

**Ε.Α.Π./ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

**4η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ**

**ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2022-2023**

**xx/03/2023**

**Ημερομηνία παράδοσης εργασίας: Κυριακή 07/05/2023**

**Καταληκτική ημερομηνία παραλαβής: Τετάρτη <sup>1</sup> 10/05/2023**

**Ημερομηνία ανάρτησης ενδεικτικών λύσεων: Σάββατο 13/05/2023**

**Καταληκτική ημερομηνία αποστολής σχολίων στον φοιτητή: Κυριακή 28/05/2023**

**ΥΠΟΕΡΓΑΣΙΑ 1.**

**(βαθμοί 25)**

Διαχείριση Αρχείων – Αποθήκευση και ανάκληση αντικειμένων με την pickle

**ΥΠΟΕΡΓΑΣΙΑ 2.**

**(βαθμοί 25)**

Βιβλιοθήκες pandas και matplotlib.pyplot

**ΥΠΟΕΡΓΑΣΙΑ 3.**

**(βαθμοί 25)**

Βιβλιοθήκη sqlite3

**ΥΠΟΕΡΓΑΣΙΑ 4.**

**(βαθμοί 25)**

Βιβλιοθήκη numPy και matplotlib.pyplot

**ΣΥΝΟΛΟ**

**(βαθμοί 100)**

---

<sup>1</sup> Σύμφωνα με τον Κανονισμό Σπουδών, η καταληκτική ημερομηνία για την παραλαβή της Γ.Ε. από το μέλος ΣΕΠ είναι η επόμενη Τετάρτη από το τέλος της εβδομάδας παράδοσης Γ.Ε.

**ΥΠΟΕΡΓΑΣΙΑ 1.****(βαθμοί 25)**

Στην υποεργασία 1 της 2<sup>ης</sup> ΓΕ ζητήθηκε να υλοποιηθεί μια μικρή εφαρμογή διαχείρισης καλαθιού αγορών σε ένα super market, που επιτρέπει στο χρήστη να προσθαφαιρέσει προϊόντα στο καλάθι και να προχωρήσει σε αγορά. Η ενδεικτική λύση, που δόθηκε, προσαρμόστηκε στο αντικειμενοστρεφές μοντέλο προγραμματισμού και δίνεται έτοιμη (ως άρθρωμα/module) στο αρχείο market.py. Στα αρχείο αυτό υπάρχουν δύο κλάσεις (Market και ShoppingCart), οι οποίες περιλαμβάνουν τις μεθόδους που αντιστοιχούν στις συναρτήσεις της ενδεικτικής λύσης και τη μέθοδο *shopping\_cart\_management*, που αντιστοιχεί στο κυρίως πρόγραμμα αυτής.

Στην προαναφερόμενη υλοποίηση καλαθιού αγορών δεν υπάρχει δυνατότητα αποθήκευσης και ανάκλησης του καλαθιού. Σε αυτή την υποεργασία καλείστε να συμπληρώσετε μια επέκταση (υποκλάση) του καλαθιού αγορών, που δίνει τη δυνατότητα το καλάθι αγορών να αποθηκευτεί σε ένα αρχείο και να ανακληθεί από αυτό (μέσω του αρθρώματος pickle).

Ειδικότερα, δίνεται το αρχείο-υπόδειγμα 1\_code\_template.py, στο οποίο περιλαμβάνονται δύο κλάσεις: η κλάση ShoppingCartStore (για την υποστήριξη των λειτουργιών της αποθήκευσης και ανάκλησης) και η κλάση StorableShoppingCart (για καλάθι αγορών που μπορεί να αποθηκευτεί σε αρχείο). Η πρώτη κλάση περιλαμβάνει δύο μεθόδους: i) τη μέθοδο *save*, για την αποθήκευση των περιεχομένων του καλαθιού σε αρχείο pickle, και ii) τη μέθοδο *load*, για την ανάγνωση περιεχομένων καλαθιού από αρχείο pickle. Η δεύτερη κλάση κληρονομεί από την ShoppingCart (του αρθρώματος/module market.py) και επαναορίζει μόνο τη μέθοδο κατασκευαστή *\_\_init\_\_* (που χειρίζεται την παράμετρο για τη αποθήκη του καλαθιού) και τη μέθοδο *shopping\_cart\_management* (για την προσθήκη επιλογών αποθήκευσης/ανάκλησης). Όλες οι άλλες μέθοδοι κληρονομούνται από την κλάση ShoppingCart.

