

**Ε.Α.Π./ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

**4η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ**

**ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2023-2024**

**02/04/2024**

**Ημερομηνία παράδοσης εργασίας: Κυριακή 12/05/2024**

**Καταληκτική ημερομηνία παραλαβής: Τετάρτη <sup>1</sup> 15/05/2024**

**Ημερομηνία ανάρτησης ενδεικτικών λύσεων: Σάββατο 18/05/2024**

**Καταληκτική ημερομηνία αποστολής σχολίων στον φοιτητή: Κυριακή 02/06/2024**

**ΥΠΟΕΡΓΑΣΙΑ 1.**

**(βαθμοί 30)**

Διαχείριση Αρχείων – Αποθήκευση και ανάκληση αντικειμένων με την pickle

**ΥΠΟΕΡΓΑΣΙΑ 2.**

**(βαθμοί 35)**

Βιβλιοθήκη numPy

**ΥΠΟΕΡΓΑΣΙΑ 3.**

**(βαθμοί 35)**

Βιβλιοθήκες pandas, 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 30)

Ένα μικρό ξενοδοχείο κάνει χρήση ενός εμπορικού συστήματος κρατήσεων το οποίο μπορεί να εξάγει τις κρατήσεις για μία ημέρα σε pickled αρχείο. Θέλει να φτιάξει μια απλή εφαρμογή η οποία μετατρέπει το αρχείο κρατήσεων σε αρχείο κειμένου. Το αρχείο περιέχει μία λίστα με πλειάδες που αντιπροσωπεύουν τις κρατήσεις. Κάθε πλειάδα περιέχει τον αριθμό κράτησης, τον αριθμό δωματίου, το επώνυμο και το όνομα του πελάτη. Σας δίνεται το αρχείο κρατήσεων “my\_reservations\_20240407” το οποίο περιέχει τις κρατήσεις για τις 07/04/2024. Κατά την έναρξη της εφαρμογής να φορτώνεται το περιεχόμενο του αρχείου των κρατήσεων. Στη συνέχεια η εφαρμογή θα αποθηκεύσει τις κρατήσεις σε αρχείο κειμένου και να κάνει έξοδο από την εφαρμογή. Το όνομα του αρχείου αποθήκευσης θα είναι το όνομα του αρχείου κρατήσεων με επέκταση ‘.txt’.

Καλείστε να:

- α) Ορίσετε τη συνάρτηση *load* η οποία παίρνει ως παράμετρο το όνομα του αρχείου και φορτώνει τη λίστα με τις κρατήσεις από το αρχείο pickle: Η συνάρτηση ανοίγει το αρχείο για ανάγνωση δυαδικών/binary δεδομένων, διαβάζει τη λίστα με τη μέθοδο *load* της pickle και επιστρέφει τη λίστα.

Σε περίπτωση σφάλματος, η συνάρτηση εμφανίζει μήνυμα αποτυχίας ανάγνωσης και επιστρέφει άδεια λίστα.

- β) Ορίσετε τη συνάρτηση *save* η οποία αποθηκεύει τις κρατήσεις της λίστας ταξινομημένες βάση του αριθμού δωματίου σε αρχείο κειμένου: Η συνάρτηση λαμβάνει ως παράμετρο το όνομα του αρχείου αποθήκευσης και τη λίστα με τις κρατήσεις, στη συνέχεια ανοίγει το αρχείο για εγγραφή δεδομένων κειμένου, ταξινομεί τη λίστα κατά αύξουσα σειρά βάση του αριθμού δωματίου, προσθέτει μια γραμμή επικεφαλίδας στη μορφή:

[Κεντραρισμένο σε 9 χαρακτήρες][Αριστερά Στοιχισμένο]

Δωμάτιο	Πελάτης
---------	---------

και αποθηκεύει κάθε κράτηση στη μορφή:

[Κεντραρισμένο σε 9 χαρακτήρες][Αριστερά Στοιχισμένο]

<αριθμός δωματίου>	<Επώνυμο Πελάτη><κενό><Όνομα Πελάτη>
--------------------	--------------------------------------

Σε περίπτωση σφάλματος, η συνάρτηση εμφανίζει μήνυμα αποτυχίας εγγραφής.

