

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΤΟΜΕΑΣ ΛΟΓΙΚΟΥ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΜΑΘΗΜΑ: ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΜΕΤΑΦΡΑΣΤΩΝ  
(CEID\_NY132)  
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021

**PRINCIPLES OF PROGRAMMING LANGUAGES  
AND COMPILER DESIGN 2021:  
PYTHON PROJECT  
ΤΕΚΜΗΡΙΩΣΗ (DOCUMENTATION) v1.0**

**ΑΤΟΜΙΚΗ ΕΠΙΜΕΛΕΙΑ ΕΡΓΑΣΙΑΣ:**  
ΧΡΗΣΤΟΣ-ΠΑΝΑΓΙΩΤΗΣ ΜΠΑΛΑΤΣΟΥΡΑΣ, (Α.Μ. : 1054335),  
Τρέχον έτος σπουδών: 4<sup>ov</sup>  
[email: [balatsouras@ceid.upatras.gr](mailto:balatsouras@ceid.upatras.gr)]

Κονιάκος Φωκίδος, Ιούνιος 2021

## Πίνακας περιεχομένων

1. ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ:.....	3
2. ΕΙΣΑΓΩΓΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ .....	3
3. ΕΚΘΕΣΗ ΚΑΤΑΣΤΑΣΗΣ ΥΛΟΠΟΙΗΣΗΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ΤΜΗΜΑΤΩΝ ΤΗΣ ΕΡΓΑΣΙΑΣ .....	3
4. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ .....	4
4.1 ΛΗΨΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ .....	5
4.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΕΞΑΓΩΓΗ ΓΡΑΦΗΜΑΤΩΝ .....	5
4.3 ΑΠΟΘΗΚΕΥΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ .....	6
4.4 ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΜΟΡΦΗ CSV .....	6
5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ .....	7
6. ΤΑ ΖΗΤΟΥΜΕΝΑ ΓΡΑΦΗΜΑΤΑ.....	9
7. Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ ΓΛΩΣΣΑ ΡΥΘΜΟΝ .....	11

## 1. ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ:

Όνοματεπώνυμο: Χρήστος – Παναγιώτης Μπαλατσούρας

Αριθμός Μητρώου: 1054335

Τρέχον έτος σπουδών: 4<sup>ο</sup>

email: [balatsouras@ceid.upatras.gr](mailto:balatsouras@ceid.upatras.gr)

## 2. ΕΙΣΑΓΩΓΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Έκδοση Python: 3.8.8

Λειτουργικό Σύστημα Υλοποίησης και Δοκιμής: Microsoft Windows 10, 20H2

Γλώσσα τεκμηρίωσης: Ελληνικά

Γλώσσα των μηνυμάτων της εφαρμογής και των γραφημάτων: Αγγλικά

Χρονικό Διάστημα Υλοποίησης: Ιούνιος 2021

Σκοπός της εργασίας είναι η εξοικείωση με τη γλώσσα προγραμματισμού Python. Αρχικά γίνεται λήψη ορισμένων δεδομένων σχετικά με τον τουρισμό στην Ευρώπη σε μορφή Microsoft Excel και στη συνέχεια γίνεται επεξεργασία των δεδομένων με σκοπό την εξαγωγή των ζητούμενων γραφημάτων από την εκφώνηση. Έπειτα, τα δεδομένα αποθηκεύονται σε μια βάση δεδομένων MySQL και εξάγονται σε αρχεία CSV.

## 3. ΕΚΘΕΣΗ ΚΑΤΑΣΤΑΣΗΣ ΥΛΟΠΟΙΗΣΗΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ΤΜΗΜΑΤΩΝ ΤΗΣ ΕΡΓΑΣΙΑΣ

Παρακάτω, παρουσιάζεται ο πίνακας με την κατάσταση της υλοποίησης των τμημάτων της εργασίας.

A/A	ΕΡΓΑΣΙΑ	ΥΛΟΠΟΙΗΣΗ	ΛΕΙΤΟΥΡΓΙΑ	ΠΑΡΑΤΗΡΗΣΕΙΣ
1	ΛΗΨΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ	ΝΑΙ	ΝΑΙ	
2	ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ	ΝΑΙ	ΝΑΙ	
3	ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΒΑΣΗ MySQL	ΝΑΙ	ΝΑΙ	<u>Η βάση δεδομένων πρέπει να έχει δημιουργηθεί πριν από την εκτέλεση της εφαρμογής.</u>
4	ΟΠΤΙΚΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ	ΝΑΙ	ΝΑΙ	
5	ΕΞΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΑΡΧΕΙΑ CSV	ΝΑΙ	ΝΑΙ	

Πίνακας 1 – Τμήματα Υλοποίησης της Εργασίας

## 4. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η συγκεκριμένη εργασία έχει σκοπό την επεξεργασία δεδομένων από τη Eurostat με σκοπό την εξαγωγή των ακόλουθων γραφημάτων:

- Nights spent at tourist accommodation establishments.
- Nights spent by non-residents at tourist accommodation establishments.
- Arrivals at tourist accommodation establishments.
- Arrivals of non-residents at tourist accommodation establishments.