Αφού μελετήσετε και κατανοήσετε τον κώδικα του αρχείου 1\_code\_template.py, καλείστε να:

- α) Ορίσετε τη μέθοδο *save* της κλάσης ShoppingCartStore, η οποία δέχεται ως παράμετρο τα περιεχόμενα του καλαθιού (λίστα) και τα αποθηκεύει σε αρχείο: Η συνάρτηση ανοίγει το αρχείο για εγγραφή δυαδικών/binary δεδομένων και γράφει στο αρχείο με τη μέθοδο *dump* της pickle. Σε περίπτωση σφάλματος, η συνάρτηση εμφανίζει μήνυμα αποτυχίας εγγραφής.
- β) Ορίσετε τη μέθοδο *load* της κλάσης ShoppingCartStore, η οποία επιστρέφει λίστα με τα αποθηκευμένα περιεχόμενα του καλαθιού τα οποία διαβάζει από το αρχείο pickle: Η συνάρτηση ανοίγει το αρχείο για ανάγνωση δυαδικών/binary δεδομένων και διαβάζει με τη μέθοδο *load* της pickle. Σε περίπτωση σφάλματος, η συνάρτηση εμφανίζει μήνυμα αποτυχίας ανάγνωσης και επιστρέφει άδεια λίστα.
- γ) Συμπληρώσετε την μέθοδο *\_\_init\_\_* της κλάσης StorableShoppingCart, ώστε να περιλαμβάνει, στις ιδιότητες του καλαθιού, ένα αντικείμενο τύπου ShoppingCartStore (που δημιουργείται με όρισμα το όνομα αρχείου, από την σχετική παράμετρο της *\_\_init\_\_*).
- δ) Προσθέσετε, στην μέθοδο *shopping\_cart\_management*, επιλογές (π.χ. 5 και 6) για την αποθήκευση σε αρχείο και ανάκληση από αρχείο του καλαθιού. Η αποθήκευση και ανάκληση να γίνεται με χρήση των μεθόδων *save* και *load* (από τα υποερωτήματα α και β) του αντικειμένου τύπου ShoppingCartStore του καλαθιού (από το υποερώτημα γ).

Θα πρέπει να χρησιμοποιηθεί αμυντικός προγραμματισμός (π.χ. με τη δομή try/except ή/και τη δομή with) όπου είναι απαραίτητος (και ειδικότερα στις μεθόδους *save* και *load*, καθώς κατά την αποθήκευση και ανάκληση μπορεί να προκληθούν διάφορα απρόσμενα σφάλματα).

Ενδεικτική έξοδος (μετά από εισαγωγή κάποιων προϊόντων, που δεν δείχνεται για οικονομία χώρου) με αποθήκευση του καλαθιού, διαγραφή ενός προϊόντος και ανάκληση ώστε να αποκατασταθεί το καλάθι όπως ήταν προ της διαγραφής.

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 2

ΑΑ	ΕΙΔΟΣ	ΤΜΧ	ΤΙΜΗ	ΤΜΧ	ΑΞΙΑ
1.	ΓΑΛΑ 1LT	4	1.5€		6.0€
2.	ΠΑΓΩΤΟ	2	2.5€		5.0€
					ΣΥΝΟΛΟ: 11.0€

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 5

Επιτυχής αποθήκευση περιεχομένων καλαθιού.

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 3

ΑΑ	ΕΙΔΟΣ	ΤΜΧ	ΤΙΜΗ	ΤΜΧ	ΑΞΙΑ
1.	ΓΑΛΑ 1LT	4	1.5€		6.0€
2.	ΠΑΓΩΤΟ	2	2.5€		5.0€
					ΣΥΝΟΛΟ: 11.0€

Επέλεξε γραμμή προϊόντος προς αφαίρεση: 2

Παρακαλώ επιβεβαιώστε την αφαίρεση (ν/ο): ν

Το προϊόν αφαιρέθηκε επιτυχώς από το καλάθι αγορών

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 2

ΑΑ	ΕΙΔΟΣ	ΤΜΧ	ΤΙΜΗ	ΤΜΧ	ΑΞΙΑ
1.	ΓΑΛΑ 1LT	4	1.5€		6.0€
					ΣΥΝΟΛΟ: 6.0€

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 6

Επιλογές

=====

1. Προσθήκη προϊόντων στο καλάθι, 2. Εμφάνιση περιεχομένου καλαθιού, 3. Αφαίρεση προϊόντος, 4. Πληρωμή, 5. Αποθήκευση καλαθιού, 6. Ανάκληση καλαθιού

Εισάγετε την επιλογή σας: 2

ΑΑ	ΕΙΔΟΣ	ΤΜΧ	ΤΙΜΗ	ΤΜΧ	ΑΞΙΑ
1.	ΓΑΛΑ 1LT	4	1.5€		6.0€
2.	ΠΑΓΩΤΟ	2	2.5€		5.0€
					ΣΥΝΟΛΟ: 11.0€

**Υπόδειξη:** Να χρησιμοποιηθεί το αρχείο **1\_code\_template.py** ως οδηγός επίλυσης.

**ΥΠΟΕΡΓΑΣΙΑ 2.**

(βαθμοί 25)

