απουσία τιμής, δηλαδή `array = [None]*n` (όπου *n* το πλήθος των στοιχείων της λίστας).

Μπορούμε να χρησιμοποιήσουμε τον τελεστή **:** όπως και στις συμβολοσειρές. Εδώ να σημειώσουμε ότι σε αντίθεση με τις συμβολοσειρές, οι οποίες δεν μπορούν να τροποποιηθούν, τα στοιχεία των λιστών μπορούν να αλλάξουν.

Οι λίστες, όπως και οι συμβολοσειρές, έχουν κάποιες εγγενείς λειτουργίες τις οποίες μπορούμε να καλέσουμε και τις οποίες ονομάζουμε μεθόδους.

Κάποιες από αυτές είναι οι παρακάτω:

**append** για προσθήκη στοιχείου στο τέλος της λίστας

**insert** για προσθήκη στοιχείου σε συγκεκριμένη θέση στη λίστα.

**pop** για αφαίρεση στοιχείου από λίστα. Αν καλέσουμε την `pop` χωρίς ορίσματα αφαιρεί το τελευταίο στοιχείο της λίστας ενώ αν δώσουμε ένα όρισμα όπως τον αριθμό 2 αφαιρείται το στοιχείο που βρίσκεται στη θέση 2.

Ο τελεστής **φέτα** είναι πολύ σημαντικός στην επεξεργασία λιστών στην Python, γιατί μπορούμε να τον χρησιμοποιήσουμε για να

πάρουμε ένα αντίγραφο μιας λίστας. Προσοχή! δεν μπορεί να γίνει το ίδιο με τις συμβολοσειρές.

Ένας δεύτερος τρόπος για να δημιουργήσουμε αντίγραφο μιας λίστας είναι έμμεσα με χρήση του τελεστή `+` ο οποίος δημιουργεί μια νέα λίστα, ως άθροισμα δύο ήδη υπάρχουσών λιστών. Για να δημιουργήσουμε αντίγραφο μιας λίστας αρκεί να προσθέσουμε την κενή `[]`.

### **Σημαντική σημείωση**

*Η εντολή `aList = aList + [ item ]`*

*κάθε φορά που καλείται δημιουργεί μια νέα λίστα ως προϊόν συνένωσης των δυο λιστών, η οποία αποθηκεύεται σε μια νέα θέση στη μνήμη. Αυτή η λειτουργία έχει υπολογιστικό κόστος σε αντίθεση με την εντολή `aList.append( item )` η οποία προσθέτει το στοιχείο `item` στο τέλος της λίστας.*

## 4.2 Ερωτήσεις κατανόησης στις συμβολοσειρές

1. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a='Trikala'
>>> b='2021'
>>> z=a+b
>>> z
```

Επίλεξε ένα:

- Trikala 2021
- Trikala2021
- Error

2. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x='1ο EPAL Trikalon'
>>> len(x)
```

Επίλεξε ένα:

- 14
- 15
- 16

3. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x='1ο EPAL Trikalon'
>>> y=x[5]+x[-2]
>>> print y
```

Επίλεξε ένα:

- Αο
- L
- ka

4. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x= '1ο EPAL Trikalon'
>>> print x+len(x)
```

Επίλεξε ένα:

- 1ο EPAL Trikalon16
- 1ο EPAL Trikalon 16
- error

5. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> a='1ο EPAL'
>>> b=a[3]+a[3:4]
>>> print b
```

Επίλεξε ένα:

- EE
- 'οΕ'
- none

6. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x='1ο EPAL'
>>> y=x[:]
>>> print y
```

Επίλεξε ένα:

- 1ο EPAL
- IndexError
- None

**7.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x='1ο EPAL'
>>> x[2]= '#'
>>> print x
```

Επίλεξε ένα:

- TypeError
- 1ο#EPAL
- Index error

**8.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
def func(a,b):
 return a*b
print mul('1EPAL','2')
```

Επίλεξε ένα:

- 1EPAL
- TypeError
- 1EPAL1EPAL

**9.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x= 'Trikala'
>>> x.upper()
```

Επίλεξε ένα:

- TRIKALA
- trikala
- Trikala

**10.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> x='1EPAL Trikalon'
```

```
>>> 'a' in x
```

Επίλεξε ένα:

- True
- 3
- 10

## 4.3 Ασκήσεις στις συμβολοσειρές

1. Γράψτε ένα πρόγραμμα που θα ζητά να δώσουμε ένα αλφαριθμητικό και στη συνέχεια έναν χαρακτήρα. Στη συνέχεια θα μετρά πόσες φορές υπάρχει ο χαρακτήρας αυτός στο αλφαριθμητικό και θα εμφανίζει τον αριθμό. (Υπάρχουν 2 τρόποι επίλυσης της άσκησης).
2. Γράψτε ένα πρόγραμμα που θα ζητά να δώσουμε μια πρόταση. Στη συνέχεια θα περικόπτει τα κενά και θα εμφανίζει τη νέα πρόταση.
3. Γράψτε ένα πρόγραμμα που θα ζητά να δώσουμε ένα αλφαριθμητικό. Στη συνέχεια δημιουργεί το αντίστροφο αλφαριθμητικό και το εμφανίζει.
4. Γράψτε μια συνάρτηση που δέχεται σαν παράμετρο μια πρόταση και επιστρέφει το πλήθος των λέξεων που περιέχει. Θεωρήστε ότι οι λέξεις χωρίζονται με το κενό.
5. Γράψτε μια συνάρτηση που δέχεται μία πρόταση και επιστρέφει ένα νέο αλφαριθμητικό που αποτελείται από τους 10 πρώτους χαρακτήρες της. Συμπληρώστε την άσκηση ώστε να μην περιλαμβάνει τα κενά στο νέο αλφαριθμητικό, έστω και αν εμφανίζει λιγότερους χαρακτήρες.
6. Γράψτε μια συνάρτηση που δέχεται μία φράση και την επιστρέφει κάνοντας κεφαλαίο το πρώτο γράμμα κάθε λέξης. Θεωρήστε ότι οι λέξεις χωρίζονται με το κενό.

7. Γράψτε μια συνάρτηση που δέχεται ένα αλφαριθμητικό και ελέγχει αν αυτό αποτελεί έγκυρο email, δηλαδή περιέχει τον χαρακτήρα @ και τελειώνει με .gr
8. Γράψτε ένα πρόγραμμα που με τη βοήθεια συνάρτησης υπολογίζει και επιστρέφει το πλήθος των φωνηέντων της λέξης (στα αγγλικά) που δέχεται ως παράμετρο.
9. Να γράψετε μια συνάρτηση η οποία θα δέχεται μια λέξη και θα επιστρέφει το μήνυμα «Καρκινική» αν η λέξη μπορεί να διαβαστεί και ανάποδα ή «Μη Καρκινική» αν δεν διαβάζεται ανάποδα.



## 4.4 Ερωτήσεις κατανόησης στις λίστες

1. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=[5,8,14,7,3]
>>> len(list1)+list1[2]
```

Επίλεξε ένα:

- *11*
- *18*
- *19*

2. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> thessalia=['Larisa','Magnisia','Karditsa','Trikala']
>>> Trikala in thessalia
```

Επίλεξε ένα:

- *True*
- *NameError*
- *False*

3. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=[5, 8, 1, 2, 3]
>>> list1[-2]=11
>>> print list1
```

Επίλεξε ένα:

- *Error*
- *[5,8,1,11,3]*
- *[5, 8, 11, 1, 3]*

4. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=[5, 8, 1, 2, 3]
>>> list1[-6]=11
```

Επίλεξε ένα:

- *IndexError*
- *[5,8,1,11,3]*
- *[11,5, 8, 11, 1]*

5. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=[15,8,14,7,13]
>>> list2=[0,0]
>>> list3=list1+list2
>>> list3
```

Επίλεξε ένα:

- *[15,8,14,7,13]*
- *[5, 8, 14, 7, 3, 0, 0]*
- *error*

6. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=[5,8,14,7,3]
>>> list2=[5]+list1[2:3]
>>> print list2
```

Επίλεξε ένα:

- *[5, 8, 14]*
- *[5, 14]*
- *IndexError*

7. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=['a', 'bc']
>>> list1.append(4)
>>> print list1
```

Επίλεξε ένα:

- *['a', 'bc', 4]*
- *error*
- *['a', 'bc']*

8. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
>>> list1=['a',15, 'bc']
>>> list1.append(12)
>>> list1.pop(0)
```

Επίλεξε ένα:

- *b*
- *a*
- *none*

9. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=['a',15, 'bc']
list1.insert(0,'d')
print list1
```

Επέλεξε ένα:

- *['d', 'a', 15, 'bc']*
- *['a', 'd', 15, 'bc']*
- *indexError*

```
10. list1=['a',15, 'bc']
list1.insert(-1,'d')
print list1
```

- *['a', 15, 'd', 'bc']*
- *['a', 15, 'bc'm 'd']*
- *indexError*

11. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[None]*4
list1.insert(2,'r')
print list1
```

Επίλεξε ένα:

- *[None]*
- *error*
- *[None, None, 'r', None, None]*

12. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[]
list2=list1+[None]
print list2
```

Επίλεξε ένα:

- *[None]*
- *error*
- *[ None]*

- 13.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[12, 6, 14]
list2=list1+list1.pop()
print list2
```

Επίλεξε ένα:

- *[12, 6]*
- *[12, 6, 14, 14]*
- *TypeError*

- 14.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[]
for i in range(5):
 list1.append(i)
print list1
```

Επίλεξε ένα:

- *[0, 1, 2, 3, 4]*
- *[]*
- *error*

- 15.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[1, 2, 3, 4]
for i in len(list1):
 if list1[i]>3:
 list1.pop()
print list1
```

Επίλεξε ένα:

- *[1, 2, 3]*
- *TypeError*
- *[1, 2, 3, 4]*

**16.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[1, 2, 3, 4]
for i in range(len(list1)):
 if list1[i]>3:
 list1.pop()
print list1
```

Επίλεξε ένα:

- *[1, 2, 3]*
- *TypeError*
- *[1, 2, 3, 4]*

**17.**

