

Εργασία 3

Προθεσμία: 6/1/2021, 23:59

Εισαγωγή

Θα ολοκληρώσετε ένα πρόγραμμα Python που κατασκευάζει μια τηλεφωνική ατζέντα και παρέχει λειτουργίες όπως εκτύπωση, αναζήτηση και ανανέωση πληροφοριών. Μέσα από αυτή την άσκηση θα κάνετε εξάσκηση στη δημιουργία και χρήση συναρτήσεων, χρήση επαναλήψεων, αλγοριθμική σκέψη, διαχείριση λεξικών, διαχείριση συμβολοσειρών και διαχείριση λιστών.

Διαβάστε όλη την εκφώνηση προσεκτικά, σκεφτείτε πώς θα σχεδιάσετε το πρόγραμμά σας και ολοκληρώστε το σταδιακά. Όταν είστε έτοιμοι να υλοποιήσετε μια από τις ζητούμενες συναρτήσεις, αφαιρέστε την εντολή `pass` από αυτή και προσθέστε τον δικό σας κώδικα. Πριν προχωρήσετε στην επόμενη συνάρτηση κάντε αρκετές δοκιμές για να βεβαιωθείτε ότι αυτό που έχετε γράψει μέχρι στιγμής λειτουργεί σωστά σε όλες τις περιπτώσεις. Εκμεταλλευτείτε το γεγονός ότι η λειτουργία εκτύπωσης της ατζέντας σας δίνεται έτοιμη.

Θα δουλέψετε πάνω στο αρχείο `project3.py` το οποίο σας δίνουμε. Η στοίχιση των εντολών στο αρχείο είναι χαρακτηρισ `tab`.

Συστάσεις/Απαιτήσεις:

Σε όλα τα μηνύματα που εκτυπώνει το πρόγραμμα τα οποία τελειώνουν σε άνω-κάτω τελεία (':') υπάρχει ένα κενό (space) μετά από αυτή.

Μετά από κάθε εντολή `input` γράψτε την εντολή `print()`.

Αν δε σας ζητείται ρητά να ελέγξετε την ορθότητα της εισόδου, τότε υποθέστε ότι θα είναι σωστή.

Εντοπίστε λειτουργίες που πιστεύετε ότι είναι αυτόνομες και κατάλληλες για ξεχωριστή υλοποίηση μέσω συνάρτησης και δημιουργήστε συναρτήσεις γι αυτές.

Όπως πάντα, προσέχετε να γράφετε ευανάγνωστο κώδικα με περιγραφικά ονόματα μεταβλητών και αποτελεσματικά σχόλια. Κάθε συνάρτηση πρέπει να έχει ένα σχόλιο που περιγράφει σύντομα τις παραμέτρους της και τι επιστρέφει. Ενδιάμεσα στον κώδικα προσθέτετε σχόλια μόνο εφόσον κρίνετε πως χρειάζεται κάποια επιπλέον διευκρίνιση η οποία πιθανώς να μην είναι προφανής από τον κώδικα.

Δεν επιτρέπεται να γράφετε κώδικα εκτός κάποιας συνάρτησης.

Παρόλο που προσπαθούμε να ρυθμίσουμε το `autolab` ώστε να μην κόβει μονάδες για λάθη σε χρήση κενών ή διαφορές κεφαλαίων/μικρών, μερικές φορές αυτό δεν επιτυγχάνεται πλήρως. Προσπαθήστε να γράφετε τα μηνύματα εξόδου ακριβώς όπως ζητείται.

Tip: Αν έχετε μεταβλητές `x` και `y` κι εκτυπώστε `print(x, y)` τότε θα εμφανιστούν με ένα κενό ανάμεσά τους. Αν θέλετε κάτι άλλο ανάμεσα, έχετε δύο επιλογές: Μπορείτε να χρησιμοποιήσετε την επιλογή `sep` με την οποία προσδιορίζετε τι θέλετε να βρίσκεται ανάμεσα στα δεδομένα που εκτυπώνει μια συγκεκριμένη `print`. Για παράδειγμα, το `print(x, y, sep='')` όπου το `''` είναι δύο μονά quotes, εκτυπώνει τις τιμές των `x, y` κολημένες, ενώ το `print(x, y, sep='*')` τις εκτυπώνει με ένα αστερίσκο ανάμεσά τους. Εναλλακτικά, εάν τα `x, y` είναι τύπου `string`, μπορείτε να χρησιμοποιήσετε πρόσθεση μεταξύ τους, για παράδειγμα, `print(x+y)` ή `print(x+'*'+y)`.