**Χρονοσειρές Μετοχών.** Στόχος της συγκεκριμένης υποεργασίας είναι η εξοικείωσή σας με τις εξωτερικές βιβλιοθήκες `pandas` και `matplotlib`. Στην υποεργασία αυτή θα φορτώσουμε εξωτερικά δεδομένα τύπου χρονοσειράς από ιστοσελίδα στον παγκόσμιο ιστό και συγκεκριμένα δεδομένα (τιμές) μετοχών εταιρειών του δείκτη Standard and Poor 500 (S&P 500) του χρηματιστηρίου της Νέας Υόρκης των Η.Π.Α. από την ιστοσελίδα του Yahoo Finance, που είναι μία από τις βασικές πηγές χρηματιστηριακών δεδομένων τα οποία είναι ελεύθερα διαθέσιμα στο ευρύ κοινό. Για την μετοχή της εταιρίας που θα επιλέγεται θα φορτώνουμε μία δομή `data frame` της βιβλιοθήκης `pandas` η οποία θα περιέχει χρονοσειρές σε μορφή στηλών με διάφορες τιμές της συγκεκριμένης μετοχής (όπως αυτές διατίθενται από το yahoo finance) για ένα χρονικό διάστημα που μας ενδιαφέρει και το οποίο θα δίνεται ως είσοδος από το χρήστη. Οι χρονοσειρές/στήλες για κάθε μετοχή που θα φορτώνουμε θα είναι οι: `Open`, `High`, `Low`, `Close`, `Adj Close`, και `Volume` με μία ακόμα στήλη `index` με την ονομασία `Date` που αφορά την αντίστοιχη ημερομηνία που σημειώνονται οι παραπάνω τιμές. Τα δεδομένα που φορτώνονται θα αποθηκεύονται και τοπικά σε αρχείο με μορφή `csv`. Στη συνέχεια για τις χρονοσειρές (στήλες του `data frame`) που φορτώνονται θα υπολογίζονται κάποια βασικά στατιστικά μέτρα με τη χρήση τη εξωτερικής βιβλιοθήκης `pandas` και θα σχεδιάζεται η γραφική τους παράσταση με τη χρήση της εξωτερικής βιβλιοθήκης `matplotlib`.

1. Φόρτωση Δεδομένων: Συμπληρώστε την συνάρτηση **`ts_timeperiod`** που σας δίνεται στο `template` ώστε το πρόγραμμα να φορτώνει δεδομένα για συγκεκριμένη χρονική περίοδο που θα δίνεται ως είσοδος από το χρήστη με τη μορφή:
  - τελευταία ημέρα της χρονικής περιόδου των τιμών της μετοχής (μορφή `dd/mm/yyyy`) και
  - πρώτη ημέρα της χρονικής περιόδου των τιμών της μετοχής (μορφή `dd/mm/yyyy`)

Κατά την εισαγωγή θα πρέπει να γίνονται έλεγχοι έτσι ώστε να διασφαλίζεται η ορθότητα των εισαγόμενων ημερομηνιών, καθώς και έλεγχος στην ημερομηνία έναρξης ώστε να μην είναι μεταγενέστερη από την ημερομηνία λήξης, καθώς επίσης και η ημερομηνία λήξης να μην είναι μεταγενέστερη από τη σημερινή ημερομηνία (εδώ θα χρειαστεί να κάνετε χρήση της βιβλιοθήκης `datetime` καθώς και της κλάσης `date` από την `datetime`).

2. Υπολογισμός Στατιστικών Μέτρων: Συμπληρώστε τη συνάρτηση **`ts_statistics`** η οποία παίρνει ως παράμετρο ένα `pandas data frame` που περιέχει τις χρονοσειρές που αφορούν μία μετοχή και υπολογίζει στατιστικά μέτρα για κάθε μία από τις χρονοσειρές (στήλες) του `data frame`. Η επιλογή της χρονοσειράς/στήλης για την οποία θα υπολογίζονται τα στατιστικά θα γίνεται από τον χρήστη ενώ θα πρέπει να υπάρχει η δυνατότητα να υπολογιστούν στατιστικά μέτρα και για κάποια/κάποιες από τις υπόλοιπες χρονοσειρές του `data frame`. Τα στατιστικά μέτρα που θα υπολογίζονται θα πρέπει να είναι αυστηρά τα εξής: `Min` (Minimum), `Max` (Maximum), `Mean` (Arithmetic Mean), `Standard Deviation` (`sd`) και `Variance` (`var`, όπου  $var=(sd)^2$ ).
3. Σχεδιασμός Γραφικής Παράστασης: Συμπληρώστε τη συνάρτηση **`ts_plot`** ώστε να αποτυπώνει γραφικά τις διάφορες χρονοσειρές/στήλες της μετοχής ως προς την αντίστοιχη ημερομηνία (στήλη `date`). Η επιλογή της χρονοσειράς/στήλης η οποία θα αποτυπώνεται γραφικά θα δίνεται ως είσοδος από το χρήστη, ενώ θα υπάρχει η δυνατότητα να σχεδιάζονται και κάποια/κάποιες από τις υπόλοιπες χρονοσειρές του `data frame`. Στον

άξονα X θα αποτυπώνεται η ημερομηνία, ενώ στον άξονα Y η χρονοσειρά/στήλη της μετοχής που κάθε φορά σχεδιάζεται.