```
list1=[1, 2, 3, 4]
for i in range(len(list1)):
 if list1[i]>3:
 list1.append(8)
print list1
```

Επίλεξε ένα:

- *[1, 2, 3, 4, 8]*
- *TypeError*
- *[1, 2, 3, 4, 8, 8]*

**18.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[1, 2, 3, 4]
for i in range(len(list1)):
 if list1[i]>3:
 list1.append('*')
 else:
 list1.append('$')
print list1
```

Επίλεξε ένα:

- *[1, 2, 3, 4]*
- *[1, 2, 3, 4, '\*', '\$']*
- *TypeError*

**19** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[1, 2, 3, 4]
a=len(list1)
i=0
while i<a
 list1.append('*')
 i+=1

print list1
```

Επίλεξε ένα:

- *[1, 2, 3, 4, '\*', '\*', '\*', '\*']*
- *[1, 2, 3, 4, '\*', '\*', '\*']*
- *IndexError*

20. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[]
a=len(list1)
i=0
while i<a
 list1.append('*')
 i+=1
```

print list1

Επίλεξε ένα:

- *[1, 2, 3, 4, '\*', '\*', '\*', '\*']*
- *[]*
- *IndexError*

21. Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[None]*5
a=len(list1)
i=0
while i<a
 list1.insert(i,'*')
 i+=1
```

print list1

Επίλεξε ένα:

- *['\*', '\*', '\*', '\*', '\*']*
- *['\*', '\*', '\*', '\*', '\*', None, None, None, None, None]*
- *IndexError*



**22.** Γράψε τι θα εμφανίσουν στη οθόνη οι παρακάτω εντολές:

```
list1=[1, 2, 3, 4]
```

```
list2=['a', 'b', 'c']
```

```
list3=[]
```

```
i=0
```

```
while i<len(list1):
```

```
 list3=list1+list2
```

```
 i+=1
```

```
print list3
```

- *[1, 2, 3, 4, 'a', 'b', 'c']*
- *[1, 'a', 3, 'b', 4]*
- *IndexError*

## 4.5 Ασκήσεις στις λίστες

1. Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι το άθροισμά τους να ξεπεράσει το 1000, θα τους αποθηκεύει σε μια λίστα και στη συνέχεια θα υπολογίζει και θα εμφανίζει:

- α. το άθροισμά τους.
- β. τον μέσο όρο των στοιχείων της λίστας.
- γ. τη μέγιστη τιμή των στοιχείων της λίστας και τη θέση της.

2. Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει αριθμούς από το πληκτρολόγιο, μέχρι να δοθεί ο αριθμός 0. Στη συνέχεια με τη χρήση λίστας θα εμφανίζει τους αριθμούς σε αντίστροφη σειρά από αυτή με την οποία τους διάβασε.

3. Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο 20 ακραίους αριθμούς και θα τους καταχωρεί σε μια λίστα. Στη συνέχεια θα διαχωρίζει τους αριθμούς σε δυο νέες λίστες, μια για τους θετικούς και μια για τους αρνητικούς. Οι αριθμοί πρέπει να παραμείνουν στη σειρά με την οποία δόθηκαν.

4. Γράψτε ένα πρόγραμμα στο οποίο δίνονται οι λίστες `list1=[4,5,8,12,4]` και `list2=[-4,9,11,6,7]` και δημιουργεί μια νέα λίστα `list3` που κάθε στοιχείο της είναι το άθροισμα των αντίστοιχων στοιχείων των λιστών `list1`, `list2`. Να εμφανίζονται τα στοιχεία της νέας λίστας.

5. Γράψτε πρόγραμμα που να δημιουργεί μια λίστα, η οποία θα περιέχει τους άρτιους αριθμούς από το 1 έως το 50. Στη συνέχεια το πρόγραμμα θα εκτυπώνει τη λίστα και το μήκος της.

**6.** Να γράψετε ένα πρόγραμμα το οποίο δημιουργεί μια λίστα η οποία περιλαμβάνει όλους τους διψήφιους άρτιους αριθμούς. Στη συνέχεια δημιουργεί μια δεύτερη λίστα η οποία περιλαμβάνει τους αριθμούς της πρώτης λίστας αλλά με αντίστροφη σειρά (δηλ. από τον μεγαλύτερο στον μικρότερο) .

**7.** Γράψτε ένα πρόγραμμα που δημιουργεί τη λίστα L με στοιχεία τους ακέραιους στο διάστημα [1001,1100] οι οποίοι είναι πολλαπλάσια του 3 ή του 7. Στη συνέχεια εμφανίζει τα στοιχεία της λίστας.

**8.** Να γράψετε ένα πρόγραμμα το οποίο θα δημιουργεί μια λίστα 30 τυχαίων ακεραίων αριθμών από το 1 ως το 30 και θα εμφανίζει:

α. Τη λίστα των αριθμών

β. Πόσοι από αυτούς είναι περιττοί και πόσοι άρτιοι.

**9.** Γράψτε ένα πρόγραμμα που ζητάει από τον χρήστη να εισάγει ακέραιους αριθμούς. Τις τιμές αυτές καταχωρεί σε μια λίστα με όνομα list1. Η εισαγωγή σταματά όταν ο χρήστης δώσει το 0. Στη συνέχεια:

α. με έναν βρόχο for εμφανίζονται ένα – ένα τα στοιχεία της λίστας

β. υπολογίζεται και εμφανίζεται ο μέσος όρος των στοιχείων της λίστας.

**10.** Συμπληρώστε την προηγούμενη άσκηση ώστε να εμφανίζει το μικρότερο στοιχείο της λίστας και τη θέση του σε αυτή με τη μορφή 1ο, 2ο, κλπ.

**11.** Γράψτε ένα πρόγραμμα που ζητάει από τον χρήστη να εισάγει 5 λέξεις. Τις λέξεις αυτές καταχωρεί σε μια λίστα με όνομα list1.

Στη συνέχεια ζητάει από τον χρήστη μια λέξη. Αν τη βρει στη λίστα εμφανίζει τη θέση της σε αυτή.

**12.** Συμπληρώστε το προηγούμενο πρόγραμμα ώστε να ζητάει συνέχεια από τον χρήστη μια λέξη και να εμφανίζει τη θέση της στη λίστα. Το πρόγραμμα τερματίζει όταν η λέξη δεν βρίσκεται στη list1.

**13.** Γράψτε ένα πρόγραμμα στο οποίο δίνεται η παρακάτω λίστα list1=["ena", "dyo", "tria", "tessera", "pente"]. Κατόπιν ζητείται από τον χρήστη ο αριθμός των στοιχείων που θα διαγραφούν από το τέλος της λίστας. Στη συνέχεια εμφανίζεται η νέα λίστα.

**14.** Συμπληρώστε την προηγούμενη άσκηση με έναν βρόχο while ώστε να εξασφαλίζεται ότι ο αριθμός που δίνουμε δεν ξεπερνά τον αριθμό των στοιχείων της λίστας γιατί τότε προκύπτει το λάθος IndexError: pop from empty list

**15.** Γράψτε ένα πρόγραμμα στο οποίο δίνεται η λίστα list1=["Mia", "Kyriaki", "proi"]. Στη συνέχεια από τη λίστα αυτή δημιουργείται η συμβολοσειρά «Mia Kyriaki proi».

**16.** Γράψτε ένα πρόγραμμα στο οποίο δίνεται η συμβολοσειρά word="Kyriaki". Στη συνέχεια δημιουργείται μια λίστα με στοιχεία τους χαρακτήρες της λέξης και τέλος εμφανίζεται η λίστα.

**17.** Γράψτε ένα πρόγραμμα στο οποίο δίνεται η συμβολοσειρά protasi="Mia Kyriaki proi". Στη συνέχεια δημιουργείται μια λίστα με στοιχεία τους πρώτους χαρακτήρες κάθε λέξης και τέλος εμφανίζεται η λίστα.

**18.** Γράψτε ένα πρόγραμμα στο οποίο δίνονται οι λίστες `list1=[1,2,3,4,5,6,7,8]` και `list2=[11,12,13,14]`. Στη συνέχεια το πρόγραμμα προσθέτει στο τέλος της λίστας `list2` τόσα μηδενικά όσα χρειάζονται για να φτάσει στο μήκος της `list1`.

Γενικεύστε τη λύση ώστε να καλύπτει και την περίπτωση που η `list1` έχει λιγότερα στοιχεία από την `list2`. Τότε τα μηδενικά προστίθενται στην `list1`.

**19.** Γράψτε ένα πρόγραμμα στο οποίο δίνονται οι λίστες `list1=[1,2,3,4,5,6,7,8]` και `list2=[11,12,13,14]`. Στη συνέχεια το πρόγραμμα αφαιρεί από το τέλος της λίστας `list1` τόσα στοιχεία όσα χρειάζονται για να φτάσει στο μήκος της `list2`.

Γενικεύστε τη λύση ώστε να καλύπτει και την περίπτωση που η `list1` έχει λιγότερα στοιχεία από την `list2`. Τότε τα στοιχεία αφαιρούνται από τη `list1`.

**20.** Να γράψετε μια συνάρτηση η οποία θα δέχεται τη λίστα `words= ["alfa", "bita", "kapa", "lamda"]` και θα επιστρέφει τη λέξη που σχηματίζεται από τα πρώτα γράμματα των λέξεων, δηλαδή το "abkl".

**21.** Να γράψετε ένα πρόγραμμα, το οποίο θα διαβάζει από το πληκτρολόγιο μια λίστα από λέξεις μέχρι να δοθεί η λέξη 'end'.

Στη συνέχεια θα εμφανίζει τη λέξη με τα περισσότερα γράμματα (υπόδειξη: Κατασκευάστε μια λίστα με το πλήθος των γραμμάτων κάθε λέξης).

Επίσης, θα εμφανίζει το πλήθος των λέξεων που τελειώνουν με το γράμμα 's'.

**22.** Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει τα ονόματα 15 αεροπορικών εταιρειών και τις αντίστοιχες εισπράξεις τους για

έναν μήνα. Να τυπώνει τα ονόματα των εταιρειών που έχουν εισπράξεις περισσότερες από τον μέσο όρο.

**23.** Να γράψετε ένα πρόγραμμα το οποίο να διαβάζει των αριθμό των τερμάτων που σημειώθηκαν σε 15 αγώνες ποδοσφαίρου μίας αγωνιστικής της Α' κατηγορίας και να υπολογίζει και να εμφανίζει:

- α. τον μέσο αριθμό τερμάτων.
- β. τα περισσότερα τέρματα.
- γ. τα λιγότερα τέρματα.
- δ. το εύρος των τερμάτων, δηλαδή τη διαφορά της μεγαλύτερης από την μικρότερη τιμή.

**24.** Να γράψετε ένα πρόγραμμα που θα διαβάζει 50 ηλικίες παιδιών που φιλοξενούνται σε μια κατασκήνωση. Οι ηλικίες αυτές πρέπει να είναι μεγαλύτερες ή ίσες με το 5 και μικρότερες ή ίσες με το 17. Το πρόγραμμα αφού κάνει έλεγχο ηλικιών στην είσοδο, θα βρίσκει και θα εμφανίζει τον αριθμό των παιδιών που έχουν ηλικία πάνω από τον μέσο όρο καθώς επίσης τη μεγαλύτερη και τη μικρότερη ηλικία.

**25.** Γράψτε ένα πρόγραμμα που ορίζει τις παρακάτω λίστες ακεραίων:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

Στη συνέχεια δημιουργεί μια νέα λίστα c που περιέχει μόνο τα κοινά στοιχεία των δυο λιστών και στη συνέχεια τυπώνει τα στοιχεία της λίστας c. Συμπληρώστε το πρόγραμμα ώστε η λίστα c να μην περιέχει διπλές τιμές.

**26.** Η Α' τάξη ενός Λυκείου αποτελείται από 85 μαθητές. Να γράψετε ένα πρόγραμμα το οποίο διαβάζει το ονοματεπώνυμο,

το φύλο και τον γενικό βαθμό κάθε μαθητή και τα τοποθετεί σε τρεις λίστες, OnEr, Filo και Bathmos αντίστοιχα. Το φύλο μπορεί να είναι "Κ" αν είναι κορίτσι ή "Α" αν είναι αγόρι, ενώ ο βαθμός είναι στην κλίμακα από 0 μέχρι 20. Στη συνέχεια εμφανίζει:

- α. Πόσα αγόρια έχουν γενικό βαθμό πάνω από τη βάση (δηλαδή το 10).
- β. Τα ονοματεπώνυμα των κοριτσιών που έχουν γενικό βαθμό κάτω από το μέσο όρο των βαθμών όλης της τάξης.

**27.** Μία Νομαρχία διοργάνωσε το 2008 σεμινάριο εθελοντικής δασοφυτεύσεως, το οποίο παρακολούθησαν 500 άτομα. Η Πυροσβεστική Υπηρεσία ζήτησε στοιχεία σχετικά με την ηλικία, το φύλο και το μορφωτικό επίπεδο εκπαίδευσης κάθε εθελοντή, προκειμένου να βγάλει στατιστικά στοιχεία. Να γράψετε πρόγραμμα το οποίο:

- α. Διαβάζει για κάθε άτομο και καταχωρίζει σε λίστες:
  - το ονοματεπώνυμο.
  - το έτος γέννησης (χωρίς να απαιτείται έλεγχος εγκυρότητας).
  - το φύλο, με αποδεκτές τιμές το "Α" για τους άνδρες και το "Γ" για τις γυναίκες.
  - το μορφωτικό επίπεδο εκπαίδευσης, με αποδεκτές τιμές "Ρ", "Δ" ή "Τ", που αντιστοιχούν σε Πρωτοβάθμια, Δευτεροβάθμια ή Τριτοβάθμια Εκπαίδευση.
- β. Υπολογίζει και εμφανίζει το πλήθος των ατόμων με ηλικία μικρότερη των 30 ετών.
- γ. Υπολογίζει και εμφανίζει το ποσοστό των γυναικών με επίπεδο Τριτοβάθμιας Εκπαίδευσης στο σύνολο των εθελοντριών.

**28.** Για την πενθήμερη εκδρομή τους, οι μαθητές αποφάσισαν να συγκεντρώσουν τα απαραίτητα χρήματα. Να γράψετε πρόγραμμα το οποίο:

- α. Διαβάζει το πλήθος των μαθητών. Αυτό δεν θα πρέπει να είναι μεγαλύτερο από 80.
- β. Διαβάζει τα επώνυμα όλων των μαθητών και τα τοποθετεί σε μια λίστα.
- γ. Δέχεται τα χρήματα που συγκεντρώσε κάθε μαθητής και τα τοποθετεί σε μια άλλη λίστα.
- δ. Υπολογίζει το συνολικό ποσό που συγκεντρώθηκε και εμφανίζει ανάλογο μήνυμα.
- ε. Εμφανίζει τα επώνυμα των μαθητών που έδωσαν περισσότερα από 100€.



## 5. Κλασσικοί Αλγόριθμοι

### 5.1 Αλγόριθμος

Ο αλγόριθμος είναι μια ακολουθία πλήρως καθορισμένων και εκτελέσιμων βημάτων -εντολών, που οδηγεί στην επίλυση ενός προβλήματος, σε πεπερασμένο χρόνο.

#### Υπολογισμός Μέγιστου Κοινού Διαιρέτη

ΜΚΔ (100, 36)

$\alpha=100, \beta=36$

$u=100 \% 36=28$

ΒΗΜΑ 1

$\alpha \leftarrow \beta \quad \alpha=36$

$\beta \leftarrow u \quad \beta=28$

ΜΚΔ (36, 28)

$\alpha=36, \beta=28$

$u=36 \% 28=12$

ΒΗΜΑ 2

$\alpha \leftarrow \beta \quad \alpha=28$

$\beta \leftarrow u \quad \beta=12$

ΜΚΔ (28, 12)

$\alpha=28, \beta=12$

$u=28 \% 12=4$

ΒΗΜΑ 3

$\alpha \leftarrow \beta \quad \alpha=12$

$\beta \leftarrow u \quad \beta=4$

ΜΚΔ (12, 4)

$\alpha=12, \beta=4$

$u=12 \% 4=0$

ΒΗΜΑ 4

$\alpha \leftarrow \beta \quad \alpha=4$

$\beta \leftarrow u \quad \beta=0$

**ΜΚΔ =4**

### Αλγόριθμος ΜΚΔ

Δεδομένα // α, β //  
 Όσο β > 0 Επανάλαβε  
 υ ← α % β  
 α ← β  
 β ← υ  
 Τέλος\_Επανάληψης  
 Αποτέλεσμα ΜΚΔ=α  
 Τέλος ΜΚΔ

### Κώδικας PYTHON

