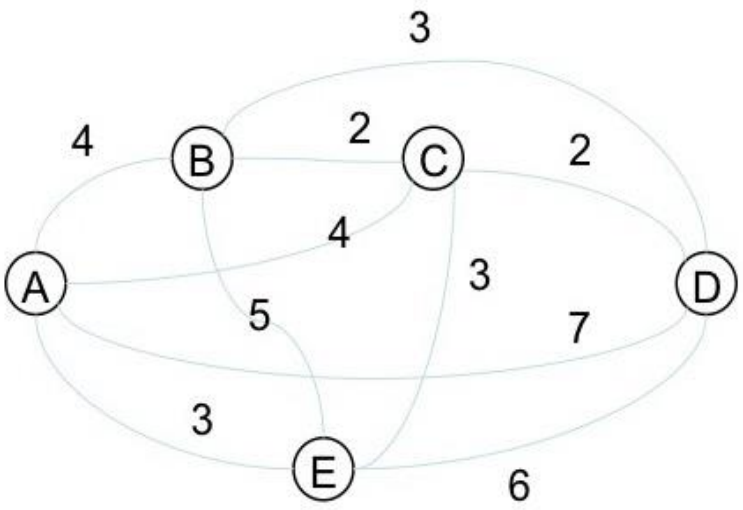
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος ***Τεχνητή νοημοσύνη και έμπειρα συστήματα***

Αριθμός εργασίας – Τίτλος εργασίας	***Προαιρετική ατομική εργασία***
Όνομα φοιτητή	Καλλίγερος Αναστάσης
Αρ. Μητρώου	Π19253
Ημερομηνία παράδοσης	4/6/2022

Εκφώνηση εργασίας

Α. Αναπτύξτε πρόγραμμα επίλυσης του Traveling Salesman Problem με χρήση γενετικών αλγορίθμων και γλώσσα προγραμματισμού της επιλογής σας. Ο γράφος αποτελείται από πλήρως διασυνδεδεμένες πόλεις όπως φαίνεται στο παρακάτω σχήμα.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1 Εισαγωγή

Αρχικά, μετά από προσεκτική μελέτη της εκφώνησης αποφάσισα να υλοποιήσω το Α. θέμα της εργασίας σε γλώσσα προγραμματισμού python.Ως εργαλείο ανάπτυξης του κώδικα επέλεξα την πιο νέα και πληρέστερη έκδοση του Visual Studio.Σκοπός της εργασίας είναι η επίλυση του προβλήματος Traveling Salesman Problem όπου ο στόχος είναι να περάσει από κάθε πόλη και να δει τα έξοδα που έχει κάθε μια διαδρομή ανάμεσα στις πόλεις. Το συγκεκριμένο πρόβλημα θα το υλοποιήσουμε μέσω των αλγορίθμων που περιέχουν γονείς και αναπαραγωγή παιδιών μέσω τυχαίων τρόπων. Παρακάτω γίνεται αναλυτική επεξήγηση του κώδικα.

2 Περιγραφή του προγράμματος

Όταν ξεκινάει το πρόγραμμα μας βάζουμε τυχαίες διαδρομές με σκοπό να περνάει από κάθε πόλη, ξεκινώντας πάντα από την πρώτη και τελειώνοντας θα ξαναγυρίσεις στην πρώτη. Οπότε κάνοντας αυτή την διαδικασία βρίσκουμε κάποιες διαδρομές και ύστερα μπορούμε να βγάλουμε το κόστος όπου έχει η κάθε διαδρομή, όταν βγάζουμε το κόστος αντιστρέφουμε το αποτέλεσμα διότι θέλουμε η βέλτιστη διαδρομή να είναι και η μεγαλύτερη. Έχοντας όλα τα στοιχεία τα αποθηκεύουμε σε δύο ληστές trips και s όπου είναι οι διαδρομές και τα κόστη τους αντίστοιχα. Στην συγκεκριμένη διαδικασία έχουμε μια while στην οποία γίνεται ο έλεγχος για την εισαγωγή δεδομένων όπου στην περίπτωση που γίνεται η εισαγωγή των πρώτων δύο στοιχείων δεν γίνεται κανένας έλεγχος. Δηλαδή αν έχουν περάσει τα δύο πρώτα στοιχεία ξεκινάει ο έλεγχος για να μην περάσει από την ίδια πόλη δύο φορές. Αφού γίνει ο έλεγχος και περάσει από όλες τις πόλεις τότε ξεκινάει η διαδικασία για το κόστος που χρειάζεται για να γίνει η συγκεκριμένη διαδικασία. Εφόσον έχουμε και την τιμή του κόστους την αντιστρέφουμε και παίρναμε τις τιμές σε νέες λίστες όπου θα περιέχει όλες τις διαδρομές και όλα τα κόστη αντίστοιχα.