**Προσοχή!** Να εφαρμοστεί αμυντικός προγραμματισμός σε όλα τα επίπεδα του προγράμματος. Το πρόγραμμα θα πρέπει να ακολουθεί τα βήματα του template. Ακολουθεί παράδειγμα εκτέλεσης.

```
Επιλογή εταιρικής μετοχής
Tickers εταιριών:, MMM, ADBE, AAPL, AMZN, AMD, AMAT, T, CAT, HPE, IBM, MSFT
Εισάγετε ένα από τα παραπάνω ticker μετοχής: amzn

Εισαγωγή χρονικής περιόδου για την οποία θα φορτωθούν τα δεδομένα
Εισάγετε την τελευταία ημέρα της χρονικής περιόδου για την οποία θα φορτωθούν τα δεδομένα
(format dd/mm/yyyy): 31/12/2022
Τελευταία ημέρα της χρονικής περιόδου: 2022-12-31
Εισάγετε την πρώτη ημέρα της χρονικής περιόδου για την οποία θα φορτωθούν τα δεδομένα (format
dd/mm/yyyy): 01/01/2022
Πρώτη ημέρα της χρονικής περιόδου: 2022-01-01

Στατιστικά μέτρα τιμών μετοχής
Χρονοσειρές της μετοχής (στήλες δεδομένων), Open, High, Low, Close, Adj Close, Volume
Εισάγετε χρονοσειρά (στήλη δεδομένων) από τις παραπάνω): close
Min = 81.8200
Max = 170.4045
Mean = 126.2600
Standard Deviation = 23.9935
Variance = 575.6867
Θέλετε να υπολογίσετε Στατιστικά Μέτρα και για άλλη Τιμή (Στήλη)?(Y/N): n

Γραφική παράσταση τιμών μετοχής
Χρονοσειρές της μετοχής (στήλες δεδομένων), Open, High, Low, Close, Adj Close, Volume
Εισάγετε χρονοσειρά (στήλη δεδομένων) από τις παραπάνω): close
Θέλετε να σχεδιάσετε και άλλη Τιμή (Στήλη)?(Y/N): n
```

**Υπόδειξη:** Να χρησιμοποιηθεί το αρχείο **2\_code\_template.py** ως οδηγός επίλυσης.

**Σημείωση:** Βεβαιωθείτε ότι στο σύστημά σας υπάρχει το πρόγραμμα *pip* (*pip3* σε MacOS, Linux) και στην συνέχεια κάνετε εγκατάσταση των βιβλιοθηκών: *pandas*, *openpyxl*, *matplotlib*, *time*, *datetime*.

**ΥΠΟΕΡΓΑΣΙΑ 3.**

(βαθμοί 25)

**Διαχείριση Βάσης Δεδομένων με SQLite:** Μία τοπική επιχείρηση Super Market, αποφασίζει να δημιουργήσει μία εφαρμογή, για τη διαχείριση του πελατολογίου της. Με τη χρήση sqlite3, έχει δημιουργηθεί μία αρχική βάση δεδομένων (*supermarket.db*), η οποία σας δίνεται και περιέχει τους εξής πίνακες:

- α) Πελάτες (**Customers**), με τα εξής πεδία: κωδικός (*id*), όνομα (*name*), επώνυμο (*surname*) και ηλεκτρονικό ταχυδρομείο (*email*). Ο κωδικός του κάθε πελάτη (*id*) δημιουργείται με αυτόματη αρίθμηση.
- β) Προϊόντα (**Products**), με τα εξής πεδία: κωδικός Προϊόντος (*prod\_id*), όνομα (*brand name*), είδος (*type*) και ημερομηνία παραγωγής (*prod\_date*).
- γ) Αγορές Προϊόντων (**Purchases**), με τα εξής πεδία: αριθμός (*no*), κωδικός πελάτη (*customer\_id*), κωδικός προϊόντος (*product\_id*), ποσότητα (*quantity*), τιμή (*price*) ημερομηνία συναλλαγής (*transaction\_date*). Ο αριθμός της αγοράς προϊόντος (*no*) δημιουργείται με αυτόματη αρίθμηση.