```
def mkd(a, b):
 while b > 0 :
 y=a % b
 a=b
 b=y
 return a
print mkd(100,36)
```

## 5.2 Σειριακή Αναζήτηση

Πολλές φορές χρειάζεται να αναζητήσουμε κάτι συγκεκριμένο μέσα σε μια συλλογή δεδομένων. Η αναζήτηση γίνεται συνήθως με βάση κάποιο χαρακτηριστικό των δεδομένων, όπως για παράδειγμα το επώνυμο, ο αριθμός ταυτότητας κ.λπ.

Η πιο απλή μορφή αναζήτησης είναι η σειριακή ή γραμμική αναζήτηση σύμφωνα με την οποία εκτελούμε εξαντλητική αναζήτηση σε όλα τα στοιχεία της συλλογής, μέχρι να βρούμε αυτό που ψάχνουμε.

Έτσι, στη χειρότερη περίπτωση, αν ξεκινήσουμε από το πρώτο στοιχείο και αυτό που ψάχνουμε είναι τελευταίο, θα κάνουμε τόσες συγκρίσεις, όσα είναι και τα στοιχεία που έχουμε.

### # linearSearch1

```
def linearSearch(array, key):
 found = False
 for item in array :
 if item == key :
 found = True
 return found
print linearSearch([1,6,2,8], 5)
```

Η έκδοση αυτή του αλγόριθμου επιστρέφει μόνο την πληροφορία αν το στοιχείο υπάρχει στη συλλογή ή όχι. Ωστόσο, πολλές φορές θέλουμε να βρούμε και τη θέση του στοιχείου στη συλλογή, ειδικά αν αυτή υλοποιείται με κάποια δομή δεδομένων που επιτρέπει την αποθήκευση των στοιχείων με κάποια διάταξη.

Έτσι μπορούμε να προσθέσουμε και μία μεταβλητή θέσης. Τώρα όμως η λογική μεταβλητή είναι περιττή, αφού η μεταβλητή θέση μπορεί να παίξει και το ρόλο της λογικής μεταβλητής ως εξής: Αν η θέση παραμείνει  $-1$ , τότε σημαίνει ότι το στοιχείο δεν βρέθηκε. Αν όμως βρεθεί, τότε θα πάρει την τιμή της θέσης του δείκτη, η οποία θα είναι μη αρνητική.

Μια άλλη προσέγγιση του κώδικα στην Python

#### # linearSearch2

```
def linearSearch(array, key):
 pos=-1
 i=0
 while pos < 0 and i < len(array) :
 if array[i] == key :
 pos = i
 i = i + 1
 return pos
print "Thesi", linearSearch([1,6,2,8], 6)
```

#### # linearSearch3

```
def linearSearch(array, key):
 pos=-1
 for i in range(len(array)):
 if array[i] == key :
 pos = i
 return pos
print "Thesi", linearSearch([1,6,2,8], 6)
```

## 5.3 Ταξινόμηση

Το πρόβλημα της ταξινόμησης αναφέρεται στην αναδιάταξη των στοιχείων μιας λίστας, ώστε αυτά να βρεθούν σε αύξουσα ή σε φθίνουσα σειρά, σε σχέση με κάποια σχέση διάταξης.

Πολλές φορές χρειάζεται να ταξινομήσουμε λίστες αριθμών, λέξεων ή εγγράφων. Για παράδειγμα, μπορεί να θέλουμε να ταξινομήσουμε τις καρτέλες των μαθητών. Θα πρέπει σε αυτή την περίπτωση να επιλέξουμε ένα χαρακτηριστικό με βάση το οποίο θα γίνει η ταξινόμηση, όπως για παράδειγμα το όνομα, ο βαθμός ή ο αριθμός μητρώου.

Το χαρακτηριστικό αυτό λέγεται **κλειδί** και συνήθως (αλλά όχι πάντα) προσδιορίζει μονοσήμαντα την οντότητα στην οποία αναφέρεται.

### Ταξινόμηση με Επιλογή

Ένας από τους πιο απλούς αλγόριθμους ταξινόμησης είναι ο αλγόριθμος ταξινόμησης με επιλογή (selection sort).

Η λογική του αλγόριθμου είναι η παρακάτω:

**Βήμα 1.** Βρίσκουμε το μικρότερο στοιχείο του πίνακα και το τοποθετούμε στην πρώτη θέση.

**Βήμα 2.** Βρίσκουμε το αμέσως μικρότερο στοιχείο του πίνακα και το τοποθετούμε στη δεύτερη θέση.

**Βήμα 3.** Συνεχίζουμε βρίσκοντας κάθε φορά το μικρότερο στοιχείο από τα στοιχεία του πίνακα που απομένουν και το τοποθετούμε στο τέλος των ήδη ταξινομημένων στοιχείων.