Αφού ολοκληρωθεί η διαδικασία με τις διαδρομές και τα κόστη και τα τυπώσουμε, αρχίζουμε τη διαδικασία για να βρούμε τυχαίους γονείς. Στην συγκεκριμένη διαδικασία παίρνουμε τυχαίες τιμές από την s (κόστη). Με αυτές τυχαίες τιμές συγκρίνουμε με τιμές ώστε να βρούμε ποιο στοιχείο είναι για να δούμε ποια διαδρομή είναι. Αφού βρούμε αυτό το στοιχείο περνάμε σε νέα λίστα (goneas) την διαδρομή την οποία βρήκαμε.

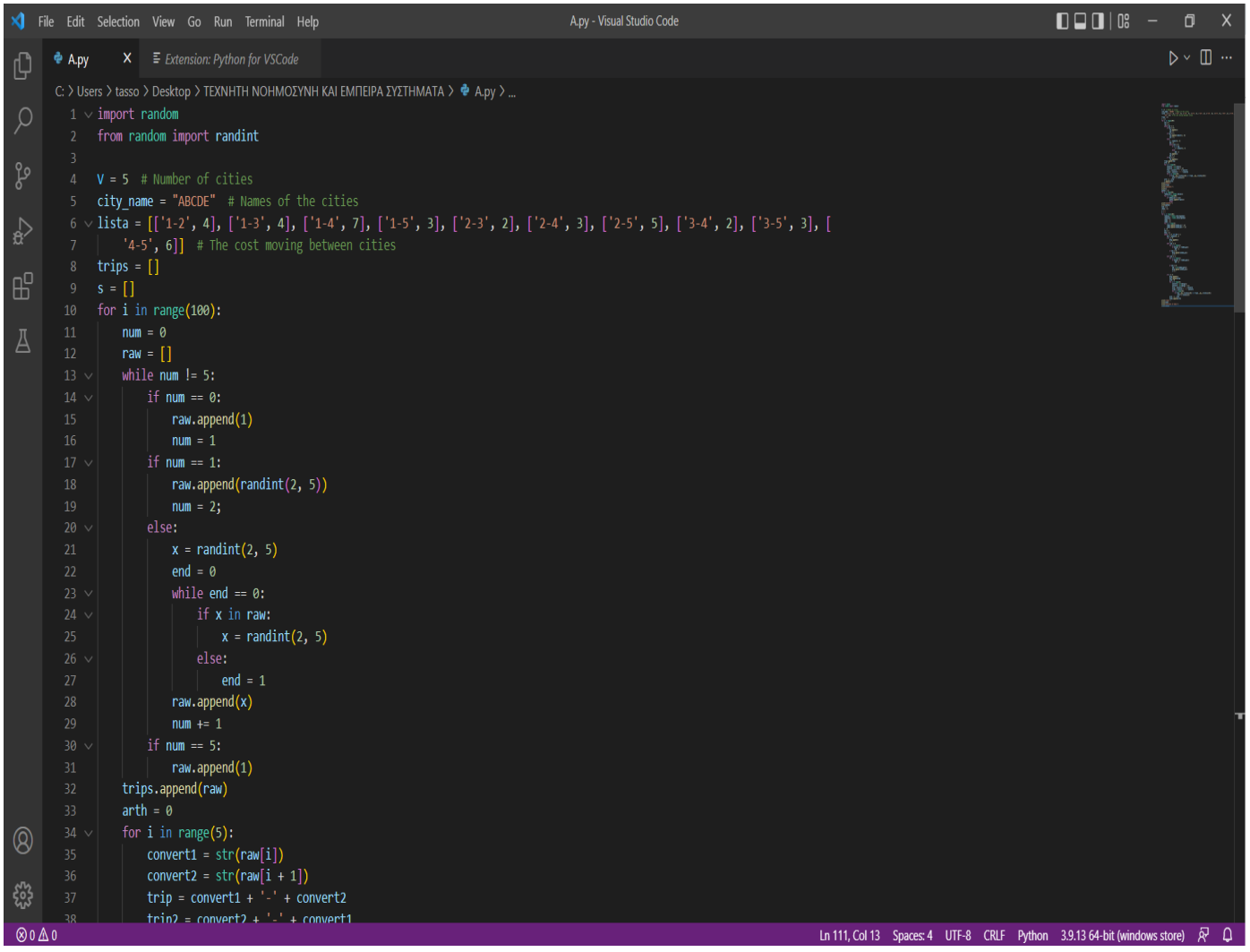
Στην διαδικασία που βγάζουμε τα παιδιά έχουμε δύο μεταβλητές την random_g και την random_g2, όπου διαλέγουν τυχαίους γονείς ώστε να πραγματοποιήσουν το κομμάτι της αναπαραγωγής, δεν χρειάζεται κάποιος έλεγχος αφού ένας γονέας μπορεί να κάνει και με