Τα δεδομένα που θα παρουσιάζονται στα παραπάνω γραφήματα, έχουν ως εξής:

**ΠΡΩΤΗ ΧΩΡΑ ΕΝΔΙΑΦΕΡΟΝΤΟΣ: ΕΛΛΑΔΑ**

**ΔΕΥΤΕΡΗ ΧΩΡΑ ΕΝΔΙΑΦΕΡΟΝΤΟΣ: ΣΟΥΗΔΙΑ**

**ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ ΕΝΔΙΑΦΕΡΟΝΤΟΣ: Η τετραετία (2016, 2017, 2018, 2019)**

### ΠΡΟΕΛΕΥΣΗ ΔΕΔΟΜΕΝΩΝ:

- Nights spent at tourist accommodation establishments. (Σύνδεσμος: <https://ec.europa.eu/eurostat/databrowser/bookmark/70974cf5-72e1-489b-b56d-092cbaa59a00?lang=en>)
- Nights spent by non-residents at tourist accommodation establishments. (Σύνδεσμος: <https://ec.europa.eu/eurostat/databrowser/bookmark/ad9f3a06-oadc-4ffb-9a00-dadf9386c99c?lang=en>)
- Arrivals at tourist accommodation establishments (Σύνδεσμος: <https://ec.europa.eu/eurostat/databrowser/bookmark/ed4b2617-4f9d-44b4-84d8-83foo4c060ca?lang=en>).
- Arrivals of non-residents at tourist accommodation establishments (Σύνδεσμος: <https://ec.europa.eu/eurostat/databrowser/bookmark/a2079f11-e948-4318-b543-bd1f630524ed?lang=en>).

### Σχεδιαστική παραδοχή:

Ουσιαστικά, χρησιμοποιήθηκαν τα Datasets «Nights spent at tourist accommodation establishments by residents/non-residents» και « Arrivals of residents/non-residents at tourist accommodation establishments», ενώ η παραμετροποίηση «residents/non-residents» έγινε από το φίλτρο «Country of Residence» που δίνει τις επιλογές «Reporting Country/Foreign Country». Με την επιλογή «Reporting Country» ορίζονται οι “residents” και με την επιλογή «Foreign Country» ορίζονται οι “non-residents”.

Τα παραπάνω δεδομένα αρχικά λαμβάνονται από την ιστοσελίδα της Eurostat σε μορφή Microsoft Excel με τη χρήση της γλώσσας προγραμματισμού python. Στη συνέχεια εξάγονται τα απαραίτητα δεδομένα για τις χώρες και τα έτη ενδιαφέροντος και δημιουργούνται τα γραφήματα με τη χρήση της γλώσσας προγραμματισμού Python. Τέλος, τα δεδομένα εξάγονται σε μορφή CSV και αποθηκεύονται σε μία βάση δεδομένων MySQL.

#### 4.1 ΛΗΨΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ

Η λήψη των δεδομένων από τη Eurostat, γίνεται με τη χρήση του module requests της γλώσσας python. Τα δεδομένα λαμβάνονται ως αρχεία Microsoft Excel με τη χρήση του συνδέσμου για λήψη των δεδομένων σε excel από τη Eurostat και μέσω της συνάρτησης requests.get(). Στον κώδικα της άσκησης έχει υλοποιηθεί μια συνάρτηση η οποία εκτελεί τη λήψη αυτών των αρχείων καλώντας τη συνάρτηση requests.get() για κάθε url που παρέχεται.

Επιπλέον, παρατηρήθηκε ότι τα URL για λήψη των δεδομένων σε Excel που παρέχει η Eurostat έχουν περιορισμένο χρονικό διάστημα που είναι ενεργά. Συνεπώς εάν κατά τη διάρκεια που δοκιμάζετε την εφαρμογή δεν λειτουργούν οι σύνδεσμοι λήψης των δεδομένων, παρακαλώ χρησιμοποιήστε τα ήδη έτοιμα αρχεία excel που έχουν υποβληθεί με την εργασία.

#### 4.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΕΞΑΓΩΓΗ ΓΡΑΦΗΜΑΤΩΝ

Η επεξεργασία των δεδομένων από τα αρχεία excel, γίνεται με τη χρήση της βιβλιοθήκης openpyxl. Έχει υλοποιηθεί στον κώδικα μια συνάρτηση read\_excel\_data() η οποία λαμβάνει ως όρισμα το αρχείο excel και εξάγει τα αντίστοιχα κελιά δεδομένων από αυτό.

Επίσης, για την αποθήκευση μόνο των επιθυμητών δεδομένων από τα αρχεία, έχει υλοποιηθεί η κλάση Dataset. Η κλάση αυτή έχει ορίσματα για τις χώρες ενδιαφέροντος, τα χρόνια ενδιαφέροντος και για τις λίστες για την αποθήκευση των τιμών που εξάγονται.

Τα γραφήματα δημιουργούνται με τη χρήση της βιβλιοθήκης Matplotlib. Κάθε γράφημα δημιουργείται με την κλήση της συνάρτησης plot() που έχει υλοποιηθεί.

### 4.3 ΑΠΟΘΗΚΕΥΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Για την αποθήκευση των δεδομένων έχει υλοποιηθεί μια βάση δεδομένων με όνομα «1054335\_python\_project\_2021». Η τεχνολογία της βάσης δεδομένων είναι **mySQL** και αναπτύχθηκε μέσω του περιβάλλοντος **phpMyAdmin**. Η βάση αυτή περιέχει έναν πίνακα για κάθε γράφημα που εξάγεται.