```
def selectionSort(array) :
 N = len(array)
 for i in range(N):
 pos = posMin(i, N, array)
 array[i], array[pos]=array[pos],array[i]
 return array

def posMin(start, size, array):
 pos = start
 for i in range(start, size):
 if array[i] < array[pos]:
 pos = i
 return pos
print selectionSort([60, 38, 98, 54, 32, 90, 20])
```

## 5.4 Ασκήσεις

1. Να τροποποιήσετε τον αλγόριθμο της ταξινόμησης με επιλογή, ώστε να ταξινομεί έναν πίνακα ακεραίων σε φθίνουσα σειρά.
2. Να τροποποιήσετε τον αλγόριθμο της σειριακής αναζήτησης, ώστε να υπολογίζει πόσες φορές εμφανίζεται το ζητούμενο στοιχείο στον πίνακα.
3. Να τροποποιήσετε τον αλγόριθμο της σειριακής αναζήτησης, ώστε να εμφανίζει στην οθόνη τις θέσεις που βρίσκονται τα πολλαπλά στοιχεία.
4. Να τροποποιήσετε τον αλγόριθμο της ταξινόμησης με επιλογή, ώστε σε κάθε βήμα του να βρίσκει το μικρότερο και το μεγαλύτερο, να τοποθετεί το μικρότερο στην αρχή του πίνακα και το μεγαλύτερο στο τέλος.

## 6. Επανάληψη

### 6.1 Ασκήσεις

1. Γράψτε μια συνάρτηση που έχει παραμέτρους δυο ακεραίους. Αν αυτοί είναι άρτιοι, υπολογίζει και επιστρέφει το άθροισμά τους. Αν τουλάχιστον ο ένας από τους δυο είναι περιττός, υπολογίζει και επιστρέφει τη διαφορά τους.
2. Γράψτε μια συνάρτηση που επιστρέφει τον μεγαλύτερο από δυο αριθμούς.
3. Γράψτε ένα πρόγραμμα που ζητάει συνέχεια δυο αριθμούς μέχρι να δώσουμε το 0 και χρησιμοποιώντας τη συνάρτηση της προηγούμενης άσκησης τυπώνει τον μεγαλύτερο.
4. Γράψτε μια συνάρτηση που έχει παράμετρο μια λέξη. Η συνάρτηση επιστρέφει την αντίστροφη της λέξης.
5. Να γραφεί συνάρτηση η οποία δέχεται έναν αριθμό και ελέγχει αν ο αυτός είναι θετικός διψήφιος. Το αποτέλεσμα του ελέγχου θα είναι μια λογική τιμή.
6. Γράψτε μια συνάρτηση που επιστρέφει τον αριθμό των λέξεων που περιέχει μια πρόταση. Θεωρήστε ότι οι λέξεις χωρίζονται με τον χαρακτήρα κενό.
7. Γράψτε μια συνάρτηση που δημιουργεί μια λίστα που έχει σαν στοιχεία το άθροισμα των στοιχείων δυο λιστών που περιέχουν τον ίδιο αριθμό ακεραίων αριθμών.

8. Γράψτε μια συνάρτηση που επιστρέφει τον μέσο όρο των στοιχείων μιας λίστας.
9. Γράψτε μια συνάρτηση που εμφανίζει έναν τυχαίο αριθμό που βρίσκεται ανάμεσα σε δυο ακераίους που περνούν σαν παράμετροι της συνάρτησης.
10. Συμπληρώστε την προηγούμενη συνάρτηση με μια τρίτη παράμετρο που ορίζει πόσους τυχαίους αριθμούς θα εμφανίζει.
11. Τροποποιήστε την προηγούμενη συνάρτηση ώστε να δημιουργεί μια λίστα με στοιχεία αυτούς τους τυχαίους αριθμούς.
12. Να γράψετε μια συνάρτηση η οποία δέχεται μια λίστα από λέξεις και επιστρέφει τη λέξη που σχηματίζεται από τα πρώτα γράμματα των λέξεων.
13. Να γράψετε μια συνάρτηση που προσομοιώνει το ρίξιμο ενός ζαριού. Αν η ζαριά είναι άρτιος αριθμός, εμφανίζει το μήνυμα «ΚΕΡΔΙΣΕΣ». Αν είναι περιττός, εμφανίζει το μήνυμα «ΕΧΑΣΕΣ».
14. Συμπληρώστε την προηγούμενη συνάρτηση με μια παράμετρο που ορίζει πόσες φορές θα ρίξετε το ζάρι. Η συνάρτηση επιπλέον μετρά και εμφανίζει πόσες φορές χάσατε και πόσες κερδίσατε.
15. Το μπαρμπούτι προέρχεται από την τούρκικη λέξη *barbut*, που σημαίνει τυχερό παιχνίδι με ζάρια.  
**Κανόνες:** Νικούν οι εξάρες, οι πεντάρες, το έξι-πέντε και οι τριάρες. Χάνουν οι άσσοι, οι δυάρες (ή διπλές), τα ντόρτια (τεσσάρες) και το ασσόδυο.



Φτιάξτε μια συνάρτηση που προσομοιώνει τη ρίψη δυο ζαριών. Σύμφωνα με τους παραπάνω κανόνες, εμφανίζει αν χάσατε ή αν κερδίσατε. Σε διαφορετική περίπτωση εμφανίζει τη ζαριά σας.

**16.** Συμπληρώστε την προηγούμενη συνάρτηση ώστε αν ούτε χάσατε ούτε κερδίσατε, να εμφανίζει τη ζαριά σας και να ρωτάει αν θέλετε να ξαναρίξετε τα ζάρια. Τότε επαναλαμβάνει τη ρίψη. Σε περίπτωση που χάνετε ή κερδίζετε, το πρόγραμμα τερματίζεται.

**17.** Φτιάξτε ένα παιχνίδι που ο χρήστης καλείται να μαντέψει έναν τυχαίο ακέραιο αριθμό που ορίζει ο υπολογιστής ανάμεσα στο 1 (κάτω όριο) και στο 32 (άνω όριο). Ο χρήστης έχει στη διάθεσή του 4 προσπάθειες. Όταν ο χρήστης δίνει αριθμό μεγαλύτερο ή μικρότερο από τον ζητούμενο, ο υπολογιστής εμφανίζει κατάλληλο μήνυμα. Όταν δώσει τον σωστό αριθμό, εμφανίζεται σχετικό μήνυμα και διακόπτεται η ροή του προγράμματος.

**18.** Συμπληρώστε το προηγούμενο πρόγραμμα ώστε ο χρήστης να ορίζει από την αρχή το άνω και το κάτω όριο των τυχαίων αριθμών, καθώς και τον αριθμό των προσπαθειών.

**19.** Συμπληρώστε το προηγούμενο πρόγραμμα ώστε άσχετα αν βρίσκουμε τον αριθμό ή όχι, να ρωτάει αν θα ξαναρχίσει το παιχνίδι κρατώντας τα ίδια όρια, αλλά ορίζοντας νέο αριθμό προσπαθειών.

**20.** Συμπληρώστε το προηγούμενο πρόγραμμα αναθέτοντας σε μια συνάρτηση την είσοδο των δυο ορίων, του αριθμού προσπαθειών και της δημιουργίας του τυχαίου αριθμού.

**21.** Συμπληρώστε το προηγούμενο πρόγραμμα ώστε να εμφανίζει πόσες φορές ο χρήστης βρίσκει τον αριθμό και πόσες φορές χάνει.

**22.** Φτιάξτε ένα πρόγραμμα που προσομοιώνει τη ρίψη ενός νομίσματος. Χρησιμοποιήστε τη συνάρτηση `randint` με παραμέτρους 1,2. Αν επιστρέψει 1 θεωρήστε ότι αυτό αντιστοιχεί σε κορώνα, ενώ το 2 αντιστοιχεί σε γράμματα. Ο χρήστης καλείται να μαντέψει το αποτέλεσμα και εμφανίζεται σχετικό μήνυμα. Συμπληρώστε το πρόγραμμα ώστε να μετρήσει τις σωστές απαντήσεις, καθώς και πόσες φορές ήρθε κορώνα και πόσες γράμματα.

# Παράρτημα: Ενδεικτικές λύσεις των ασκήσεων

## Κεφάλαιο 2.3 - Δομή επιλογής

### #askisi 1

```
num=input("Dose enan akeraio:")
if num % 2 == 0:
 print "Artios"
else:
 print "Perittos"
```

### #askisi 2

```
num1=input("Dose ton proto arithmo:")
num2=input("Dose ton deftero arithmo:")
if num1 > num2:
 print "0", num1 ,"einai megaliteros apo ton", num2
else:
 print "0", num1 ,"einai mikroteros apo ton", num2
```

### #askisi 3

```
letter=raw_input("Dose ena kefalaio gramma - A, B, C, D, E ")
if letter == 'A':
 print "Alice"
elif letter == 'B':
 print "Beatrice"
elif letter == 'C':
 print "Chris"
elif letter == 'D':
 print "Dean"
```

```
else:
 print "Edward"
```

#### **#askisi 4**

```
letter=raw_input("Dose ena gramma:")
if letter == 'a' or letter == 'e' or letter == 'i' or letter == 'o' or letter
== 'u':
 print "fonien"
else:
 print "simfono"
```

#### **#askisi 5**

