**1η εργασία**

**ΓΡΑΜΜΙΚΟΣ & ΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

**4ο ΕΞΑΜΗΝΟ**

**Φοροτζίδης Χρήστος Α.Μ.1453**

Η φιλοσοφία συγγραφής του προγράμματος είναι διαίρει και βασίλευε. Η γλώσσα προγραμματισμού είναι σε Python και το περιβάλλον ανάπτυξης το Visual Studio Code. Για αυτό χωρίζει όλες τις διαδικασίες σε πολλές συναρτήσεις ,οι οποίες είναι οι παρακάτω:

Καλούνται στην main:

* opener(), βρίσκεται στο openfiles.py
* size\_function(filename,all\_error) , βρίσκεται στο size\_checker\_and\_Eqin.py
* B\_array\_creator(filename,size\_of\_rows,all\_error), βρίσκεται στο B\_array.py
* check\_if\_\_miss\_symbol(filename), βρίσκεται στο check\_numbers\_after\_restrictions.py
* find\_arrays\_A\_C(filename,size\_of\_rows,size\_of\_column,all\_error), βρίσκεται στο C\_A\_finder.py

Καλείται στην C\_A\_finder.py

* seperate(words,number\_of\_variables,counter\_of\_line,error\_counter) , βρίσκεται στο seperation.py

Καλείται στην seperation.py

* give\_the\_right\_form(aword), βρίσκεται στο change\_shape.py

Ξεκινάει το πρόγραμμα από την main και απευθείας καλεί την συνάρτηση opener().

Αυτή η συνάρτηση ζητάει να από τον χρήστη να εισάγει το όνομα του αρχείου που θέλει να γίνει η επεξεργασία. Το αρχείο οφείλει να βρίσκεται στο ίδιο path με το πρόγραμμα. Αν το αρχείο δεν βρίσκεται ή ο χρήστης δεν εισάγει σωστά το όνομα του αρχείου ,τότε τον ενημερώνει για την κατάσταση στην οποία βρίσκεται.

Γυρνάει το αρχείο και το στέλνουμε στην συνάρτηση size\_function(filename,all\_error) .Μας γυρνάει τα εξής all\_error,size\_of\_rows, size\_of\_column,Eqin. Γνωρίζοντας τα μεγέθη του προβλήματος μπορούμε να προχωρήσουμε .Σε περίπτωση που οι περιορισμοί είναι διαφορετικό νούμερο από τις μεταβλήτες του προβλήματος , μας ενημερώνει. Επίσης μας ενημερώνει αν λείπει κάποιο σύμβολο >,<,= και μας εμφανίζει στην σειρά που γίνεται αυτό(εκτός της αντικειμενικής συνάρτησης που δεν έχει )

'Επειτα καλεί την B\_array\_creator(filename,size\_of\_rows,all\_error) και μας γυρνάει τα εξής all\_error,B. Σε αυτήν συνάρτηση γυρνάει τον πίνακα Β και μας ενημερώνει αν δεν βρίσκεται αριθμός μετά από κάποιον περιορισμό.

Ύστερα καλεί την find\_arrays\_A\_C(filename,size\_of\_rows,size\_of\_column,all\_error) και μας γυρνάει τα εξής all\_error,X,A. 'Οπου το Χ είναι ένας πίνακας size\_of\_column\*1 και μετά στην main βάζουμε τα στοιχεία του Χ στον πίνακα C που είναι μεγέθος 1\*size\_of\_column. Μέσα στην find\_arrays\_A\_C καλούμε την seperate(words,number\_of\_variables,counter\_of\_line,error\_counter) και μας γύρναει έναν πίνακα size\_of\_column\*1 ,ουσιαστικά στέλνουμε κάθε γραμμή και μας το γυρνάει έτοιμη σε ένα πίνακα. Μέσα στην seperate() ,αν η γραμμή που εχεί σταλθεί είναι αλφαριθμητικό που είναι 1 στοιχείο στον πίνακα ,τότε καλεί την give\_the\_right\_form(aword) και γυρνάει το αλφαριθμητικό σε πίνακα πολλών στοιχείων ,ώστε η seperate() να μπορεί να ολοκληρώσει τις εργασίες της. Η seperate() ελέγχει αν λείπει ο αριθμός που προσδιορίζει τις μεταβλητές ,μας βγάζει το κατάλληλο μήνυμα που βρίσκεται η έλλειψη και ενημερώνει με την σειρά τις προηγούμενες συναρτήσεις. Επίσης μας ελέγχει αν λείπει κάποιο + ή - μεταξύ των μεταβλητών,μας βγάζει το κατάλληλο μήνυμα που βρίσκεται η έλλειψη και ενημερώνει με την σειρά τις προηγούμενες συναρτήσεις.

Έπειτα καλεί την check\_if\_\_miss\_symbol (filename) οπού ελεγχεί αν λείπει κάποιος αριθμός μετά τους περιορισμός . Αυτή γράφτηκε ως ανάγκη για επικάλυψει κάποιων σφαλμάτων που γινόντουσαν στην B\_array\_creator.

Έπειτα στην main γίνονται οι έλεγχοι για το MinMax και το Subject/s.t./st

Όλοι οι έλεγχοι ενημερώνουν το all\_error και αν έχει τουλάχιστον 1 error τότε το πρόγραμμα δεν ολοκληρώνει την δημιουργια του αρχείου εξόδου. Και μας ενημερώνει πόσα error βρήκε.

Παρατηρήσεις

Το πρόγραμμα επανλαμβάνει 4 φορές το να διαβάσει το αρχείο ,αυτό θα μπορούσε να αποφεύγετε αλλά ήταν αργά οταν το συνειδητοποίησα ,όμως η σύγχρονη υπολογιστική δύναμη των υπολογιστών το υπερκαλύπτουν.Δηλαδή η διαφόρα θα ήταν σε microseconds , ούτε καν σε seconds .

Έγιναν κάποια λάθη εκτίμησης για το μέγεθος του προγράμματος , πίστευα θα είναι 50% λιγότερο από αυτό που έγραψα , και δεν χρησιμοποίησα το git ώστε να δημιουργώ branch. Αυτό θα με βοηθούσε αρκετά στο σημείο που ξεκίνησα να διορθώνω διάφορα error . Επισής όταν ξεκινήσα να βάζω διάφορα catch error για τις απαιτήσεις τις εργασίας εκεί θα με βοηθούσε επίσης.

Γενικότερα προσπάθησα να δίνω όσο πιο σαφής ονόματα στις μεταβλητές . Έγραψα αρκετά σχόλια σε διάφορα σημεία ,ώστε να γίνεται αντιληπτό τι κάνω.

Θα δείτε πολλά σκουπίδια τύπου #print ,θεώρησα οτι έπρεπε να τα αφήσω ώστε σε μια μελλοντική παρέμβαση να μπορώ να ξέρω πως έκανα ελέγχους όταν το έγραφα.

Για τους πίνακες χρησιμοποίησα την δομή της python list αντί του numpy array.

Ευχαριστώ , αν υπάρχουν παρατηρήσεις θα ήθελα πολύ να τις μάθω

επικοινωνήστε μαζί μου it1453@uom.edu.gr