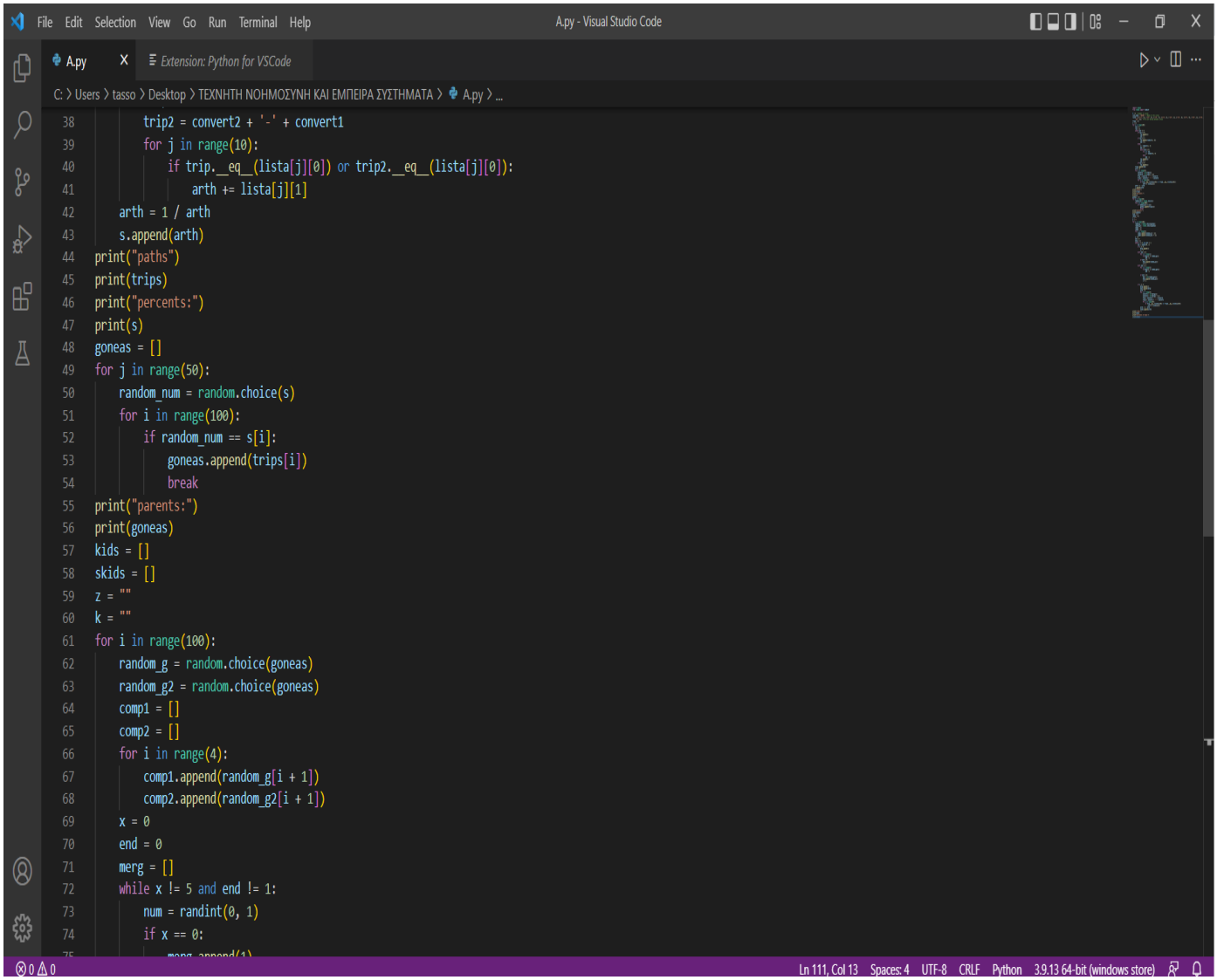
τον εαυτό του παιδί αλλά και να πραγματοποιηθεί η μετάλλαξη που είναι ένας γονέας να κάνει παιδί με το παιδί του. Δημιουργούμε μία επανάληψη ώστε να γίνει όλη η διαδικασία μέσα σε αυτή. Όπως ισχύει στην αρχική διαδρομή έτσι και στα παιδιά αρχίζει και τελειώνει πάντα στην πρώτη πόλη επομένως την αρχή γίνεται εισαγωγή δεδομένων κατευθείαν χωρίς κάποιον έλεγχο. Για τις υπόλοιπες πόλεις έχουμε τη μεταβλητή num, η οποία διαλέγει τυχαία ανάμεσα στους δύο γονείς. Δημιουργούμε μία άλλη επανάληψη η οποία κάνει έλεγχο αν το στοιχείο από το τυχαίο γονιό υπάρχει ήδη μέσα στη λίστα του παιδιού τότε το πρόγραμμα θα βγαίνει από την επανάληψη και θα ξαναρχίσει τη διαδικασία από την αρχή. Αν δεν ισχύει αυτό τότε κατευθείαν περνάει το νέο στοιχείο στην λίστα του παιδιού. Η ίδια η διαδικασία γίνεται και στους δύο γονείς, αν η μεταβλητή X όπου μετράμε πόσα στοιχεία είναι στην λίστα του παιδιού γίνει 5 τότε περνάει ένα τελευταίο στοιχείο στη λίστα του παιδιού, πού είναι η πρώτη πόλη και στη μεγαλύτερη λίστα που περιέχει όλα τα παιδιά περνάει και το συγκεκριμένο κι ξανακάνει την ίδια διαδικασία για να βγάλει το κόστος του παιδιού.