```
num1, num2, num3=input("Dose 3 akeraious:")
if num1 > num2:
 max=num1
else:
 max=num2
if num3 > max:
 max=num3
print "O megaliteros einai o", max
```

#### **#askisi 6**

```
num1, num2, num3=input("Dose 3 akeraious:")
if num1 < num2:
 min=num1
else:
 min=num2
if num3 < min:
 min=num3
print "O mikroteros einai o", min
```

### #askisi 7

```
etos=input("Dose to etos:")
if etos % 4==0 and (etos % 100 !=0 or etos % 400 ==0):
 print "Disekto"
else:
 print "Oxi disekto"
```

### #askisi 8

```
a, b, c =input("Dose tous sintelestes tis exisisis:")
d=b**2-4*a*c
if d<0:
 print "H exisosi exei rizes migadikes"
elif d==0:
 x=(-b+d*1/2)/2*a
 print "H exisosi exei mia riza dipli:",x
else:
 x1=(-b+d*1/2)/2*a
 x2=(-b-d*1/2)/2*a
 print x1, x2
```

### #askisi 9

```
number1, number2, number3=input("Dwse 3 akeraious:")
if number1==number2 or number1==number3 or
number2==number3:
 print "Yparxoun idioi arithmoi"
else:
 print "Oloi oi arithmoi einai diaforetikoi"
```

#### #askisi 10

```
number1, number2, number3=input("Dwse 3 akeraious:")
if number1==number2:
 print "Υparxoun idioi arithmoi",number1, number2
elif number1==number3:
 print "Υparxoun idioi arithmoi",number1, number3
elif number2==number3:
 print "Υparxoun idioi arithmoi",number2, number3
else:
 print "Oloi oi arithmoi einai diaforetikoi"
```

#### #askisi 11

```
num1, num2, num3, num4=input("Dose 4 akeraious:")
max=num1
if num2 > max:
 max=num2
if num3 > max:
 max=num3
if num4 > max:
 max=num4
print "O megaliteros einai o", max
```

#### #askisi 12

```
num1, num2=input("Dose 2 akeraious:")
if (num1 % 2 ==0 and num2 % 2 ==0) or (num1 % 2 !=0 and num2
% 2 !=0):
 mo=(num1+num2)/2
 print "o mesos oros ton arithmon einai:", mo
else:
 diafora= abs(num1-num2)
```

```
print "H apoliti timi tis diaforas ton dio ariumon einai", diafora
#askisi 13
num1, num2=input("Dose 2 akeraious:")
if num1*num2>0:
 piliko=float(num1)/num2
 print "To piliko ton dio arithmon einai:", piliko
else:
 diafora=num1 - num2
 print "H diafora ton dio ariumon einai:", diafora
```

#### **#askisi 14**

```
Kouponia=int(input("Dose kouponia: "))
if Kouponia<500:
 print("Kamia ekptosh")
elif Kouponia<1001:
 print("5 euro ekptosh")
elif Kouponia<2001:
 print("18 euro ekptosh")
else:
 print("54 euro ekptosh")
```

#### **#askisi 15**

```
xronos=input("Dose ta lepta omilias:")
sms=input("Dose ton arithmo sms:")
megabyte=raw_input("Xreistikes epileon megabyte (nai/oxi):")

if megabyte=="nai":
 axia_megabyte=5
else:
 axia_megabyte=0
```

```
pagio=22
axia_sms=sms*0.12
fpa=24./100
if xronos<=200:
 xreosi=pagio+axia_sms+axia_megabyte
else:
 xreosi=pagio+axia_sms+axia_megabyte+(xronos-200)*0.045

teliki_xreosi=xreosi+xreosi*fpa
print teliki_xreosi
```



## Κεφάλαιο 2.4 - Δομή for

### #askisi1

```
for i in range(1,11):
 print i
```

### #askisi2

```
for i in range(10,0,-1):
 print i
```

### #askisi3

```
ilikia=input("Dose tin ilikia sou:")
for i in range(ilikia+1):
 if i%2==0:
 print i
```

### #askisi4

```
for i in range(51):
 if i%8==0:
 print i
```

### #askisi5

```
number=input("Dose enan akeraio:")
for i in range(number,number+1):
 for j in range(1,11):
 print i,"*",j,"=", i*j
 print
```

### **#askisi6**

```
for i in range(50,-11,-1):
 if i%2==0:
 print i
```

### **#askisi7**

```
sum=0
plithos_artion=0
for i in range(10):
 number=input("Dose akeraio:")
 if number %2 == 0:
 sum+=number
 plithos_artion+=1
print sum/plithos_artion
```

### **#askisi8**

```
import random
plithos_monopsiphion=0
plithos_dipsiphion=0
plithos_tripsiphion=0

for i in range(50):
 number=random.randint(1,1000)
 if number <10:
 plithos_monopsiphion+=1
 elif number <100:
 plithos_dipsiphion+=1
 else:
 plithos_tripsiphion+=1
```

```
pososto_monopsiphion=(plithos_monopsiphion/50.0)*100
pososto_dipsiphion=(plithos_dipsiphion/50.0)*100
pososto_tripsiphion=(plithos_tripsiphion/50.0)*100

print "pososto monopsiphion:",pososto_monopsiphion, "%"
print "pososto dipsiphion:",pososto_dipsiphion, "%"
print "pososto tripsiphion:",pososto_tripsiphion, "%"
```

#### **#askisi9**

```
s=0
count=0.
for i in range(10,100):
 s+=i
 count+=1
mo=s/count
print "To athrisma einai:",s
print "O mesos oros einai:", mo
```

#### **#askisi10**

```
megaliteroi_15=0
plithos_artion=0
for i in range(20):
 number=input("Dose akeraio:")
 if number % 2 == 0:
 plithos_artion+=1
 if number > 15:
 megaliteroi_15+=1
print "Plithos artion:",plithos_artion
print "Arithmoi megaliteroi tou 15:",megaliteroi_15
```

### #askisi11

```
count=0
for i in range(100,1000):
 if i % 7 == 0:
 count+=1
 print i

print "ta tripsifia p ollaplasia tou 7 einai:",count
```

### #askisi12

```
s=0
for i in range(1,10):
 if i % 2 == 0:
 s+=i**2
print s
```

### #askisi13

```
count=0
for i in range(14):
 epidosi=input("Dose tin epidosi tou athliti:")
 if epidosi>80:
 count+=1
print count
```

### #askisi14

```
for i in range(1,5):
 for j in range(i):
 print '*',
 print
```

**#askisi15**

```
for i in range(5):
 for j in range(i):
 print '*',
 print
```

```
for i in range(5,0,-1):
 for j in range(i):
 print '*',
 print
```

## Κεφάλαιο 2.5 - Δομή while

### #askisi 1

```
for i in range(1,21):
 print i
print
```

```
i=1
while i <=20:
 print i
 i+=1
```

### #askisi 2

```
number=input("Dose arithmo:")
plithos=0
artioi=0
perittoi=0
sum_perittoi=0
```

```
while number !=0:
 number=input("Dose arithmo:")
 plithos+=1
 if number %2 ==0 and number !=0:
 artioi+=1
 elif number %2 !=0 and number !=0:
 perittoi+=1
 sum_perittoi+=number
```

```
mo_perittoi=sum_perittoi/float(perittoi)
print "Plithos arithmon:", plithos-1
```

```
print "Artioi:", artioi
print "Perittoi:", perittoi
print "Mesos oros peritton:", mo_perittoi
```

### #askisi 3

```
i=1
b1=0
b2=0
sum=0
bathmos_max=0
bathmos_min=20

while i <=15:
 bathmos=input("Dose ton bathmo:")
 sum+=bathmos
 if bathmos > bathmos_max:
 bathmos_max=bathmos
 if bathmos < bathmos_min:
 bathmos_min=bathmos
 if bathmos>16:
 b1+=1
 if bathmos<=20 and bathmos >=18:
 b2+=1
 i+=1

print "Mesos oros tetraminou:", sum/15
print "Megaliteros bathmos:", bathmos_max
print "Mikroteros bathmos:", bathmos_min
print "Bathmoi megarliteroi tou 16:", b1
print "pososto bathmon apo 18 -20:", b2/15.*100
```

#### #askisi 4

```
athroisma=0
i=0

while athroisma <=2000:
 i+=1
 athroisma+=i

print "To athroisma ginetai megalitewro tou 2000 ston",i, "o oro"
```

#### #askisi 5

```
i=0
number=input("Dose ton arithmo:")
while number >0:
 print number**2
 number=input("Dose ton arithmo:")
 i+=1

print "Diabastikan", i, "arithmoi"
```

#### #askisi 6

```
s=0
bathmos_max=0
onoma_mathiti=" "
arithmos_mathiton1=0
arithmos_mathiton2=0
bathmos=0

onoma=raw_input("Dose to onoma tou mathiti:")
while onoma!="$":
```



```
bathmos=input("Dose to bathmo tou mathiti:")
arithmos_mathiton1=arithmos_mathiton1+1
s=s+bathmos
if bathmos> bathmos_max:
 bathmos_max=bathmos
 onoma_mathiti=onoma
if bathmos<10:
 arithmos_mathiton2=arithmos_mathiton2+1
onoma=raw_input("Dose to onoma tou mathiti:")
mesos_oros=1.0*s/arithmos_mathiton1
pososto=1.0*arithmos_mathiton2/arithmos_mathiton1*100

print "O mesos oros tis bathmologias einai:",mesos_oros
print "O megaliteros bathmos einai:",bathmos_max
print "Kai ton pire o mathitis:",onoma_mathiti
print "Pososto mathiton me bathmo kato apo 10:",pososto
```

### **#askisi 7**

```
paragontiko=1
number=input("Dose thetiko akeraio arithmo:")
if number==0:
 paragontiko=1
else:
 for i in range(1,number+1):
 paragontiko*=i

print "To paragontiko einai:", paragontiko
```

### #askisi 8

```
melisses=1200
eti=0
while melisses <=2000:
 melisses=melisses+melisses*3.8/100
 eti+=1
print "Xreiazontai:",eti, "eti"
```

### #askisi 9