Το σχήμα της βάσης είναι το παρακάτω:

1054335_python_project_2021_schema	
<div><div>1054335_python_project_2021 nights_spent</div><div><div>id : int(11)</div><div>country : varchar(50)</div><div>year : year(4)</div><div># nights_spent : int(11)</div></div></div>	<div><div>1054335_python_project_2021 nights_spent_non_residents</div><div><div>id : int(11)</div><div>country : varchar(50)</div><div>year : year(4)</div><div># nights_spent : int(11)</div></div></div>
<div><div>1054335_python_project_2021 arrivals</div><div><div>id : int(11)</div><div>country : varchar(50)</div><div>year : year(4)</div><div># arrivals : int(11)</div></div></div>	<div><div>1054335_python_project_2021 arrivals_non_residents</div><div><div>id : int(11)</div><div>country : varchar(11)</div><div>year : year(4)</div><div># arrivals : int(11)</div></div></div>

Κατά παραδοχή, η βάση δεδομένων πρέπει να έχει δημιουργηθεί και όλοι οι πίνακες να είναι άδειοι πριν την εκτέλεση της εφαρμογής της εργασίας. Η εφαρμογή που υλοποιήθηκε σε python μόνο αποθηκεύει τα δεδομένα στη βάση.

Για την επικοινωνία με τη βάση δεδομένων σε mySQL, χρησιμοποιήθηκε η βιβλιοθήκη `mysql.connector` της python.

### 4.4 ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΜΟΡΦΗ CSV

Για την αποθήκευση σε αρχείο μορφής csv χρησιμοποιήθηκε η βιβλιοθήκη `csv` της python.

Για κάθε αρχείο, αρχικά καλείται η συνάρτηση `writerow()` για να εκτυπώσει τις επικεφαλίδες στην πρώτη γραμμή του αρχείου και μετά καλείται επαναληπτικά η `writerow()` για να εκτυπώσει τα δεδομένα για κάθε κατηγορία γραφήματος στο αντίστοιχο αρχείο.

## 5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

Κατά την εκτέλεση της εφαρμογής, στο τερματικό εμφανίζεται η παρακάτω έξοδος:

```
Principles of Programming Languages and Compiler Design
Python Project 2021
Author: Christos-Panagiotis Mpalatsouras, StudentID = 1054335

This script extracts information about tourism in Europe
First Country of Interest: Greece
Second Country of Interest: Sweden
Years of interest: [2016, 2017, 2018, 2019]
Requested plots:
1. Nights spent at tourist accommodation establishments
2. Nights spent by non-residents at tourist accommodation establishments
3. Arrivals at tourist accommodation establishments
4. Arrivals of non-residents at tourist accommodation establishments

This Script Will Download Required Data from Eurostat Website
Downloading: Nights spent at tourist accommodation establishments by residents, from URL: https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LTVE/xlsx/en/download/26198e8b-f8c7-404c-948f-2c24fb85ca8c
File Downloaded Successfully
Downloading: Nights spent at tourist accommodation establishments by non-residents, from URL: https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LTVE/xlsx/en/download/1fd52f3-abc1-613b-b1c9-3e8f93ab133a
File Downloaded Successfully
Downloading: Arrivals of residents at tourist accommodation establishments, from URL: https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LTVE/xlsx/en/download/363a8169-aae7-46b5-b8d2-7e14568db868
File Downloaded Successfully
Downloading: Arrivals of non-residents at tourist accommodation establishments, from URL: https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LTVE/xlsx/en/download/7ff88378-1f46-4644-82fd-89a35058e7df
File Downloaded Successfully

Download Completed Successfully!
Downloaded 4 files.

Extracting useful information from the downloaded files...
Plotting the data...
Writing to database: Nights spent by residents
Writing to database: Nights spent by non residents
Writing to database: Arrivals by residents
Writing to database: Arrivals by non residents
Extracting Nights Spent by Residents in CSV...
Extracting Nights Spent by Non Residents in CSV...
Extracting Arrivals by Residents in CSV...
Extracting Arrivals by Non Residents in CSV...

Process finished with exit code 0
```

Επίσης εμφανίζονται τα ζητούμενα γραφήματα μέσω της Matplotlib. Τα γραφήματα παρουσιάζονται σε επόμενη ενότητα αυτού του εγγράφου.

Το σχήμα της βάσης δεδομένων παρουσιάζεται στην προηγούμενη σελίδα.

Μετά την εκτέλεση της εφαρμογής, έχουν αποθηκευτεί τα δεδομένα που εξήχθησαν στους αντίστοιχους πίνακες της βάσης δεδομένων, όπως φαίνεται παρακάτω:

### Πίνακας «nights\_spent»:

























			id	country	year	nights_spent
			ID	Country	Year	Nights spent at tourist accommodation establishmen...
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			1	Greece	2016	22107192
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			2	Greece	2017	21974593
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			3	Greece	2018	24064088
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			4	Greece	2019	23623077
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			5	Sweden	2016	43236883
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			6	Sweden	2017	44044499
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			7	Sweden	2018	45171193
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			8	Sweden	2019	47195047



