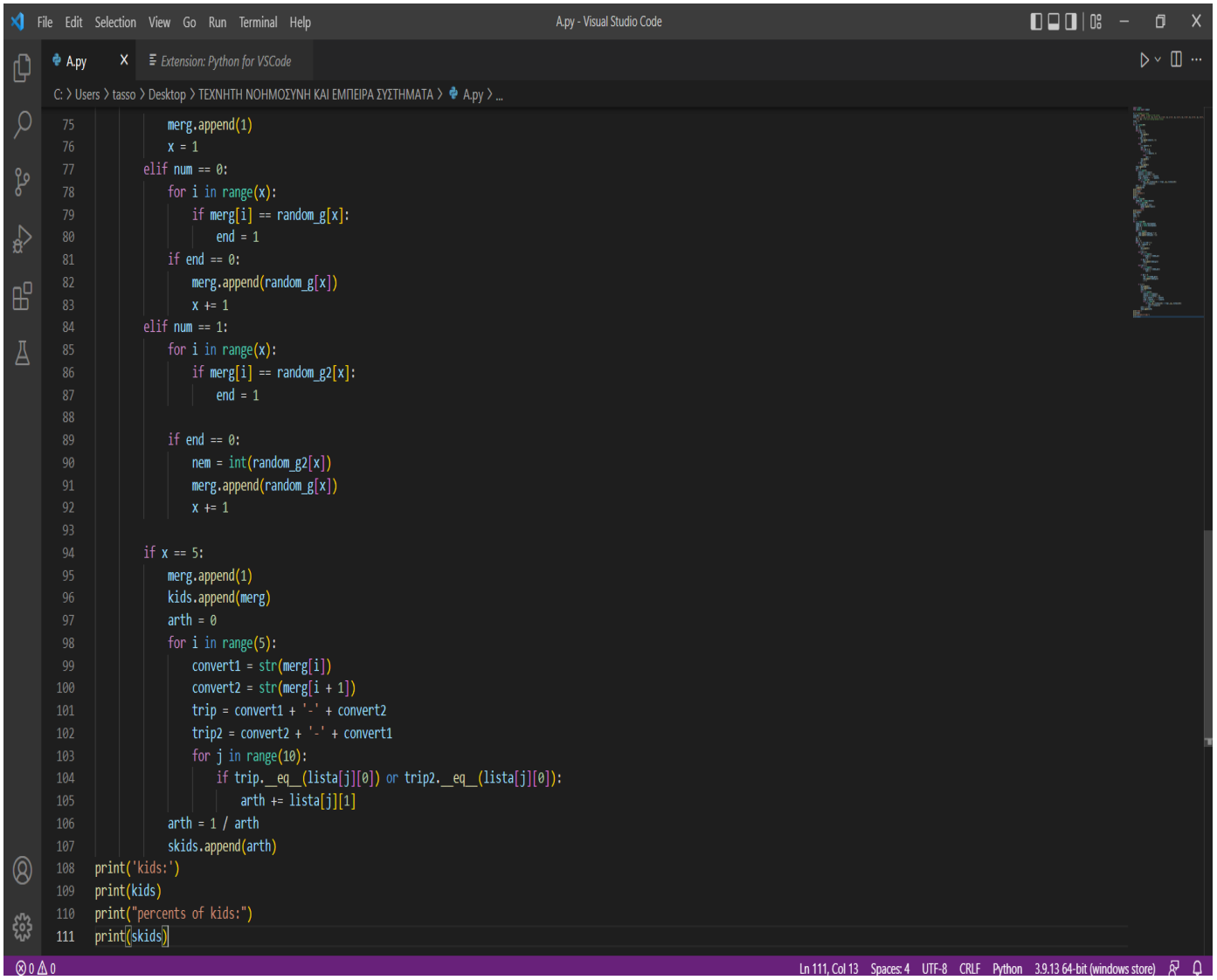
Έχοντας κάνει αυτήν την διαδικασία έχουμε ολοκληρώσει τον αλγόριθμο και έχουμε υλοποιήσει το ερώτημα και τις ανταλλακτικές τις οποίες μπορούν να δοθούν στην λύση του προβλήματος.