Άσκηση

Οι επαφές της τηλεφωνικής ατζέντας μπορούν να αποθηκευτούν σε ένα λεξικό όπου το κλειδί είναι ένα tuple αποτελούμενο από το *επώνυμο* και το *μικρό όνομα* της κάθε επαφής. Σε κάθε κλειδί αντιστοιχεί ένα λεξικό που περιλαμβάνει ζεύγη κλειδιών/τιμών για την αποθήκευση των παρακάτω πληροφοριών:

- *διεύθυνση email* (κλειδί 'email' και τιμή τύπου string),
- *τηλεφωνικοί αριθμοί* (κλειδί 'phone' και τιμή τύπου λεξικού το οποίο με τη σειρά του έχει κλειδιά 'mobile', 'home' και 'work' και αντίστοιχες τιμές τύπου string), και
- *μία λίστα από tags* (κλειδί 'tags' και τιμή μια λίστα από strings).

Το παρακάτω παράδειγμα παρουσιάζει μια ατζέντα με δύο επαφές:

```
phonebook = {  
    ('Papadopoulos', 'Dimitris'): {  
        'email': '-',  
        'phone': {  
            'mobile': 6995555555,  
            'home': 2421099999,  
            'other': '-',  
        },  
        'tags': ['neighbor', 'computer wiz', 'friend']  
    },  
    ('Nikolaou', 'Maria'): {  
        'email': 'maria@myemail.gr',  
        'phone': {  
            'mobile': '-',  
            'home': 2421055555,  
            'other': '-',  
        },  
        'tags': ['friend', 'coworker']  
    }  
}
```

Μη-διαθέσιμες πληροφορίες κωδικοποιούνται ως το string '-'.

Σας παρέχουμε τον παρακάτω κώδικα στο αρχείο `project3.py` :

- Μια συνάρτηση με όνομα `contact2csv` η οποία παίρνει ως ορίσματα:
 - ένα tuple αποτελούμενο από το επώνυμο και μικρό όνομα μιας επαφής και
 - ένα dict με τα στοιχεία που έχουν αποθηκευθεί σε μια τηλεφωνική ατζέντα για το άτομο με το συγκεκριμένο ονοματεπώνυμο.

Η συνάρτηση επιστρέφει ένα string το οποίο αποτελείται από τις πληροφορίες που εμφανίζονται στα ορίσματα σε μορφή CSV¹, με τη σειρά: *επώνυμο, όνομα, email, κινητό, τηλέφωνο σπιτιού, άλλο τηλέφωνο, tags*. Τα tags είναι ταξινομημένα και χωρισμένα μεταξύ τους με το σύμβολο της πρόσθεσης, +.

- Μια συνάρτηση με όνομα `phonebook2csv` η οποία παίρνει ως όρισμα μία τηλεφωνική ατζέντα της μορφής που περιγράφεται παραπάνω και επιστρέφει ένα string που αποτελείται από ένα τίτλο και ακολούθως τις επαφές σε μορφή CSV (όπως τις επιστρέφει η συνάρτηση `contact2csv`²) και ταξινομημένες με βάση το ονοματεπώνυμο.
 - Για παράδειγμα, για την παραπάνω ατζέντα η συνάρτηση `phonebook2csv` θα επιστρέψει το string:

```
Last,First,Email,Mobile,Home,Other,Tags
Nikolaou,Maria,maria@myemail.gr,-,2421055555,-,coworker+friend
Papadopoulos,Dimitris,-,6995555555,2421099999,-,computer wiz+friend+neighbor
```

- Την κύρια επανάληψη του προγράμματος όπου εκτυπώνεται ένα μενού επιλογών και ανάλογα με την επιλογή του χρήστη καλείται μια συνάρτηση που υλοποιεί την επιθυμητή λειτουργία.

Ακολουθούν οι περιγραφές των επιμέρους συναρτήσεων που πρέπει να υλοποιήσετε. Κάθε αναφορά στην τηλεφωνική ατζέντα ή σε CSV υπονοεί τις μορφές που έχουν περιγραφεί παραπάνω.