Θα πρέπει να χρησιμοποιηθεί αμυντικός προγραμματισμός (π.χ. με τη δομή try/excerpt 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).

Η πρώτες γραμμές του αρχείου εξόδου φαίνεται παρακάτω:

```
Δωμάτιο  Πελάτης
101      Καλομύτρου Γεωργία
102      Καλομύτρου Γεωργία
103      Αγαπίου Ερωτόκριτος
```

**ΥΠΟΕΡΓΑΣΙΑ 2.**

(βαθμοί 35)

Οι τιμές αγοράς της ενέργειας σε MWh (Mega Watt Hour) για τις 24 ώρες (ωριαίες περιόδους) μιας συγκεκριμένης μέρας είναι: είναι: 56.7, 73.53, 62.55, 55.01, 57.55, 75.89, 80, 91.7, 86.69, 81, 71.93, 58.8, 60.37, 60.37, 67, 73.43, 78.71, 79.04, 100.03, 83.43, 77.31, 51.47, 60.37, 67.08. Η ποσότητα ενέργειας σε MWh η οποία αγοράστηκε κάθε χρονική περίοδο είναι: 12, 30, 15, 67, 56, 43, 32, 15, 46, 48, 96, 123, 12, 7, 69, 3, 60, 213, 34, 56, 78, 22, 56, 78.

Με τη χρήση της βιβλιοθήκης `numpy`:

(α) Να δημιουργηθεί:

- ι) ένας πίνακας με όνομα `energy_prices` που να περιέχει τις τιμές ενέργειας,
- ιι) ένας πίνακας με όνομα `quantity` που να περιέχει τις ποσότητες που διακινήθηκαν και
- iii) ένας πίνακας 4 γραμμών και έξι στηλών με όνομα `energy_by_six` στον οποίο κάθε γραμμή να περιέχει τις τιμές καθενός από τα 4 εξάωρα της μέρας ο οποίος και θα τυπωθεί

(β) Να υπολογιστεί το συνολικό κόστος της ενέργειας που αγοράστηκε σε όλες τις χρονικές στιγμές.

(γ) Να υπολογιστεί πόσες MWh διακινήθηκαν στην αγορά σε όλες στις χρονικές στιγμές συνολικά.

(δ) Να τυπωθεί η ελάχιστη τιμή ανά εξάωρο με την χρήση του πίνακα `energy_by_six`.

(ε) Να τυπωθεί η μέση τιμή ανά εξάωρο με την χρήση του πίνακα `energy_by_six`.

(στ) Να ταξινομηθούν οι τιμές (αύξουσα σειρά) χωρίς να τροποποιηθούν τα δεδομένα του `energy_prices`.

Παράδειγμα εκτέλεσης:

Η παρουσίαση των δεδομένων σε εξάωρη βάση είναι:

```
[ [ 56.7  73.53  62.55  55.01  57.55  75.89]
  [ 80.   91.7   86.69  81.    71.93  58.8 ]
  [ 60.37 60.37  67.    73.43  78.71  79.04]
  [100.03 83.43  77.31  51.47  60.37  67.08]]
```

Το συνολικό κόστος είναι 91342.23

Διακινήθηκαν 1271 MWh στην αγορά

Το εξάωρο 1 η ελάχιστη τιμή είναι 55.01

Το εξάωρο 2 η ελάχιστη τιμή είναι 58.80

Το εξάωρο 3 η ελάχιστη τιμή είναι 60.37

Το εξάωρο 4 η ελάχιστη τιμή είναι 51.47

Το εξάωρο 1 η μέση τιμή είναι 63.54

Το εξάωρο 2 η μέση τιμή είναι 78.35

Το εξάωρο 3 η μέση τιμή είναι 69.82

Το εξάωρο 4 η μέση τιμή είναι 73.28

Η ταξινομημένες τιμές της ενέργειας είναι: [ 51.47 55.01 56.7 57.55 58.8 60.37 60.37 60.37 62.55 67.