### Πίνακας «nights\_spent\_non\_residents»:

			id	country	year	nights_spent
			ID	Country	Year	Nights spent at tourist accommodation establishmen...
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			1	Greece	2016	87912850
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			2	Greece	2017	97034421
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			3	Greece	2018	118876323
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			4	Greece	2019	119971390
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			5	Sweden	2016	13997265
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			6	Sweden	2017	14638702
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			7	Sweden	2018	15685681
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			8	Sweden	2019	15980234

### Πίνακας «arrivals»:

			id	country	year	arrivals
			ID	Country	Year	Arrivals of residents at tourist accommodation est...
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			1	Greece	2016	8080042
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			2	Greece	2017	8142571
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			3	Greece	2018	9264746
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			4	Greece	2019	9163555
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			5	Sweden	2016	22524467
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			6	Sweden	2017	23033004
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			7	Sweden	2018	23519609
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			8	Sweden	2019	24490475

### Πίνακας «arrivals\_non\_residents»:

			id	country	year	arrivals
			ID	Country	Year	Arrivals of non-residents at tourist accommodation...
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			1	Greece	2016	16915996
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			2	Greece	2017	19068697
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			3	Greece	2018	24320893
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			4	Greece	2019	25038498
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			5	Sweden	2016	6550337
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			6	Sweden	2017	6840963
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			7	Sweden	2018	7217241
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
			8	Sweden	2019	7407227

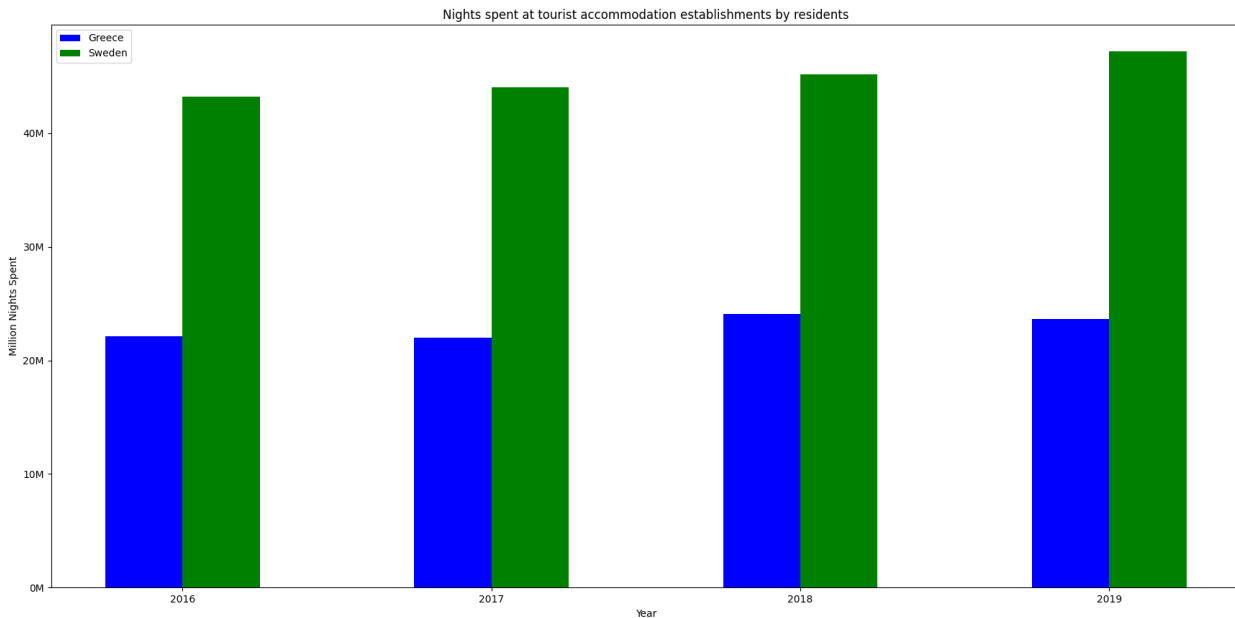
Επίσης, κατά την εκτέλεση της εφαρμογής εξάγονται τα δεδομένα σε μορφή CSV, στα αντίστοιχα αρχεία που περιλαμβάνονται στην υποβολή της εργασίας.