`add_contact` (phbook) :

Παίρνει ως όρισμα μια τηλεφωνική ατζέντα και υλοποιεί την προσθήκη μιας επαφής:

Χρησιμοποιεί την εντολή `input` για να εμφανίσει το μήνυμα `'Enter contact in CSV format: '` και να διαβάσει από το πληκτρολόγιο τα στοιχεία μίας επαφής σε μορφή CSV.

Αν υπάρχει ήδη επαφή γι αυτό το άτομο στην ατζέντα, εκτυπώνει το μήνυμα `'Contact X,Y already exists.'` όπου X το επώνυμο και Y το μικρό όνομα του ατόμου, χωρίς να κάνει κάτι άλλο. Διαφορετικά προσθέτει μια νέα επαφή με τα στοιχεία που δόθηκαν κι εκτυπώνει το μήνυμα `'Contact X,Y added.'` όπου X το επώνυμο και Y το μικρό όνομα του ατόμου. Σε αυτά τα μηνύματα, ανάμεσα στα ονόματα υπάρχει κόμμα χωρίς κενά ενώ ανάμεσα σε όλα τα υπόλοιπα μέρη του κάθε μηνύματος υπάρχει ένα κενό.

1 CSV (comma-separated values) είναι ένας τρόπος αναπαράστασης δεδομένων σε μορφή string με κόμμα ανάμεσα σε διαδοχικά πεδία.

2 Δείτε πώς χρησιμοποιούμε τη συνάρτηση `contact2csv` για την επιμέρους λειτουργία εκτύπωσης μίας επαφής. Όχι μόνο είναι πιο κατανοητός ο κώδικας έτσι, αλλά η ύπαρξη της `contact2csv` θα μας διευκολύνει σε άλλες περιπτώσεις που ενδιαφερόμαστε να εκτυπώσουμε μόνο μία επαφή.

search_by_name(phbook) :

Παίρνει ως όρισμα μια τηλεφωνική ατζέντα και υλοποιεί αναζήτηση επαφής με δεδομένο το μέρος ενός ονόματος. Διαβάστε όλη την περιγραφή της συνάρτησης και σκεφτείτε με τι σειρά θα υλοποιήσετε κάθε λειτουργία πριν αρχίσετε να γράφετε κώδικα.

Χρησιμοποιεί την εντολή `input` για να εμφανίσει το μήνυμα `'Enter partial name: '` και να διαβάσει από το πληκτρολόγιο μέρος ενός ονόματος.

Αν στην ατζέντα δεν υπάρχουν άτομα με επώνυμο ή μικρό όνομα που να περιλαμβάνει το μέρος ονόματος που δόθηκε, χωρίς διάκριση σε κεφαλαία/μικρά, τότε εκτυπώνει το μήνυμα `'No results.'` και επιστρέφει `None`.

Διαφορετικά, για κάθε επαφή που ταιριάζει, εκτυπώνει σε νέα γραμμή **αύξοντα αριθμό αποτελέσματος, μια τελεία, ένα κενό, επώνυμο, κόμμα, ένα κενό και το μικρό** όνομα αυτής. Τα αποτελέσματα πρέπει να εμφανίζονται ταξινομημένα με κριτήριο το επώνυμο, ή για ίδια επώνυμα, το μικρό. Ο αύξων αριθμός αποτελέσματος ξεκινά από 1.

Εφόσον βρέθηκαν επαφές που ταιριάζουν, χρησιμοποιεί την εντολή `input` για να εμφανίσει το μήνυμα `'Choose result: '` και να διαβάσει από το πληκτρολόγιο τον αύξοντα αριθμό του αποτελέσματος που επιθυμεί να δει ο χρήστης. Αν ο χρήστης δώσει αριθμό εκτός του σωστού εύρους αποτελεσμάτων, εκτυπώνεται το μήνυμα `'Choose between 1 and X.'` όπου **X** ο μέγιστος αύξων αριθμός κι επαναλαμβάνεται η ανάγνωση. Εφόσον δοθεί έγκυρος αριθμός, εκτυπώνει σε CSV μορφή την επαφή που αντιστοιχεί σε αυτά τα στοιχεία κι επιστρέφει το κλειδί της επαφής.