67.08 71.93 73.43 73.53 75.89 77.31 78.71 79.04 80. 81. 83.43 86.69 91.7 100.03]

**Σημείωση:** Βεβαιωθείτε ότι στο σύστημά σας υπάρχει το πρόγραμμα `pip` (`pip3` σε MacOS, Linux) και στην συνέχεια κάνετε εγκατάσταση της βιβλιοθήκης `numpy`

**ΥΠΟΕΡΓΑΣΙΑ 3.**

(βαθμοί 35)

Το αρχείο `3_code_template.py` περιέχει το υπόδειγμα ενός προγράμματος που διαχειρίζεται κάποια ημερήσια κλιματικά δεδομένα (Θερμοκρασία, Υγρασία και Υετός) πέντε πόλεων του Νομού Έβρου (Αλεξανδρούπολη, Φέρες, Σουφλί, Διδυμότειχο, Ορεστιάδα).

**ΖΗΤΕΙΤΑΙ:**

A) να δημιουργήσετε ένα pandas DataFrame για τον χειρισμό των ημερήσιων κλιματικών δεδομένων που σας δίνονται.

B) Να συμπληρώσετε τις παρακάτω συναρτήσεις:

- 1) `def display_climate_info()` που προβάλλει όλα τα κλιματικά δεδομένα
- 2) `def calculate_descriptive_statistics()` που επιστρέφει τα περιγραφικά στατιστικών (descriptive statistics) όλων των κλιματικών δεδομένων
- 3) `def calculate_average_temp()` # που επιστρέφει την μέση θερμοκρασία
- 4) `def plot_humidity()` # που δημιουργεί και εμφανίζει ένα ραβδόγραμμα (bar plot) της υγρασίας

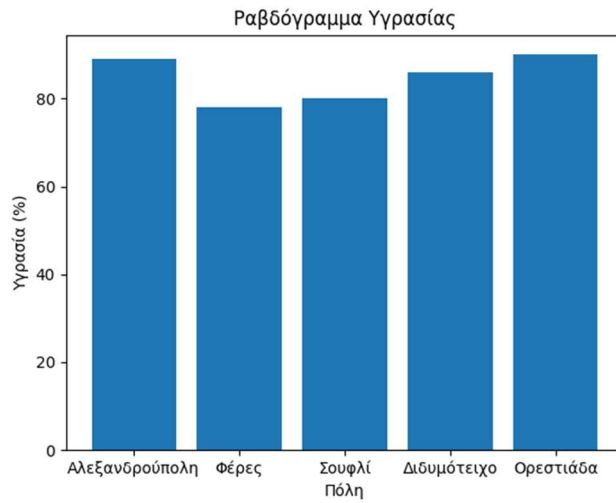
Γ) Να δημιουργήσετε ένα μενού επιλογών με βάση τις συναρτήσεις που έχετε δημιουργήσει.

**Υπόδειξη:** Να χρησιμοποιηθεί το αρχείο `3_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`, `numpy` και `matplotlib`

**ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ:**

```
Μενού Κλιματικών Δεδομένων:
1. Προβολή κλιματικών δεδομένων
2. Υπολογισμός περιγραφικών στατιστικών (descriptive statistics)
3. Υπολογισμός μέσης θερμοκρασίας
4. Εκτύπωση ραβδογράμματος υγρασίας
5. Έξοδος
Εισάγετε την επιλογή σας: 1
Κλιματικά Δεδομένα:
      Πόλη  Θερμοκρασία  Υγρασία  Υετός
0  Αλεξανδρούπολη      21      89   56.7
1        Φέρες        20      78   64.3
2        Σουφλί        17      80   59.1
3  Διδυμότειχο        19      86   48.9
4    Ορεστιάδα        20      90   47.6
.....
Εισάγετε την επιλογή σας: 2
Περιγραφικά Στατιστικά:
      Θερμοκρασία  Υγρασία  Υετός
count    5.000000    5.000000    5.000000
mean     19.400000   84.600000   55.320000
std       1.516575    5.366563    7.029367
min       17.000000   78.000000   47.600000
25%       19.000000   80.000000   48.900000
50%       20.000000   86.000000   56.700000
75%       20.000000   89.000000   59.100000
max       21.000000   90.000000   64.300000
.....
Εισάγετε την επιλογή σας: 3
Μέση Θερμοκρασία: 19.40
.....
Εισάγετε την επιλογή σας: 4
```



.....  
Εισάγετε την επιλογή σας: 5  
Έξοδος. Καλή συνέχεια!

**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.

\*\*\*\*\*