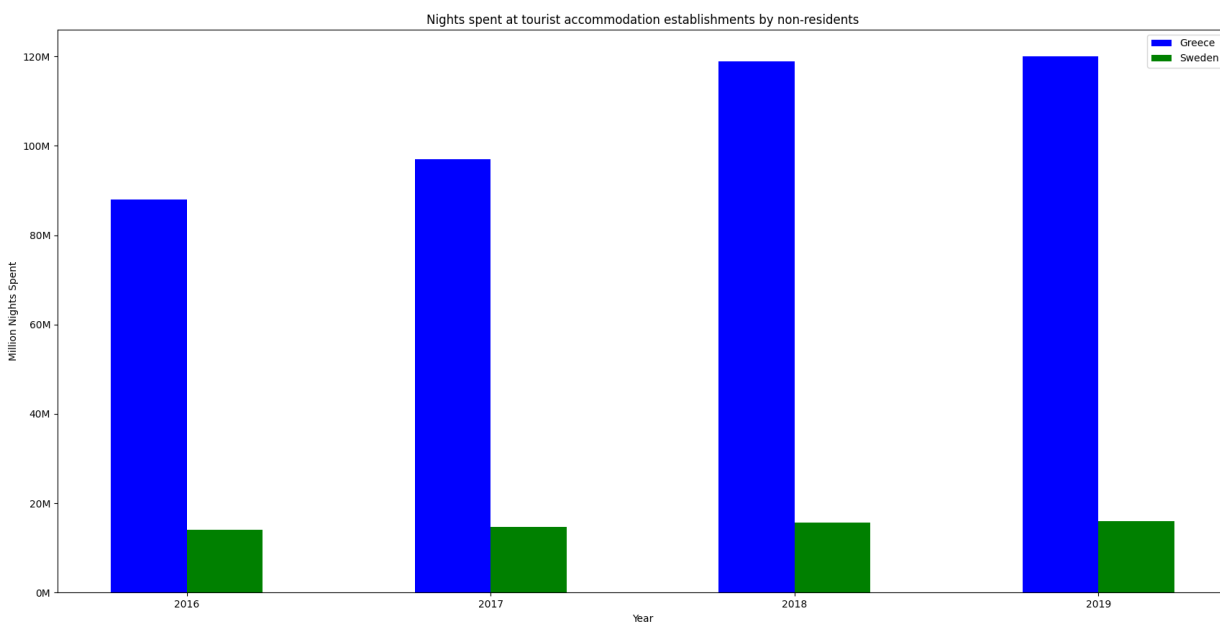
## 6. ΤΑ ΖΗΤΟΥΜΕΝΑ ΓΡΑΦΗΜΑΤΑ

Σε αυτή την ενότητα παρουσιάζονται τα γραφήματα που ζητείται να εξαχθούν σύμφωνα με την εκφώνηση με τίτλο και υπόμνημα.

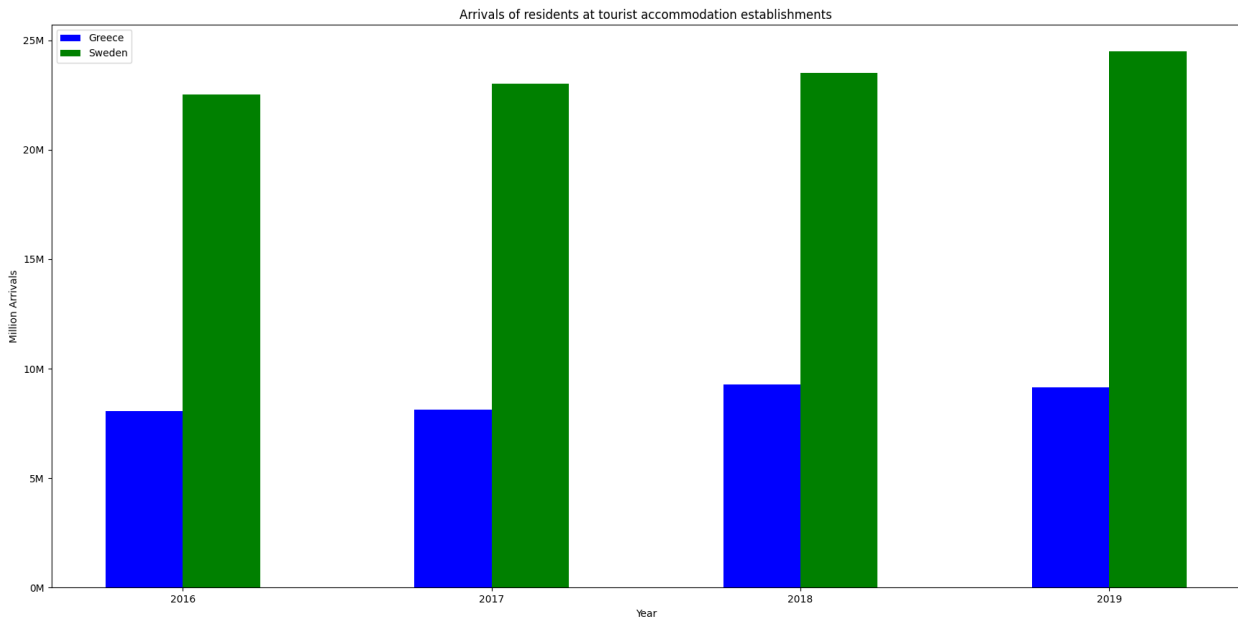
**Γράφημα «Nights spent at tourist accommodation establishments»:**