Στους παραπάνω πίνακες της βάσης δεδομένων supermarket.db έχουν γίνει κάποιες αρχικές καταχωρήσεις δεδομένων. Χρησιμοποιείτε την εφαρμογή DB browser for SQLite για να επιθεωρήσετε τα δεδομένα (δείτε τη σχετική σημείωση το τέλος της εκφώνησης).

Ζητείται να υλοποιηθεί εφαρμογή, η οποία θα εμφανίζει μενού με τις ακόλουθες επιλογές:  
1) Προβολή πελατών, 2) Προβολή προϊόντων, 3) Προβολή αγορών, 4) Καταχώρηση πελάτη 5) Διαγραφή πελάτη και 6) Έξοδος

Ειδικότερα:

- α) Θα πρέπει να συμπληρώσετε τη συνάρτηση *show\_records(table)*, η οποία θα παρουσιάζει τις εγγραφές ενός από τους πίνακες Customers, Products ή Purchases. Η συνάρτηση θα δέχεται ως όρισμα το όνομα του εκάστοτε πίνακα. Για την προβολή των εγγραφών του πίνακα θα πρέπει να χρησιμοποιήσετε το SQL script: "*SELECT \* from {}*". Η συνάρτηση *show\_records (table)* θα πρέπει να εμφανίζει τις ονομασίες των πεδίων του πίνακα καθώς και τις σχετικές εγγραφές.
- β) Θα πρέπει να συμπληρώσετε τη συνάρτηση *register\_customer(name, surname, email)*, η οποία θα καταχωρεί ένα νέο πελάτη στον πίνακα Customers. Η συνάρτηση θα δέχεται ως ορίσματα το όνομα, το επώνυμο και το email του πελάτη. Για την εισαγωγή ενός νέου πελάτη θα πρέπει να χρησιμοποιήσετε το SQL script: "*INSERT INTO Customer (name, surname, email) VALUES (?, ?, ?)*"
- γ) Θα πρέπει να συμπληρώσετε τη συνάρτηση *delete\_customer(code)*, η οποία θα διαγράφει ένα πελάτη από τον πίνακα Customers. Η συνάρτηση θα δέχεται ως όρισμα τον κωδικό του πελάτη. Για τη διαγραφή ενός πελάτη θα πρέπει να χρησιμοποιήσετε το SQL script: "*DELETE FROM Customers WHERE id=(?)*". Να γίνει έλεγχος, μέσω της συνάρτησης *hasRows(table)*, εάν υπάρχουν εγγραφές πελατών προς διαγραφή, πριν εκτελεστεί η συνάρτηση διαγραφής. Στην περίπτωση που δεν υπάρχουν, να εκτυπώνεται ανάλογο μήνυμα και να συνεχίζει η εκτέλεση του μενού επιλογών.

Σε καθεμία από τις παραπάνω συναρτήσεις, να χρησιμοποιήσετε χειρισμό λαθών (π.χ. με τη δομή try/except ή/και τη δομή with) κατά τη σύνδεση με τη βάση δεδομένων supermarket.db και την εκτέλεση των εντολών SQL.

**Υπόδειξη:** Να χρησιμοποιηθεί το αρχείο **3\_code\_template.py** ως οδηγός επίλυσης.

**Σημείωση:** <sup>1</sup>Οι υπογραμμίσεις συγκεκριμένων πεδίων κατά τον ορισμό των πινάκων, σημαίνει πως λειτουργούν ως πεδία – κλειδιά για τους αντίστοιχους πίνακες. <sup>2</sup>Μπορείτε να διαχειριστείτε τη βάση δεδομένων *supermarket.db* με τη χρήση της εφαρμογής *DB Browser (SQLite)*. Μπορείτε να εγκαταστήσετε τη συγκεκριμένη εφαρμογή στον υπολογιστή σας από την ιστοσελίδα: <https://sqlitebrowser.org/dl/>

Παράδειγμα εκτέλεσης:

Επιλογές συστήματος:

1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος

Δώστε την επιλογή σας: 4

Καταχώρησε το όνομα του πελάτη: ΝΙΚΟΣ

Καταχώρησε το επώνυμο του πελάτη: ΠΑΠΑΝΙΚΟΛΑΟΥ

Καταχώρησε το email του πελάτη:nikrap@eap.gr

Επιλογές συστήματος:

1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος

Δώστε την επιλογή σας: 4

Καταχώρησε το όνομα του πελάτη: ΓΕΩΡΓΙΟΣ

Καταχώρησε το επώνυμο του πελάτη: ΖΑΧΑΡΙΑΔΗΣ

Καταχώρησε το email του πελάτη:gzah@eap.gr

Επιλογές συστήματος:

1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος

Δώστε την επιλογή σας: 4

Καταχώρησε το όνομα του πελάτη: ΕΛΕΝΗ

Καταχώρησε το επώνυμο του πελάτη: ΚΩΝΣΤΑΝΤΙΝΙΔΗ

Καταχώρησε το email του πελάτη:elenkon@eap.gr

Επιλογές συστήματος:

1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος

Δώστε την επιλογή σας: 4

Καταχώρησε το όνομα του πελάτη: ΓΕΩΡΓΙΑ

Καταχώρησε το επώνυμο του πελάτη: ΠΑΤΣΑΝΤΩΝΑΚΗ

Καταχώρησε το email του πελάτη:geopats@eap.gr

Δώστε την επιλογή σας: 1

['id', 'name', 'surname', 'email']

\*\*\*\*\*

(1, 'ΝΙΚΟΣ', 'ΠΑΠΑΝΙΚΟΛΑΟΥ', 'nikrap@eap.gr')

(2, 'ΓΕΩΡΓΙΟΣ', 'ΖΑΧΑΡΙΑΔΗΣ', 'gzah@eap.gr')

(3, 'ΕΛΕΝΗ', 'ΚΩΝΣΤΑΝΤΙΝΙΔΗ', 'elenkon@eap.gr')

(4, 'ΓΕΩΡΓΙΑ', 'ΠΑΤΣΑΝΤΩΝΑΚΗ', 'geopats@eap.gr')

\*\*\*\*\*

Επιλογές συστήματος:

1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη

```
6. Έξοδος
Δώστε την επιλογή σας: 5
Δώσε το ID του πελάτη προς διαγραφή: 4
Επιλογές συστήματος:
1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος
Δώστε την επιλογή σας: 1
['id', 'name', 'surname', 'email']
*****
(1, 'ΝΙΚΟΣ', 'ΠΑΠΑΝΙΚΟΛΑΟΥ', 'nikpap@eap.gr')
(2, 'ΓΕΩΡΓΙΟΣ', 'ΖΑΧΑΡΙΑΔΗΣ', 'gzah@eap.gr')
(3, 'ΕΛΕΝΗ', 'ΚΩΝΣΤΑΝΤΙΝΙΔΗ', 'elenkon@eap.gr')
*****
Επιλογές συστήματος:
1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος
Δώστε την επιλογή σας: 2
['prod_id', 'band_name', 'type', 'prod_date']
*****
(100, 'ΓΑΛΑΤΕΣ', 'ΓΑΛΑΚΤΟΚΟΜΙΚΑ', '17/2/2023')
(101, 'ΜΠΙΣΚΟΤΕΣ', 'ΜΠΙΣΚΟΤΑ', '5/1/2023')
(102, 'ΠΑΓΩΤΕΣ', 'ΓΑΛΑΚΤΟΚΟΜΙΚΑ', '10/2/2023')
*****
Επιλογές συστήματος:
1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος
Δώστε την επιλογή σας: 3
['no', 'customer_id', 'product_id', 'quantity', 'price', 'transaction_date']
*****
(1, 1, 100, 3, 12.5, '17/2/2023')
(2, 2, 101, 1, 5.5, '17/2/2023')
*****
Επιλογές συστήματος:
1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος
Δώστε την επιλογή σας: 9
Λάθος επιλογή. Παρακαλώ επιλέξτε από 1 έως 6

Επιλογές συστήματος:
1. Προβολή Πελατών
2. Προβολή Προϊόντων
3. Προβολή Αγορών
4. Καταχώρηση Πελάτη
5. Διαγραφή Πελάτη
6. Έξοδος
Δώστε την επιλογή σας: 6
Τέλος Προγράμματος!
```



**ΥΠΟΕΡΓΑΣΙΑ 4.**

(βαθμοί 25)

**Βέλτιστη αγορά τροφίμων:** Σκοπός της υποεργασίας αυτής είναι η επίδειξη των δυνατοτήτων της βιβλιοθήκης `numpy`, για την εύρεση της ποσότητας κάθε τροφής που χρειαζόμαστε, από ένα συνδυασμό τροφών, ώστε να λάβουμε όλα τα απαραίτητα θρεπτικά συστατικά, σε ημερήσια βάση. Ζητείται να συμπληρώσετε το πρόγραμμα του αρχείου `4_template.py` που περιέχει τα δεδομένα του ζητούμενου προγράμματος τα οποία είναι: α) μια λίστα `Foods` από τρεις τροφές, β) μια λίστα `Cost` με το κόστος κάθε τροφής από αυτές ανά γραμμάριο, γ) μια λίστα `Ingredients` από τρία Θρεπτικά Συστατικά (ΘΣ), δ) τρεις λίστες `I1`, `I2`, `I3` (μια για κάθε ΘΣ της `Ingredients` με την σειρά που βρίσκονται σε αυτή) με την περιεκτικότητα κάθε γραμμαρίου των τριών τροφών στο ΘΣ αυτό (με την σειρά που βρίσκονται αυτά στην `Foods`), ε) μια λίστα `Daily_Consumption` με την Συνιστώμενη Ημερήσια Ποσότητα (ΣΗΠ) σε γραμμάρια που πρέπει να λαμβάνουμε από το κάθε ΘΣ (με την σειρά πάλι της `Ingredients`) και στ) μια λίστα `Discount` που περιέχει τρεις εκπτώσεις σε τρία διαφορετικά καταστήματα (5%, 10% και 15%).