Για παράδειγμα, σε μια ατζέντα που περιλαμβάνει τις δύο επαφές που αναφέρονται στο αρχικό παράδειγμα, αν το μέρος ονόματος είναι `ri`, τότε η συνάρτηση θα εμφανίσει αρχικά τα αποτελέσματα:

1. Nikolaou, Maria
2. Papadopoulos, Dimitris

ενώ αν το μέρος ονόματος είναι `mar`, θα επιστρέφει αρχικά το αποτέλεσμα:

1. Nikolaou, Maria

update_contact(phbook) :

Παίρνει ως όρισμα μια τηλεφωνική ατζέντα και υλοποιεί ανανέωση στοιχείων συγκεκριμένης επαφής.

Αναζητά την επαφή με βάση το όνομα, *ακριβώς* όπως η συνάρτηση `search_by_name`.

Αν δε βρεθεί επαφή, εκτυπώνει το μήνυμα `'Contact not found.'` και τερματίζει.

Διαφορετικά, χρησιμοποιεί την εντολή `input` για να εμφανίσει το μήνυμα `'Field to update: '` και να διαβάσει από το πληκτρολόγιο το πεδίο της επαφής προς ανανέωση. Έγκυρα πεδία είναι `'email'`, `'phone'`, `'tags'`.

Αν δε δοθεί έγκυρο πεδίο, εκτυπώνει το μήνυμα `'Invalid field X.'` όπου **X** το πεδίο που έδωσε ο χρήστης κι επαναλαμβάνει το βήμα.

Διαφορετικά, χρησιμοποιεί την εντολή `input` για να εμφανίσει το μήνυμα `'New value: '` και να διαβάσει από το πληκτρολόγιο τη νέα τιμή που θα αποθηκευτεί σε αυτό το πεδίο.

Στην περίπτωση που ανανεώνεται το `email`, ελέγχει αν έχει δοθεί συντακτικά έγκυρη διεύθυνση. Θεωρούμε ότι μια διεύθυνση `email` είναι συντακτικά έγκυρη αν ισχύουν τα παρακάτω:

- περιλαμβάνει ακριβώς ένα `'@'` και
- το κομμάτι μετά το `'@'` περιλαμβάνει τουλάχιστον μια τελεία και

- στο κομμάτι μετά το '@' υπάρχουν τουλάχιστον δύο χαρακτήρες πριν και τουλάχιστον δύο χαρακτήρες μετά από κάθε τελεία.

(Για ευκολία, δεν κάνουμε έλεγχο για ύπαρξη μη-έγκυρων χαρακτήρων. Μη χρησιμοποιήσετε regular expressions ή επιπλέον πακέτα της Python για τον έλεγχο εγκυρότητας του email).

Αν η διεύθυνση δεν είναι έγκυρη, εκτυπώνει το μήνυμα 'Invalid email X.' όπου X το email που έδωσε ο χρήστης, διαφορετικά αποθηκεύει τη νέα διεύθυνση στη θέση της παλιάς, εκτυπώνει το μήνυμα 'Email updated.'.

Στην περίπτωση που ανανεώνεται το τηλέφωνο, ελέγχει αν αυτό αποτελείται μόνο από ψηφία. Αν όχι, εκτυπώνει το μήνυμα 'Invalid phone X.' όπου X το τηλέφωνο που έδωσε ο χρήστης. Διαφορετικά, αν ο αριθμός ξεκινά από 2 τον αποθηκεύει στο 'home', αν ξεκινά από 6 στο 'mobile', και σε κάθε άλλη περίπτωση στο 'other', αντικαθιστώντας πιθανώς ήδη υπάρχοντα αριθμό. Ακολουθώντας εκτυπώνει 'Phone updated.'.

Στην περίπτωση που ανανεώνονται τα tags, εφόσον το νέο tag που δόθηκε (υποθέστε θα δοθεί μόνο ένα) δεν υπάρχει ήδη στη λίστα από tags (χωρίς να γίνεται διάκριση σε κεφαλαία/μικρά κατά τον έλεγχο), το προσθέτει σε αυτή κι εκτυπώνει το μήνυμα 'Tags updated.'. Αν το νέο tag υπάρχει ήδη, εκτυπώνει το μήνυμα 'Tag X exists.' όπου X το tag που έδωσε ο χρήστης.

search_tag(phbook):

Παίρνει ως όρισμα μια τηλεφωνική ατζέντα και υλοποιεί αναζήτηση με βάση τα tags:

Χρησιμοποιεί την εντολή input για να εμφανίσει το μήνυμα "Enter tags:" και να διαβάσει από το πληκτρολόγιο ένα ή περισσότερα tags. Υποθέστε ότι αν δοθούν περισσότερα από ένα, θα υπάρχει κόμμα ανάμεσα σε διαδοχικά tags.