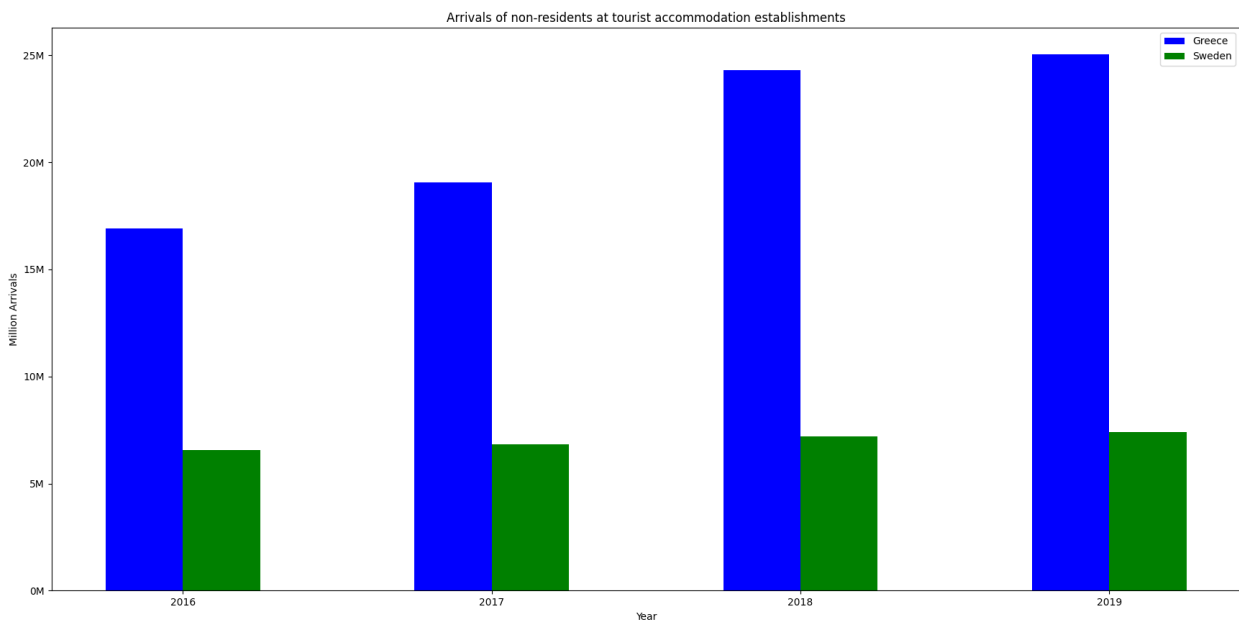
**Γράφημα «Nights spent by non-residents at tourist accommodation establishments»:**