```
melisses=input("Dose ton arxiko arithmo melisson:")
melisses_max=input("Dose ton megisto arithmo melisson:")
eti=0
while melisses <=melisses_max:
 melisses=melisses+melisses*3.8/100
 eti+=1
print "Xreiazontai:",eti, "eti"
```

### #askisi 10

```
athroisma=0
i=0
while athroisma<=150 and i<=10:
 number=input("Dose arithmo:")
 i+=1
 athroisma+=number
print "Diabastikan", i, "arithmoi"
print "To athroisma tous einai:",athroisma
```

**#askisi 11**

```
n1, n2 = 0, 1
count=1
while count <=10:
 print n1
 n3 = n1 + n2
 n1 = n2
 n2 = n3
 count += 1
```

## Κεφάλαιο 3.3 - Συναρτήσεις

### #askisi5

```
def phonienta(letter):
 if letter == 'a' or letter == 'e' or letter == 'i' or letter == 'o' or letter ==
 'u':
 print 'To', letter, 'einai phonien'
 else:
 print 'To', letter, 'den einai phonien'

letter=raw_input("Dose enan charaktira:")
phonienta(letter)
```

### #askisi6

```
def phonienta(letter,met1):
 if letter == 'a' or letter == 'e' or letter == 'i' or letter == 'o' or letter ==
 'u':
 print 'To', letter, 'einai phonien'
 met1+=1
 else:
 print 'To', letter, 'den einai phonien'
 return met1

met1=0
met2=0
letter=raw_input("Dose enan charaktira:")
while letter!= '.':
 met2+=1
 met1=phonienta(letter,met1)
 letter=raw_input("Dose enan charaktira:")
```

```
print 'Sinolikos arithmos phonienton:', met1
print 'Pososto phonienton:',float(met1)/met2*100
```

#### **#askisi7a**

```
import random
def zari():
 x=random.randint(1,6)
 print "efera",x

zari()
```

#### **#askisi7b**

```
import random
def zari():
 x=random.randint(1,6)
 print "efera",x

zaries=input("Poses fores tha rikso to zari?:")
for i in range():
 zari(zaries)
```

#### **#askisi7c**

```
import random
def zari():
 x=random.randint(1,6)
 print "efera",x

apantisi=raw_input(" tha rikso to zari?:")
while apantisi=="Y":
 zari()
 apantisi=raw_input(" tha rikso to zari?:")
```

### #askisi8

```
def forologia(kivismos):
 if kivismos>0 and kivismos<1000:
 foros=110
 elif kivismos<1399:
 foros=150
 elif kivismos <1999:
 foros=225
 else:
 foros=600
 return foros

sinolo_forou=0
for i in range(10):
 kivika=input("Dose ta kivika tou autikinitou:")
 poso=forologia(kivika)
 sinolo_forou+=poso
 print "Gia to autokinito afto o foros einai:",poso

print 'Forologia gia ola ta autokinita:', sinolo_forou
```

### #askisi9

```
def misthodosia(ores, oromisthio):
 ak_misthos=ores*oromisthio
 foros=ak_misthos*20/100
 k_asfalisi=ak_misthos*6/100
 return ak_misthos,foros, k_asfalisi
```

### #askisi10

```
def misthodosia(ores, oromisthio):
 ak_misthos=ores*oromisthio
 foros=ak_misthos*20/100
```

```
k_asfalisi=ak_misthos*6/100
return ak_misthos,foros, k_asfalisi
```

```
for i in range(20):
 onoma=raw_input("Dose to onoma toy ypallilou:")
 ores=input("Dose tis ores poy douleypse:")
 oromisthio=input("Dose to oromisthio:")
 x,y,z= misthodosia(ores,oromisthio)
 katharos_misthos=x-y-z

 print "Onoma ypallilou:", onoma
 print "Akatharistos misthos ypallilou:", x
 print "Foros eisodimatos ypallilou:", y
 print "Kratiseis asfalisis ypallilou:", z
 print "Kathares apodohes ypallilou:", katharos_misthos
```

### **#askisi11**

```
def amoivi(ores):
 if ores>0 and ores<41:
 misthos=ores*5
 elif ores <51:
 misthos=40*5+(ores-40)*7
 else:
 misthos=40*5+ 10*7+(ores-50)*9
 return misthos

kostos=0
met=0
for i in range(48):
 onoma=raw_input("Dose to onoma toy ypallilou:")
 ores=input("Dose tis ores poy douleypse:")
```

```
while ores<=0:
 ores=input("Dose thetiko arithmo:")
```

```
x= amoivi(ores)
kostos+=x
if x>=700:
 met+=1
print "Onoma ypallilou:", onoma
print "Amoivi ypallilou:", x
```

```
pososto=met/48.0*100
print "Pososto ypallilon me mistho >700:", pososto
```



## Κεφάλαιο 4.3 - Συμβολοσειρές

### #askisi1

```
word=raw_input("Dose ena string:")
char=raw_input("Dose ton xaraktira pou tha anazitiso:")
count=0
```

```
for i in range(len(word)):
 if word[i]==char:
 count+=1
```

```
print "O xaraktiras ", char, "brethike ", count, " fores"
```

### #askisi2

```
word=raw_input("Dose ena string:")
word2=""
```

```
for char in word:
 if char!=" ":
 word2+=char
```

```
print "to neo strin xoris kena einai:", word2
```

### #askisi3

```
word=raw_input("Dose ena string:")
word2=""
for i in range(len(word)-1,-1,-1):
 word2+=word[i]
```

```
print "to antistrofo string einai:", word2
```

#### #askisi4

```
def fun1(sentence):
 word=0

 for char in sentence:
 if char==" "
 word+=1
 print "to plithos ton lexeon einai:", word+1

protasi=raw_input("Dose mia protasi:")
fun1(protasi)
```

#### #askisi5

```
def fun1(sentence):
 word=0
 new_sentence=""
 for i in range(10):
 if sentence[i]!=" "
 new_sentence+=sentence[i]
 return new_sentence

protasi=raw_input("Dose mia protasi:")
print fun1(protasi)
```

#### #askisi6

```
def NewString(s):
 N = len(s)
 new = s[0].upper()
 for i in range(0,N):
 if s[i-1] == " "
 new += s[i].upper()
 else:
```

```
 new += s[i]
return new
```

```
protasi=raw_input("Dose mia protasi:")
print NewString(protasi)
```

### **#askisi7**

```
def email(s):
 t=len(s)

 if ('@' in s) and ("." ==s[t-3])and ("g" ==s[t-2]) and ("r" ==s[t-1]):
 print "Egkiro email"
 else:
 print "Lathos email"
```

```
protasi=raw_input("Dose ena email:")
email(protasi)
```

### **#askisi8**

```
def phonienta(word):
 phonien = "aeoiu"
 count = 0
 for letter in word :
 if letter in phonien:
 count += 1
 return count
```

```
protasi=raw_input("Dose mia protasi:")
print "Ta phonienta tis protasis einai:",phonienta(protasi)
```

### #askisi9

```
def antistrofi(word):
 word2=""
 n=len(word)
 for i in range(n-1,-1,-1):
 word2+=word[i]
 if word==word2:
 return "karkiniki"
 else:
 return "Oxi karkiniki"

word=raw_input("Dose mia lexi:")
print antistrofi(word)
```

## Κεφάλαιο 4.5 - Λίστες

### #askisi1

```
list1=[]
sum=0.
Count=0
while sum<=1000:
 number=input("dose enan arithmo:")
 count+=1
 list1.append(number)
 sum+=number
m_oros=sum/count

print "To athroisma einai:", sum
print "O meoso oros einai:", m_oros

maximum=list1[0]
thesi=0
for i in range(len(list1)):
 if list1[i]>maximum:
 maximum=list1[i]
 thesi=i+1
print "To megisto stoixeio tis listas einai to:",maximum
print "kai brisketai sth thesi", thesi
```

### #askisi2

```
list1=[]
number=input("dose enan arithmo:")
while number!=0:
```

```
list1.append(number)
number=input("dose enan arithmo:")
print list1
for i in range(len(list1)-1,-1,-1):
 print list1[i],
```

#### **#askisi3**

```
list1=[]
list2=[]
list3=[]
for i in range(5):
 number=input("dose enan arithmo:")
 list1.append(number)

for j in list1:
 if j > 0:
 list2.append(j)
 else:
 list3.append(j)

print list2
print list3
```

#### **#askisi4**

```
list1=[4,5,8,12,4]
list2=[-4,9,11,6,7]
list3=[]
x=len(list1)
for i in range(x):
 list3.append(list1[i]+list2[i])
print list3
```

**#askisi5**

```
list1=[]
for i in range(51):
 if i%2==0:
 list1.append(i)

print list1
print len(list1)
```

**#askisi6**

```
list1=[]
list2=[]
for i in range(9,99):
 if i%2==0:
 list1.append(i)

n=len(list1)
for i in range(n-1,0,-1):
 list2.append(list1[i])
print list2
```

**#askisi7**

```
L = []
for i in range(1001,1101):
 if i%3 == 0 or i%7 == 0:
 L.append(i)

print L
```

**#askisi8**

```
import random
list1=[]
count1=count2=0
for i in range(30):
 num=random.randint(1,30)
 list1.append(num)
 if num%2==0:
 count1+=1
 else:
 count2+=1

print list1
print "Artioi", count1
print "Perittooi", count2
```

**#askisi9**

```
list1=[]
sum=0.
Count=0
number=input("Dose akeraio h 0 gia telos:")
while number!=0:
 list1.append(number)
 sum+=number
 count+=1
 number=input("Dose akeraio h 0 gia telos:")

for number in list1:
 print number,

mo=sum/count
print "mesos oros:",mo
```



### #askisi10

```
list1=[]
sum=0.
Number=input("Dose akeraio h 0 gia telos:")
while number!=0:
 list1.append(number)
 sum+=number
 number=input("Dose akeraio h 0 gia telos:")

for number in list1:
 print number,

mo=sum/len(list1)
print "mesos oros:",mo
maximum=list1[0]
a=len(list1)
for i in range(a):
 if list1[i] > maximum:
 maximum=list1[i]
 index=i

print "to megalitero stoixeio einai to ",maximum," kai einai to ",
index+1,"o"
```

### #askisi11