Συμβολίζουμε: α)  $b_i$  την ΣΗΠ από το ΘΣ  $i$  ( $B = [b_1 \ b_2 \ b_3]^T$ , δηλαδή στήλη, είναι η λίστα `Daily_Consumption`), β)  $q_i$  (με  $Q = [q_1 \ q_2 \ q_3]^T$ ) την ημερήσια ποσότητα που καταναλώνουμε από την τροφή  $i$  (ζητούμενο), και γ)  $a_{ij}$  (στοιχεία του πίνακα  $A$ ) την ποσότητα από το θρεπτικό συστατικό  $i$  που περιέχει ανά μονάδα βάρους η τροφή  $j$  ( $A = [I1 \ I2 \ I3]$ ). Έτσι προκύπτει ένα γραμμικό σύστημα εξισώσεων της μορφής:

$$\begin{aligned} a_{11}q_1 + a_{12}q_2 + a_{13}q_3 &= b_1 \\ a_{21}q_1 + a_{22}q_2 + a_{23}q_3 &= b_2 \\ a_{31}q_1 + a_{32}q_2 + a_{33}q_3 &= b_3 \end{aligned}$$

Η μοναδική λύση του παραπάνω συστήματος  $Q = A^{-1} * B^T$  δίνει την ποσότητα που πρέπει να αγοράζουμε από κάθε τροφή ημερησίως για να έχουμε ακριβώς την ΣΗΠ από κάθε ΘΣ. Δηλαδή με την χρήση της `transpose` για τον  $B$  (εύρεση  $B^T$ ) και της `linalg.solve` με ορίσματα  $A$  και  $B^T$  από την `numpy` βρίσκουμε την ζητούμενη λύση.

Έχοντας το διάνυσμα κόστους  $C$  (λίστα `Cost`, που περιέχει το κόστος κάθε τροφής ανά γραμμάριο), το συνολικό κόστος της διατροφής  $P$  υπολογίζεται ως το γινόμενο  $P = C * Q$  (άθροισμα γινομένων κόστους ανά γραμμάριο κάθε τροφής επί ποσότητα σε γραμμάρια κάθε τροφής). Η πράξη αυτή γίνεται με χρήση της `dot` από την `numpy` και ορίσματα τους δύο αυτούς πίνακες.

**ΖΗΤΕΙΤΑΙ:**

α) Να υπολογίσετε την ποσότητα που πρέπει να αγοράζεται από κάθε τροφή κάθε μέρα.

β) Να υπολογίσετε το συνολικό κόστος της συνιστώμενης διατροφής και το κόστος για την αγορά κάθε μίας από τις τροφές. (Υπόδειξη: Να γίνει χρήση της βιβλιοθήκης `multiply` από την `numpy` που θα πολλαπλασιάζονται ο `Cost` με τον ανάστροφο (`transpose`)  $Q$ )

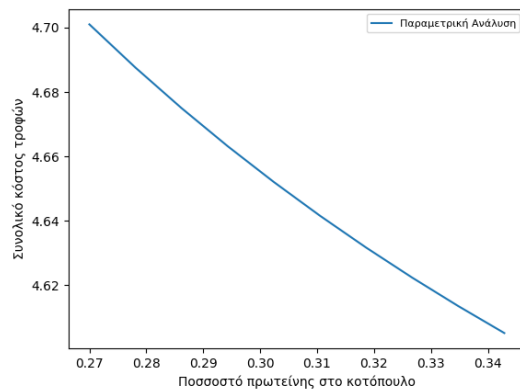
γ) Να βρείτε το συνολικό κόστος συνιστώμενης διατροφής σε κάθε μια από τις τρεις εκπτώσεις του πίνακα `Discount`.