## Γράφημα «Arrivals at tourist accommodation establishments»:



## Γράφημα «Arrivals of non-residents at tourist accommodation establishments»:



## 7. Ο ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ ΓΛΩΣΣΑ PYTHON

Σε αυτή την ενότητα, παρουσιάζεται ο κώδικας της εφαρμογής που υλοποιήθηκε σε γλώσσα προγραμματισμού python για τους σκοπούς της εργασίας.

```
# Principles of Programming Languages and Compiler Design (CEID - University of
Patras)
# Python Project 2021
# Christos-Panagiotis Balatsouras, StudentID: 1054335

# import python libraries
import requests
from openpyxl import load_workbook
from matplotlib.ticker import FuncFormatter
import matplotlib.pyplot as plt
import numpy as np
import warnings
import mysql.connector
import csv

# Global Variables
files_count = 0
Country_1 = "Greece"
Country_2 = "Sweden"
Years = [2016, 2017, 2018, 2019]
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="1054335_python_project_2021"
)

class Dataset:
    def __init__(self, country1, country2, years):
        self.first_country = country1
        self.second_country = country2
        self.data1 = []
        self.data2 = []
        self.interval = years

def download(files_downloaded_count): # download data in .xlsx format from the WEB
    # URLs
    url_nights_spent_residents = "https://ec.europa.eu/eurostat/databrowser-
backend/api/query/1.0/LIVE/xlsx/en/download/24190e8b-f8c7-404c-960f-2c24fb85ca8c"
    url_nights_spent_non_residents = "https://ec.europa.eu/eurostat/databrowser-
backend/api/query/1.0/LIVE/xlsx/en/download/1fdc52f3-abc1-413b-b1c9-3e8f93ab133a"
    url_arrivals_residents = "https://ec.europa.eu/eurostat/databrowser-
backend/api/query/1.0/LIVE/xlsx/en/download/343a8149-aae7-46b5-b0d2-7e145484b060"
    url_arrivals_non_residents = "https://ec.europa.eu/eurostat/databrowser-
backend/api/query/1.0/LIVE/xlsx/en/download/7ff88378-1f46-4644-82fd-09a35058e7df"

    # Download process
    print("Downloading: Nights spent at tourist accommodation establishments by
residents, from URL: ", url_nights_spent_residents)
    req_nights_spent_residents = requests.get(url_nights_spent_residents,
allow_redirects=True)
    open('nights_spent_by_residents.xlsx',
```

```
'wb').write(req_nights_spent_residents.content)
print("File Downloaded Successfully")
files_downloaded_count += 1

print("Downloading: Nights spent at tourist accommodation establishments by non-
residents, from URL: ", url_nights_spent_non_residents)
req_nights_spent_non_residents = requests.get(url_nights_spent_non_residents,
allow_redirects=True)
open('nights_spent_by_non_residents.xlsx',
'wb').write(req_nights_spent_non_residents.content)
print("File Downloaded Successfully")
files_downloaded_count += 1

print("Downloading: Arrivals of residents at tourist accommodation establishments,
from URL: ", url_arrivals_residents)
req_arrivals_residents = requests.get(url_arrivals_residents,
allow_redirects=True)
open('arrivals_by_residents.xlsx', 'wb').write(req_arrivals_residents.content)
print("File Downloaded Successfully")
files_downloaded_count += 1

print("Downloading: Arrivals of non-residents at tourist accommodation
establishments, from URL: ", url_arrivals_non_residents)
req_arrivals_non_residents = requests.get(url_arrivals_non_residents,
allow_redirects=True)
open('arrivals_by_non_residents.xlsx',
'wb').write(req_arrivals_non_residents.content)
print("File Downloaded Successfully")
files_downloaded_count += 1

print("\nDownload Completed Successfully!\nDownloaded ", files_downloaded_count,
"files.\n")

def read_excel_data(file_name): # Extract Useful Information from Excel Files
    dataset = Dataset(Country_1, Country_2, Years)
    year_columns = []
    row1 = None
    row2 = None

    with warnings.catch_warnings(record=True):
        warnings.simplefilter("always")
        wb = load_workbook(file_name)
        sheet_1 = wb['Sheet 1']
        for row in sheet_1.iter_rows(min_row=10, max_col=25, max_row=53):
            for cell in row:
                if cell.value == Country_1:
                    row1 = cell.row
        for row in sheet_1.iter_rows(min_row=10, max_col=25, max_row=53):
            for cell in row:
                if cell.value == Country_2:
                    row2 = cell.row
        for row in sheet_1.iter_rows(min_row=10, max_col=25, max_row=10):
            for cell in row:
                for item in Years:
                    if cell.value == str(item):
                        col = cell.column
                        year_columns.append(col)

    numofyears = len(year_columns)
    # print(row1)
```

```
# print(row2)
# print(year_columns)

for row in sheet_1.iter_rows(min_col=year_columns[0], min_row=row1,
max_col=year_columns[numofyears-1], max_row=row1):
    for cell in row:
        if isinstance(cell.value, float):
            dataset.data1.append(int(cell.value))
# print(dataset.data1)

for row in sheet_1.iter_rows(min_col=year_columns[0], min_row=row2,
max_col=year_columns[numofyears-1], max_row=row2):
    for cell in row:
        if isinstance(cell.value, float):
            dataset.data2.append(int(cell.value))
# print(dataset.data2)

return dataset

def format_millions(x, pos): # Function for the formatting of the numbers in the plot
    return '%1.0fM' % (x * 1e-6)

def plot(country1, country2, years, data1, data2, xlabel, ylabel, title):
    x_pos = np.arange(len(years))
    width = 0.25 # bar width

    formatter = FuncFormatter(format_millions)

    fig, ax = plt.subplots()
    r1 = ax.bar(x_pos - width / 2, data1, width, label=country1, color='b')
    r2 = ax.bar(x_pos + width / 2, data2, width, label=country2, color='g')

    ax.set_ylabel(ylabel)
    ax.set_xlabel(xlabel)
    ax.set_title(title)
    ax.set_xticks(x_pos)
    ax.yaxis.set_major_formatter(formatter)
    ax.set_xticklabels(years)
    ax.legend()

    fig.tight_layout()
    plt.show()

def main():
    print("Principles of Programming Languages and Compiler Design")
    print("Python Project 2021")
    print("Author: Christos-Panagiotis Mpalatsouras, SudentID = 1054335\n")
    print("This script extracts information about tourism in Europe")
    print("First Country of Interest: ", Country_1)
    print("Second Country of Interest: ", Country_2)
    print("Years of interest: ", Years)

    print("Requested plots: ")
    print("1. Nights spent at tourist accommodation establishments")
    print("2. Nights spent by non-residents at tourist accommodation establishments")
    print("3. Arrivals at tourist accommodation establishments")
    print("4. Arrivals of non-residents at tourist accommodation establishments")
```

```
print("\nThis Script Will Download Required Data from Eurostat Website")
download(files_count)

print("Extracting useful information from the downloaded files...")
Nights_spent_residents = read_excel_data('nights_spent_by_residents.xlsx')
Nights_spent_non_residents = read_excel_data('nights_spent_by_non_residents.xlsx')
Arrivals_residents = read_excel_data('arrivals_by_residents.xlsx')
Arrivals_non_residents = read_excel_data('arrivals_by_non_residents.xlsx')