3 Επίδειξη της λύσης

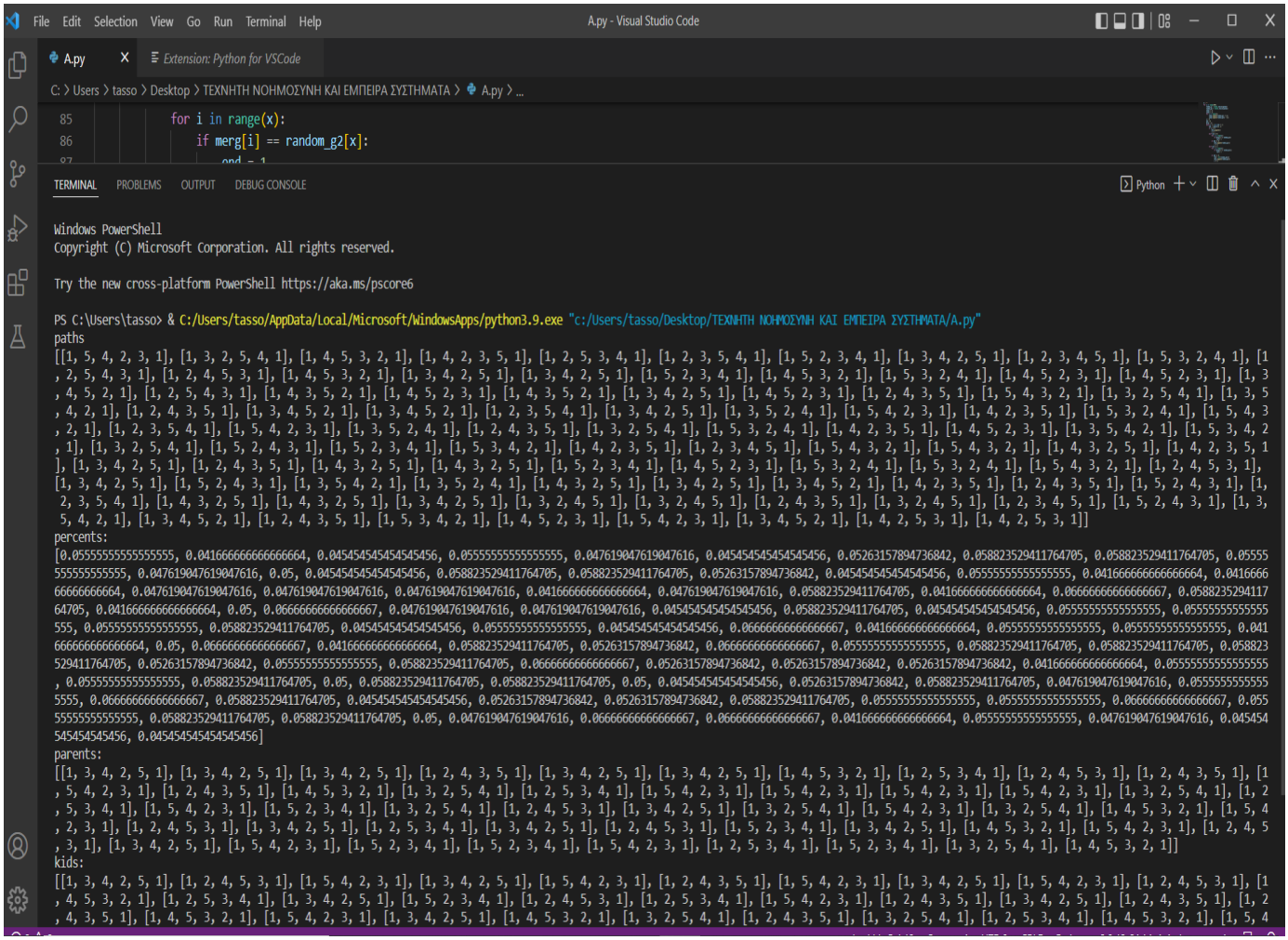
Παρουσίαση του κώδικα:

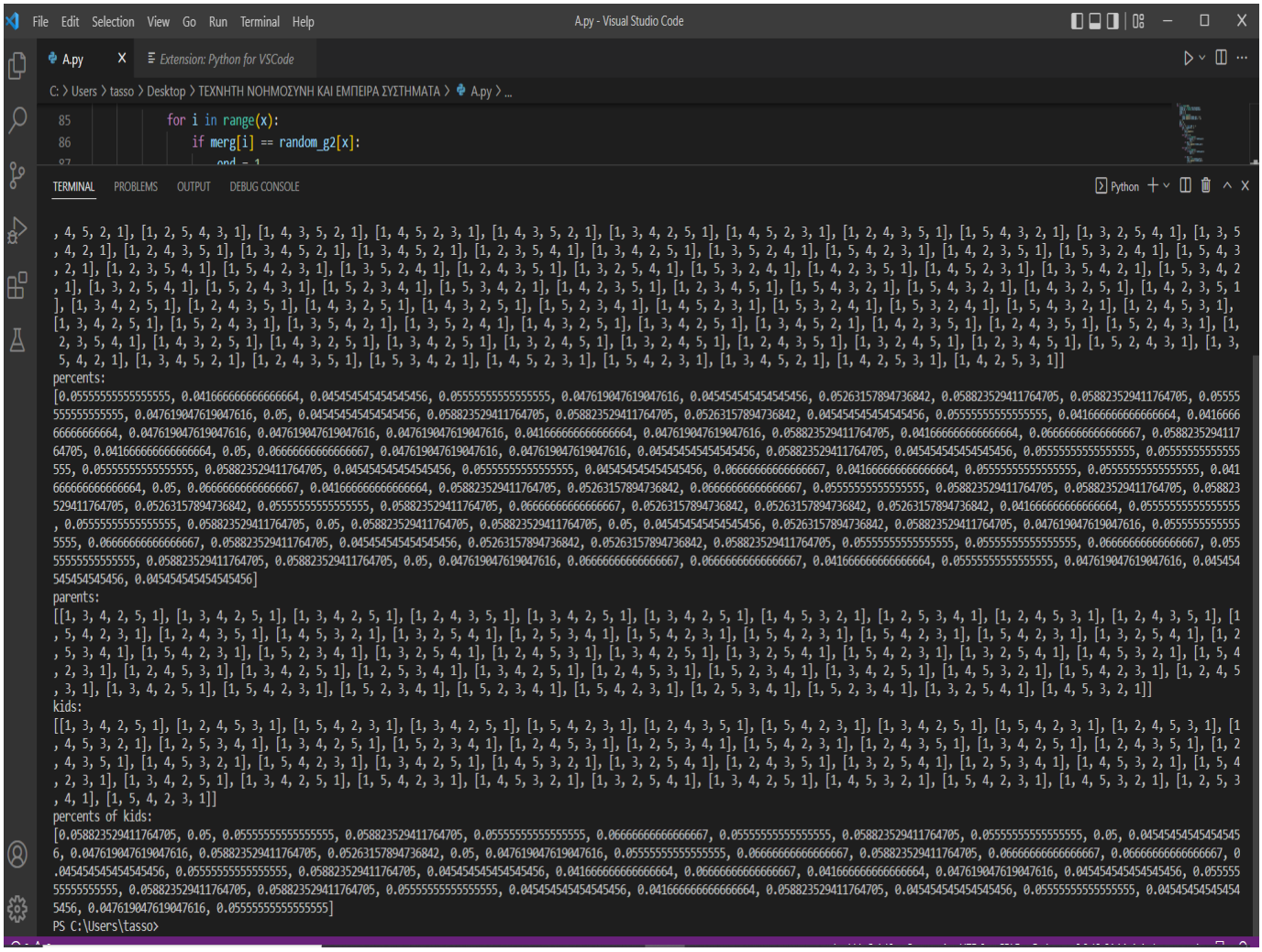



File Edit Selection View Go Run Terminal Help
A.py - Visual Studio Code
A.py
Extension: Python for VSCode
C: > Users > tasso > Desktop > ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ > A.py > ...
38 trip2 = convert2 + '-' + convert1
39 for j in range(10):
40 if trip.__eq__(lista[j][0]) or trip2.__eq__(lista[j][0]):
41 arth += lista[j][1]
42 arth = 1 / arth
43 s.append(arth)
44 print("paths")
45 print(trips)
46 print("percents:")
47 print(s)
48 goneas = []
49 for j in range(50):
50 random_num = random.choice(s)
51 for i in range(100):
52 if random_num == s[i]:
53 goneas.append(trips[i])
54 break
55 print("parents:")
56 print(goneas)
57 kids = []
58 skids = []
59 z = ""
60 k = ""
61 for i in range(100):
62 random_g = random.choice(goneas)
63 random_g2 = random.choice(goneas)
64 comp1 = []
65 comp2 = []
66 for i in range(4):
67 comp1.append(random_g[i + 1])
68 comp2.append(random_g2[i + 1])
69 x = 0
70 end = 0
71 merg = []
72 while x != 5 and end != 1:
73 num = randint(0, 1)
74 if x == 0:
0 0
Ln 111, Col 13 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (windows store)