δ) Να γίνει η γραφική παράσταση του κόστους της διατροφής (άξονας  $\psi$ ) σε σχέση με την περιεκτικότητα της πρωτεΐνης στο κοτόπουλο (άξονας  $\chi$ ). Η περιεκτικότητα της πρωτεΐνης στο κοτόπουλο δίδεται στον πίνακα `A_sens` και το αντίστοιχο κόστος έχει ήδη υπολογιστεί στον πίνακα `cost_sens`. (Υπόδειξη: Να γίνει χρήση της βιβλιοθήκης `plot` από την `matplotlib.pyplot`)

**Σημείωση:** Βεβαιωθείτε αρχικά ότι στο σύστημά σας υπάρχει το πρόγραμμα `pip` και στην συνέχεια κάνετε εγκατάσταση των βιβλιοθηκών: `numpy`, `matplotlib.pyplot` (`pip3` σε MacOS, Linux)

Ακολουθεί παράδειγμα εκτέλεσης.

Θα αγοραστούν 139.39 γραμμάρια Κοτόπουλο.  
Θα αγοραστούν 246.46 γραμμάρια Τυρί.  
Θα αγοραστούν 513.13 γραμμάρια Ρύζι.  
Το συνολικό κόστος είναι 4.70 ευρώ  
Θα χρειαστούν 0.70 ευρώ για αγορά Κοτόπουλο.  
Θα χρειαστούν 2.46 ευρώ για αγορά Τυρί.  
Θα χρειαστούν 1.54 ευρώ για αγορά Ρύζι.  
Με έκπτωση 5% το κόστος θα είναι 4.47 ευρώ.  
Με έκπτωση 10% το κόστος θα είναι 4.23 ευρώ.  
Με έκπτωση 15% το κόστος θα είναι 4.00 ευρώ.



**Γενικές Υποδείξεις:**

I) Για τις απαντήσεις της εργασίας μπορείτε να ανατρέξετε στη συμπληρωματική βιβλιογραφία που δίνεται και στα βοηθητικά κείμενα που υπάρχουν στον δικτυακό τόπο / portal της θεματικής ενότητας. Συνιστάται να προσθέσετε στο τέλος της εργασίας σας κατάλογο βιβλιογραφίας.

II) Οδηγίες σχετικές με τον κώδικα

- Το όνομα κάθε .py αρχείου να περιλαμβάνει το επώνυμό σας με λατινικούς χαρακτήρες, το χαρακτήρα της υπογράμμισης και τον αριθμό του συγκεκριμένου υποερωτήματος (π.χ. αν το επώνυμό σας είναι Γεωργίου, τότε ο κώδικας για την υποεργασία 1β θα έχει το όνομα Georgiou\_1b.py). Κάθε αρχείο κώδικα που θα παραδοθεί θα πρέπει τουλάχιστον να περνάει τη φάση της διερμηνείας χωρίς συντακτικά σφάλματα.
- Τα αρχεία .py θα πρέπει να τα ανοίξετε και να τα επεξεργαστείτε με το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα IDLE. Ο κώδικας να είναι επαρκώς σχολιασμένος, σωστά στοιχισμένος και ενσωματωμένος μέσα στο έγγραφο Word, με τις απαντήσεις σας σε γραμματοσειρά courier.
- Στο έγγραφο της απάντησής σας και στο αρχείο του κώδικα θα πρέπει να δίνεται ολόκληρο το πρόγραμμα, επισημαίνοντας με σχόλια πού απαντάτε κάθε ερώτημα ώστε να θεωρούνται πλήρεις οι απαντήσεις.
- Όλα τα .py αρχεία με τον πηγαίο κώδικα και το .doc αρχείο κειμένου να υποβληθούν μέσω της πλατφόρμας <https://study.eap.gr>.

III) Τρόπος παράδοσης εργασίας:

α) Οι απαντήσεις πρέπει να είναι γραμμένες με χρήση **επεξεργαστή κειμένου** (π.χ. **Word**) σε σελίδες **διαστάσεων A4 χωρίς χρώματα**. Το αρχείο να περιέχει ως **πρώτη σελίδα** το κείμενο του **Εντύπου Υποβολής – Αξιολόγησης** και ως δεύτερη σελίδα τον τίτλο «Σχόλια προς τον φοιτητή» (θα συμπληρωθεί από τον καθηγητή σας). Οι απαντήσεις στις υποεργασίες θα αρχίζουν από την τρίτη σελίδα, **χωρίς να επαναλαμβάνονται οι εκφωνήσεις**. Κάθε υποεργασία θα αρχίζει από νέα σελίδα. Για την απάντησή σας θα πρέπει να χρησιμοποιείτε υποχρεωτικά το **Πρότυπο Υποβολής Γραπτής Εργασίας**.

β) Το .doc αρχείο κειμένου να υποβληθεί στη διεύθυνση <https://study.eap.gr> με **όνομα αρχείου το επώνυμό σας με λατινικούς χαρακτήρες και τον Αριθμό Μητρώου σας**, π.χ. Ioannou\_82345.

IV) Η καλή παρουσίαση της εργασίας λαμβάνεται υπόψη στην αξιολόγηση της εργασίας.

\*\*\*\*\*