# Plotting the data
print("Plotting the data...")
plot(Nights_spent_residents.first_country, Nights_spent_residents.second_country,
Years,
    Nights_spent_residents.data1, Nights_spent_residents.data2, "Year", "Million
Nights Spent",
    "Nights spent at tourist accommodation establishments by residents")
plot(Nights_spent_non_residents.first_country,
Nights_spent_non_residents.second_country, Years,
    Nights_spent_non_residents.data1, Nights_spent_non_residents.data2, "Year",
"Million Nights Spent",
    "Nights spent at tourist accommodation establishments by non-residents")
plot(Arrivals_residents.first_country, Arrivals_residents.second_country, Years,
    Arrivals_residents.data1, Arrivals_residents.data2, "Year", "Million
Arrivals",
    "Arrivals of residents at tourist accommodation establishments")
plot(Arrivals_non_residents.first_country, Arrivals_non_residents.second_country,
Years,
    Arrivals_non_residents.data1, Arrivals_non_residents.data2, "Year", "Million
Arrivals",
    "Arrivals of non-residents at tourist accommodation establishments")

# Writing data to Database
mycursor = mydb.cursor()
id = 1
increment = 0
print("Writing to database: Nights spent by residents")
query = "INSERT INTO nights_spent (id, country, year, nights_spent) VALUES (%s,
%s, %s, %s)"
for item in Nights_spent_residents.data1:
    val = (id, Country_1, Nights_spent_residents.interval[increment], item)
    mycursor.execute(query, val)
    mydb.commit()
    increment += 1
    id += 1
increment = 0
for item in Nights_spent_residents.data2:
    val = (id, Country_2, Nights_spent_residents.interval[increment], item)
    mycursor.execute(query, val)
    mydb.commit()
    increment += 1
    id += 1
increment = 0
id = 1
print("Writing to database: Nights spent by non residents")
query = "INSERT INTO nights_spent_non_residents (id, country, year, nights_spent)
VALUES (%s, %s, %s, %s)"
for item in Nights_spent_non_residents.data1:
    val = (id, Country_1, Nights_spent_non_residents.interval[increment], item)
    mycursor.execute(query, val)
    mydb.commit()
    increment += 1
```



```
        id += 1
    increment = 0
    for item in Nights_spent_non_residents.data2:
        val = (id, Country_2, Nights_spent_non_residents.interval[increment], item)
        mycursor.execute(query, val)
        mydb.commit()
        increment += 1
        id += 1
    increment = 0
    id = 1
    print("Writing to database: Arrivals by residents")
    query = "INSERT INTO arrivals (id, country, year, arrivals) VALUES (%s, %s, %s, %s)"
    for item in Arrivals_residents.data1:
        val = (id, Country_1, Arrivals_residents.interval[increment], item)
        mycursor.execute(query, val)
        mydb.commit()
        increment += 1
        id += 1
    increment = 0
    for item in Arrivals_residents.data2:
        val = (id, Country_2, Arrivals_residents.interval[increment], item)
        mycursor.execute(query, val)
        mydb.commit()
        increment += 1
        id += 1
    increment = 0
    id = 1
    print("Writing to database: Arrivals by non residents")
    query = "INSERT INTO arrivals_non_residents (id, country, year, arrivals) VALUES (%s, %s, %s, %s)"
    for item in Arrivals_non_residents.data1:
        val = (id, Country_1, Arrivals_non_residents.interval[increment], item)
        mycursor.execute(query, val)
        mydb.commit()
        increment += 1
        id += 1
    increment = 0
    for item in Arrivals_non_residents.data2:
        val = (id, Country_2, Arrivals_non_residents.interval[increment], item)
        mycursor.execute(query, val)
        mydb.commit()
        increment += 1
        id += 1

    # Write CSV Files
    with open('nights_spent_residents_file.csv', mode='w', newline='') as nights_spent_residents_file:
        print("Extracting Nights Spent by Residents in CSV...")
        nights_spent_residents_writer = csv.writer(nights_spent_residents_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_NONNUMERIC)
        index = 0
        nights_spent_residents_writer.writerow(['Country', 'Year', 'Nights_spent'])
        for item in Nights_spent_residents.data1:
            nights_spent_residents_writer.writerow([Country_1, Nights_spent_residents.interval[index], item])
            index += 1
        index = 0
        for item in Nights_spent_residents.data2:
            nights_spent_residents_writer.writerow([Country_2, Nights_spent_residents.interval[index], item])
```

```
        index += 1
    nights_spent_residents_file.close()

    with open('nights_spent_non_residents_file.csv', mode='w', newline='') as
nights_spent_non_residents_file:
        print("Extracting Nights Spent by Non Residents in CSV...")
        nights_spent_non_residents_writer =
csv.writer(nights_spent_non_residents_file, delimiter=',', quotechar='"',
quoting=csv.QUOTE_NONNUMERIC)
        index = 0
        nights_spent_non_residents_writer.writerow(['Country', 'Year',
'Nights_spent'])
        for item in Nights_spent_non_residents.data1:
            nights_spent_non_residents_writer.writerow([Country_1,
Nights_spent_non_residents.interval[index], item])
            index += 1
        index = 0
        for item in Nights_spent_non_residents.data2:
            nights_spent_non_residents_writer.writerow([Country_2,
Nights_spent_non_residents.interval[index], item])
            index += 1
    nights_spent_non_residents_file.close()

    with open('arrivals_residents_file.csv', mode='w', newline='') as
arrivals_residents_file:
        print("Extracting Arrivals by Residents in CSV...")
        arrivals_residents_writer = csv.writer(arrivals_residents_file, delimiter=',',
quotechar='"', quoting=csv.QUOTE_NONNUMERIC)
        index = 0
        arrivals_residents_writer.writerow(['Country', 'Year', 'Arrivals'])
        for item in Arrivals_residents.data1:
            arrivals_residents_writer.writerow([Country_1,
Arrivals_residents.interval[index], item])
            index += 1
        index = 0
        for item in Arrivals_residents.data2:
            arrivals_residents_writer.writerow([Country_2,
Arrivals_residents.interval[index], item])
            index += 1
    arrivals_residents_file.close()

    with open('arrivals_non_residents_file.csv', mode='w', newline='') as
arrivals_non_residents_file:
        print("Extracting Arrivals by Non Residents in CSV...")
        arrivals_non_residents_writer = csv.writer(arrivals_non_residents_file,
delimiter=',', quotechar='"', quoting=csv.QUOTE_NONNUMERIC)
        index = 0
        arrivals_non_residents_writer.writerow(['Country', 'Year', 'Arrivals'])
        for item in Arrivals_non_residents.data1:
            arrivals_non_residents_writer.writerow([Country_1,
Arrivals_non_residents.interval[index], item])
            index += 1
        index = 0
        for item in Arrivals_non_residents.data2:
            arrivals_non_residents_writer.writerow([Country_2,
Arrivals_non_residents.interval[index], item])
            index += 1
    arrivals_non_residents_file.close()

main()
```