```
list1=[]
for i in range(5):
 word=raw_input("Dose mia lexi:")
 list1.append(word)

lexi=raw_input("Dose ti lexi 121enesis anazitiso:")
for i in range(5):
```

```
if list1[i]==lexi:
 print "Η lexi brethike sti thesi ", i
```

#### **#askisi12**

```
list1=[]
for i in range(5):
 words=raw_input("Dose mia lexi:")
 list1.append(words)

lexi=raw_input("Dose ti lexi pou anazitiso:")
while lexi in list1:
 i=0
 for i in range(5):
 if list1[i]==lexi:
 print "Η lexi brethike sti thesi ", i
 lexi=raw_input("Dose ti lexi 122enesis anazitiso:")
```

#### **#askisi13**

```
list1=["ena", "dyo", "tria", "122enesis122", "pente"]
num=input("Posa stoixeia tha diagrafoun?")
for i in range(num):
 list1.pop()

print list1
```

#### **#askisi14**

```
list1=["ena", "dyo", "tria", "122enesis122", "pente"]
x=len(list1)
num=input("Posa stoixeia tha diagrafoun?")
while num>x:
 print "Den mporoun na diagrafoun perissotera apo", x, "stoixeia"
 num=input("Posa stoixeia tha diagrafoun?")
```

```
for i in range(num):
 list1.pop()
```

```
print list1
```

#### **#askisi15**

```
protasi=""
list1=["Mia", "Kyriaki", "proi"]
for word in list1:
 protasi+=word+" "
print protasi
```

#### **#askisi16**

```
word="Kyriaki"
list1=[]
for char in word:
 list1.append(char)
print list1
```

#### **#askisi17**

```
protasi="Mia Kyriaki proi"
list1=[]
x=len(protasi)
list1.append(protasi[0])
for i in range(x):
 if protasi[i]==" ":
 list1.append(protasi[i+1])
print list1
```

**#askisi18a**

```
list1=[1,2,3,4,5,6,7,8]
list2=[11,12,13,14]
x1=len(list1)
x2=len(list2)
x=abs(x1-x2)
for i in range(x):
 if x1>x2:
 list2.append(0)
 else:
 list1.append(0)

print list1
print list2
```

**#askisi18b**

```
list1=[1,2,3,4,5,6,7,8]
list2=[11,12,13,14]
x1=len(list1)
x2=len(list2)
for i in range(x1-x2):
 list2.append(0)

for j in range(x2-x1):
 list1.append(0)

print list1
print list2
```

**#askisi19**

```
list1=[1,2,3,4,5,6,7,8]
list2=[11,12,13,14]
x1=len(list1)
x2=len(list2)
x=abs(x1-x2)
for i in range(x):
 if x1>x2:
 list1.pop()
 else:
 list2.pop()

print list1
print list2
```

**#askisi19b**

```
list1=[1,2,3,4,5,6,7,8]
list2=[11,12,13,14]
x1=len(list1)
x2=len(list2)

for i in range(x1-x2):
 list1.pop()
for j in range(x2-x1):
 list2.pop()

print list1
print list2
```

#### #askisi20

```
def func1(word_list):
 new_word=""
 for word in word_list:
 new_word+=word[0]
 return new_word

words=["alfa", "bita", "126enes", "lamda"]
print func1(words)
```

#### #askisi21

```
list1=[]
list2=[]
count=0
word=raw_input("Dose mia lexi:")
while word!="end":
 list1.append(word)
 list2.append(len(word))
 word=raw_input("Dose mia lexi:")

maximum=list2[0]
lexi=list1[0]
for i in range(1,len(list2)):
 if list2[i]>maximum:
 maximum=list2[i]
 lexi=list1[i]
for word in list1:
 if word[-1]=='s':
 count+=1

print "H megalkiteri lexi einai:", lexi
```

```
print "kai exei arithmo grammaton:",maximum
print "yparchoyn", count,"lexeis poy teleionoun se s"
```

#### **#askisi22**

```
list1=[]
list2=[]
sum_eispraxi=0.
For i in range(15):
 etaireia=raw_input("Dose to onoma tis etaireias:")
 eispraxi=input("Dose tin miniea eispraxi:")
 list1.append(etaireia)
 list2.append(eispraxi)
 sum_eispraxi+=eispraxi

m_oros_eispraxi=sum_eispraxi/15

for i in range(15):
 if list2[i] > m_oros_eispraxi:
 print "Etaireia", list1[i], "eispraxi", list2[i]
```

#### **#askisi23**

```
list1=[]
sum_termata=0
for i in range(3):
 termata=input("Dose ton arithmo termaton:")
 list1.append(termata)
 sum_termata+=termata

m_oros_termata=sum_termata/15
```

```
max_termata=list1[0]
min_termata=list1[0]
for i in range(15):
 if list1[i]>max_termata:
 max_termata=list1[i]
 if list1[i]<min_termata:
 min_termata=list1[i]

evros_termaton=max_termata-min_termata
print "Ta perissotera termata:", max_termata
print "Ta ligotera termata:",min_termata
print "To eyros termatovn:",evros_termaton
```

#### #askisi24

```
list1=[]
sum_ilikies=0
count=0
i=0
while i<50:
 ilikia=input("Dose tin ilikia tou paidiou:")
 if ilikia >=5 and ilikia <=17:
 i+=1
 list1.append(ilikia)
 sum_ilikies+=ilikia

m_oros_ilikia=sum_ilikies/50
print list1

max_ilikia=list1[0]
min_ilikia=list1[0]
```



```
for i in range(50):
 if list1[i]>m_oros_ilikia:
 count+=1
 if list1[i]>max_ilikia:
 max_ilikia=list1[i]
 if list1[i]<min_ilikia:
 min_ilikia=list1[i]

print "Arithmos paidion me ilikia pano apo ton meso oro:", count
print "H megaliteri ilikia:",max_ilikia
print "H mikroteri ilikia:",min_ilikia
```

#### #askisi25

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
c=[]
for num1 in a:
 for num2 in b:
 if num1==num2:
 c.append(num1)

for i in range(lenl-2):
 if c[i]==c[i+1]:
 c.pop(i)

print c
```

### #askisi26

```
OnEp=[]
Filo=[]
Bathmos=[]
sum1=0
count=0
for i in range(5):
 onomateponimo=raw_input("Dose to onomateponymo tou mathiti:")
 filo=raw_input("Dose to filo:")
 bathmos=input("Dose ton bathmo:")
 OnEp.append(onomateponimo)
 Filo.append(filo)
 Bathmos.append(bathmos)
 if filo=="A" and bathmos>=10:
 count+=1
 sum1+=bathmos

print count, "Agoria exoun geniko bathmo pano apo ti basi"

m_oros=sum1/5.
For i in range(5):
 if Filo[i]=="K" and Bathmos[i]< m_oros:
 print "Kato apo ton meso oro exei h ",OnEp[i]
```

### #askisi27

```
OnEp=[]
EtosGennisis=[]
Filo=[]
Mepipedo=[]
TREXON_ETOS=2020
```

```

count=0
count2=0
for i in range(5):
 onomateponimo=raw_input("Dose to onomateponymo:")
 OnEp.append(onomateponimo)
 etos=input("Dose to etos 131enesis:")
 ilikia=TREXON_ETOS-etos
 if ilikia < 30:
 count+=1
 EtosGennisis.append(etos)
 filo=raw_input("Dose to filo:")
 while filo not in["A", "G"]:
 print "Dose sosta to filo (A h G):"
 filo=raw_input("Dose to filo:")
 Filo.append(filo)
 mepipedo=raw_input("Dose to morfotiko epipedo (P, D, T) :")
 while mepipedo not in["P", "D", "T"]:
 print "Dose sosta to morfotiko epipedo (P, D, T) :":
 mepipedo=raw_input("Dose to morfotiko epipedo (P, D, T) :")
 Mepipedo.append(mepipedo)

print "atoma me ilikia mikroteri apo 30 eti:", count

for i in range(5):
 if Filo[i]=="G" and Mepipedo[i]=="T":
 count2+=1

pososto=count2/5.0*100
print "To pososto ton ginaikon me epipedo tritobathmias ekpaideysis
einai:", pososto

```

### #askisi28

```
i=0
epwnuma=[]
xrhmata=[]
plithos=input("Dwse to plithos tw n mathitwn:")
while plithos>80:
 print "To plithos prepei na einai mikrotero apo 80!"
 plithos=input("Dwse to plithos tw n mathitwn:")

for i in range(plithos):
 ep=raw_input("Dwse to epwnumo tou mathiti:")
 xr=input("Dwse ta xrhmata tou mathiti:")
 epwnuma.append(ep)
 xrhmata.append(xr)

sum_xrhmata=0
for x in xrhmata:
 sum_xrhmata+=x

print "Sugkentrwthikan sunolika",sum_xrhmata,"euro"

for i in range(plithos):
 if xrhmata[i]>80:
 print "O mathitis",epwnuma[i],"edwse parapanw apo 80 euro"
```

## Κεφάλαιο 5.1 - Ασκήσεις επανάληψης

### #askisi1

```
def func1(x,y):
 if x%2==0 and y%2==0:
 return x+y
 elif x%2!=0 or y%2!=0:
 return x-y

print func1(2,4)
print func1(5,2)
print func1(9,5)
```

### #askisi2

```
def func1(x,y):
 if x > y:
 return x
 else:
 return y

maximum = func1(2,4)
print "Ο megaliteros arithmos einai o ", maximum
maximum = func1(8,7)
print "Ο megaliteros arithmos einai o ", maximum
```

### #askisi3

```
def func1(x,y):
 if x > y:
 return x
 else:
 return y
```

```
num1, num2 = input("Dose 2 akeraious:")
while num1 !=0 and num2 != 0:
 maximum = func1(num1,num2)
 print "O megaliteros arithmos einai o ", maximum
 num1, num2 = input("Dose 2 akeraious:")
```

### #askisi4

```
def func1(x):
 y=""
 for i in range(len(x)-1,-1,-1):
 y+=x[i]
 return y
```

```
word= raw_input("Dose mia lexi:")
antistrofi=func1(word)
print "H antistrofi tis lexis einai ", antistrofi
```