Για κάθε tag που αναζητείται, εξετάζοντάς τα σε αύξουσα αλφαβητική σειρά, εκτυπώνει το tag με άνω-κάτω τελεία.

Μετά, αναζητά τις επαφές στις οποίες εμφανίζεται το tag (χωρίς διάκριση κεφαλαίου/μικρού, αλλά να ταιριάζει όλο το tag, όχι μόνο μέρος αυτού). Η εξέταση των επαφών γίνεται επίσης αλφαβητικά με βάση το επώνυμο ή για ίδια επώνυμο το μικρό όνομα. Κάθε μία που βρίσκει, την εκτυπώνει σε μορφή CSV με ένα tab στην αρχή της γραμμής. Αν δεν υπάρχουν επαφές με κάποιο tag, εκτυπώνει tab και το μήνυμα 'No results.\n'. Ο χαρακτήρας tab γράφεται '\t'³.

Για παράδειγμα, σε μια ατζέντα που περιλαμβάνει τις δύο επαφές που αναφέρονται στο αρχικό παράδειγμα, η αναζήτηση των tags 'friend,computer wizard,coworker' θα εμφανίσει την έξοδο:

```
computer wizard:
```

```
No results
```

```
coworker:
```

```
Nikolaou,Maria,maria@myemail.gr,-,2421055555,-,coworker+friend
```

```
friend:
```

```
Nikolaou,Maria,maria@myemail.gr,-,2421055555,-,coworker+friend
```

```
Papadopoulos,Dimitris,-,6995555555,2421099999,-,computer wiz+friend+neighbor
```

Παρατηρήστε πως τα tags που αναζητούμε εμφανίζονται με αλφαβητική σειρά (computer wizard, coworker, friend), και το ίδιο ισχύει και με τις επαφές που εμφανίζονται στην περίπτωση του friend.

³ Για παράδειγμα, η εντολή `print('\t')` εκτυπώνει ένα χαρακτήρα tab.

Αρχεία ελέγχου

Θα σας δώσουμε ενδεικτικές εισόδους και τις αντίστοιχες εξόδους, ώστε να μπορείτε να επιβεβαιώσετε την ορθότητα των αποτελεσμάτων σας.

Για κάθε έλεγχο μπορείτε να δείτε την είσοδο στο αρχείο `inX.txt` όπου `X` αύξων αριθμός του ελέγχου. Οι αντίστοιχες εξοδοί είναι στα αρχεία `outX.txt`.

ΠΡΟΣΟΧΗ:

- Τα αρχεία εισόδου περιέχουν ότι γράφει στο πληκτρολόγιο ο χρήστης που εκτελεί το πρόγραμμα.
- Τα αρχεία εξόδου περιέχουν μόνο τα μηνύματα που εμφανίζει το πρόγραμμα στην οθόνη κι έχουν παραχθεί αυτόματα. Αυτό σημαίνει πως δεν εμφανίζονται τα δεδομένα που εισήγαγε ο χρήστης ή πιθανά "enter" που πάτησε κατά την εισαγωγή των δεδομένων. Μη μπερδευτείτε με αυτό. Εσείς ακολουθείτε ακριβώς την εκφώνηση όσον αφορά την έξοδο.

Οδηγίες αποστολής στο Autolab

1. Συνδεθείτε στο [autolab](#), κι επιλέξτε την εργασία `project3`
2. Κάντε κλικ στο μήνυμα ακαδημαϊκής ακεραιότητας και μετά στο SUBMIT. Στο παράθυρο που θα εμφανιστεί εντοπίστε κι επιλέξτε το αρχείο `project3.py` που περιέχει το πρόγραμμά σας ώστε να το ανεβάσετε στο `autolab`.
3. Θα σας πάει αυτόματα στη σελίδα όπου μετά από λίγο θα εμφανιστεί ο βαθμός σας. Θα χρειαστεί να ανανεώσετε τη σελίδα μερικές φορές μέχρι να εμφανιστεί βαθμός. Ανατρέξτε στην εκφώνηση του labo για περισσότερες πληροφορίες αναφορικά με το πώς να διαβάζετε τα αποτελέσματα του `autolab`.