File Edit Selection View Go Run Terminal Help
A.py - Visual Studio Code
A.py
Extension: Python for VSCode
C: > Users > tasso > Desktop > ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ > A.py > ...
merg.append(1)
x = 1
elif num == 0:
for i in range(x):
if merg[i] == random_g[x]:
end = 1
if end == 0:
merg.append(random_g[x])
x += 1
elif num == 1:
for i in range(x):
if merg[i] == random_g2[x]:
end = 1
if end == 0:
nem = int(random_g2[x])
merg.append(random_g[x])
x += 1
if x == 5:
merg.append(1)
kids.append(merg)
arth = 0
for i in range(5):
convert1 = str(merg[i])
convert2 = str(merg[i + 1])
trip = convert1 + '-' + convert2
trip2 = convert2 + '-' + convert1
for j in range(10):
if trip.__eq__(lista[j][0]) or trip2.__eq__(lista[j][0]):
arth += lista[j][1]
arth = 1 / arth
skids.append(arth)
print('kids:')
print(kids)
print("percents of kids:")
print(skids)
Ln 111, Col 13 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (windows store)

Παράδειγμα εκτέλεσης του κώδικα:





4 Βιβλιογραφικές Πηγές

1. Σημειώσεις GUNET2(Τεχνητή νοημοσύνη και έμπειρα συστήματα).
2. Σημειώσεις από διαλέξεις μαθημάτων.