### #askisi5

```
def func1(x):
 if x>=10 and x<100:
 return True
 else:
 return False
```

```
num= input("Dose enan arithmo:")
apotelesma=func1(num)
if apotelesma:
 print "O arithmos einai thetikos kai dipsiphios"
else:
 print "O arithmos den einai thetikos h dipsiphios"
```

#### **#askisi6**

```
def func1(protasi):
 count=1
 for i in range(len(protasi)):
 if protasi[i]==" ":
 count+=1
 return count
```

```
x=func1("aa bb r")
print x
```

#### **#askisi7**

```
def func1(list1, list2):
 list3=[]
 for i in range(len(list1)):
 list3.append(list1[i]+list2[i])
 return list3
```

```
list1=[5,8,9]
list2=[1,7,2]
list3=func1(list1, list2)
print list3
```

**#askisi8**

```
def m_oros(list1):
 athroisma=0.
 for stoixeio in list1:
 athroisma+=stoixeio
 mo=athroisma/len(list1)
 return mo

mesos_oros=m_oros([2,4,6])
print mesos_oros
```

**#askisi9**

```
import random
def tyxaioi_arithmoi(num1, num2):
 number=random.randint(num1, num2)
 print number

tyxaioi_arithmoi(5,25)
```

**#askisi10**

```
import random
def tyxaioi_arithmoi(num1, num2,num3):
 for i in range(num3):
 number=random.randint(num1, num2)
 print number

tyxaioi_arithmoi(5,25,10)
```



**#askisi11**

```
import random
def tyxaioi_arithmoi(num1, num2,num3):
 list1=[]
 for i in range(num3):
 number=random.randint(num1, num2)
 list1.append(number)
 print list1

tyxaioi_arithmoi(5,25,10)
```

**#askisi12**

```
def func1(word_list):
 new_word=""
 for word in word_list:
 new_word+=word[0]
 return new_word

words=["alfa", "bita", "kapa", "lamda"]
print func1(words)
```

**#askisi13**

```
import random
def zari():
 die=random.randint(1,6)
 if die%2==0:
 print "Kerdises"
 else:
 print "Exases"

zari()
```

#### #askisi14

```
import random
def zari(num):
 count1=0
 count2=0
 for i in range(num):
 die=random.randint(1,6)
 if die%2==0:
 print "Kerdises"
 count1+=1
 else:
 print "Exases"
 count2+=1
 print "Kerdises ", count1, " fores"
 print "Exases ", count2, " fores"
zari(5)
```

#### #askisi15

```
import random
def barbuti():
 die1=random.randint(1,6)
 die2=random.randint(1,6)
 if (die1==6 and die2==6) or (die1==5 and die2==5) or (die1==3
and die2==3) or (die1==5 and die2==6) or (die1==6 and die2==5):
 print "Kerdises. Eferes ", die1, die2
 elif (die1==1 and die2==1) or (die1==2 and die2==2) or (die1==4
and die2==4) or (die1==1 and die2==2) or (die1==2 and die2==1):
 print "Exases. Eferes ", die1, die2
 else:
 print "Eferes", die1, die2
ap=barbuti()
```

### #askisi16

```
import random
def barbuti():
 flag=True
 while flag==True:
 die1=random.randint(1,6)
 die2=random.randint(1,6)
 if (die1==6 and die2==6) or (die1==5 and die2==5) or (die1==3
and die2==3) or (die1==5 and die2==6) or (die1==6 and die2==5):
 print "Kerdises. Eferes ", die1, die2
 flag=False
 elif (die1==1 and die2==1) or (die1==2 and die2==2) or
(die1==4 and die2==4) or (die1==1 and die2==2) or (die1==2 and
die2==1):
 print "Exases. Eferes ", die1, die2
 flag=False
 else:
 print "Eferes", die1, die2
 apantisi=raw_input("Tha xanapexeis (Y/N):")
 if apantisi=="Y":
 flag=True
 else:
 flag=False

ap=barbuti()
```

### #askisi17

```
import random
secret = random.randint(1,32)
flag = True
tries = 4
```

```
while flag and tries > 0:
 print"Apomenoun", tries, "prospatheies."
 number = input("Mantepse ton arithmo:")
 if number > secret:
 tries = tries - 1
 print"Lathos. Einai mikroteros."
 elif number < secret:
 tries = tries - 1
 print"Lathos. Einai megaliteros."
 else:
 print"Sosta!"
 flag = False

if flag==True:
 print"Htan o:",secret
```

### **#askisi18**

```
import random
low=input("Dose ton mikrotero arithmo:")
high=input("Dose ton megalitero arithmo:")
tries=input("Dose ton arithmo ton prospatheion:")
secret = random.randint(low,high)
flag = True
```

```
while flag and tries > 0:
 print"Apomenoun", tries, "prospatheies."
 number = input("Mantepse ton arithmo:")
 if number > secret:
 tries = tries - 1
 print"Lathos. Einai mikroteros."
 elif number < secret:
```

```
 tries = tries - 1
 print"Lathos. Einai megaliteros."
else:
 print"Sosta!"
 flag = False

if flag==True:
 print"Htan o:",secret

#askisi19

def func1():
 print "1. Eisagogi"
 print "2. Emfanisi"
 print "3. Anazitisi"
 print "4. Exodos"
 epilogi=input("Dose mia epilogi (1-4):")
 return epilogi

def func2(onoma,onomata):
 onomata.append(onoma)
 return onomata

def func3(onomata):
 for i in range(len(onomata)):
 print onomata[i]

def func4(onomata):
 return 0

onomata=[]
```

```
while True:
 epilogi=func1()
 if epilogi==1:
 onoma=raw_input("Dose to onoma tou mathiti:")
 func2(onoma,onomata)
 if epilogi==2:
 func3(onomata)
 if epilogi==3:
 func4(onomata)
 if epilogi==4:
 break
```

### **#askisi20**

```
import random
def dataFunction():
 low=input("Dose ton mikrotero arithmo:")
 high=input("Dose ton megalitero arithmo:")
 tries=input("Dose ton arithmo ton prospatheion:")
 secretNumber = random.randint(low,high)
 return secretNumber,tries
```

```
secret, tries= dataFunction()
flag1 = True
flag2 = True
```

```
while flag2 == True:
 while flag1 and tries > 0:
 print"Apomenoun", tries, "prospatheies."
 number = input("Mantepse ton arithmo:")
 if number > secret:
 tries = tries - 1
```

```
 print"Lathos. Einai mikroteros."
elif number < secret:
 tries = tries - 1
 print"Lathos. Einai megaliteros."
else:
 print"Sosta!"
 flag1 = False

if flag1 == True:
 print"Htan o:",secret

answer=raw_input("Tha kaneis alli prospatheia? (y/n)")
if answer == "y":
 flag2 = True
 flag1 = True
 secret, tries= dataFunction()
elif answer == "n":
 flag2 = False
```

### **#askisi21**

```
import random
def dataFunction():
 low=input("Dose ton mikrotero arithmo:")
 high=input("Dose ton megalitero arithmo:")
 tries=input("Dose ton arithmo ton prospatheion:")
 secretNumber = random.randint(low,high)
 return secretNumber,tries

secret, tries= dataFunction()
flag1 = True
flag2 = True
```

```
count1 = 0
count2 = 0
while flag2 == True:
 count2 += 1
 while flag1 and tries > 0:
 print"Apomenoun", tries, "prospatheies."
 number = input("Mantepse ton arithmo:")
 if number > secret:
 tries = tries - 1
 print"Lathos. Einai mikroteros."
 elif number < secret:
 tries = tries - 1
 print"Lathos. Einai megaliteros."
 else:
 print"Sosta!"
 flag1 = False
 count1 += 1

 if flag1 == True:
 print"Htan o:",secret

 answer=raw_input("Tha kaneis alli prospatheia? (y/n)")
 if answer == "y":
 flag2 = True
 flag1 = True
 secret, tries= dataFunction()
 elif answer == "n":
 flag2 = False

print "arithmos prospatheion:", count2
print "petiximenes prospatheies:", count1
print "apotiximenes prospatheies:", count2 - count1
```



### #askisi22

```
import random
flag = True
korona=0
grammata=0
count=0

while flag==True:
 nomisma = random.randint(1,2)
 if nomisma==1:
 korona+=1
 else:
 grammata+=1

 epilogi = input("Korona (1) h grammata (2):")
 if epilogi==nomisma:
 print"Sosto"
 count+=1
 else:
 print"Lathos"
 apantisi=raw_input("Tha xanadokimaseis?(n/o)")
 if apantisi=="n":
 flag=True
 else:
 flag=False

print "Sostes apantiseis:", count
print "Korona:", korona
print "Grammata", grammata
```



## Βιβλιογραφία

1. Αράπογλου Α., Βραχνός Ε., Κανίδης Ε., Μακρυγιάννης Π., Μπελεσιώτης Β., Τζήμας Δ. Αρχές Προγραμματισμού Υπολογιστών, 2020, Ινστιτούτο Τεχνολογίας Υπολογιστών και Εκδόσεων «Διόφαντος»
2. Ζάχαρης Ν., Μαυρομμάτης Γ., Εξ αποστάσεως εκπαίδευση στη γλώσσα Python, Εθνικό Κέντρο Δημόσιας διοίκησης και Αυτοδιοίκησης

### Διαδικτυακές πηγές

1. Αγγελιδάκης Ν., Εισαγωγή στον προγραμματισμό με την Python <http://aggelid.mysch.gr/pythonbook/> Πρόσβαση 2/9/2021
2. Μανής Γ., Εισαγωγή στον Προγραμματισμό με αρωγό τη γλώσσα Python <https://repository.kallipos.gr/handle/11419/2745> Πρόσβαση 2/9/2021
3. A byte of Python <https://open.umn.edu/opentextbooks/textbooks/581> Πρόσβαση 2/9/2021
4. Μαγκούτης Κ., Νικολάου Χ., Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python, <https://repository.kallipos.gr/handle/11419/1708> Πρόσβαση 2/9/2021

### Διαδικτυακά μαθήματα

ΠΕΚΕΣ Θεσσαλίας – Σεμινάριο στην Python

Mathesis -Εισαγωγή στην Python (<https://mathesis.cup.gr>)