*Πανεπιστήμιο Ιωαννίνων*

*Πολυτεχνική Σχολή*

*Τμήμα Μηχανικών Η/Υ και Πληροφορικής*

**ΜΥΥ802: Μεταφραστές**

Διδάσκων: Γ. Μανής

Ακαδημαϊκό έτος: 2022-2023

Βουτσινάς Ανδρέας Α.Μ.: 3153

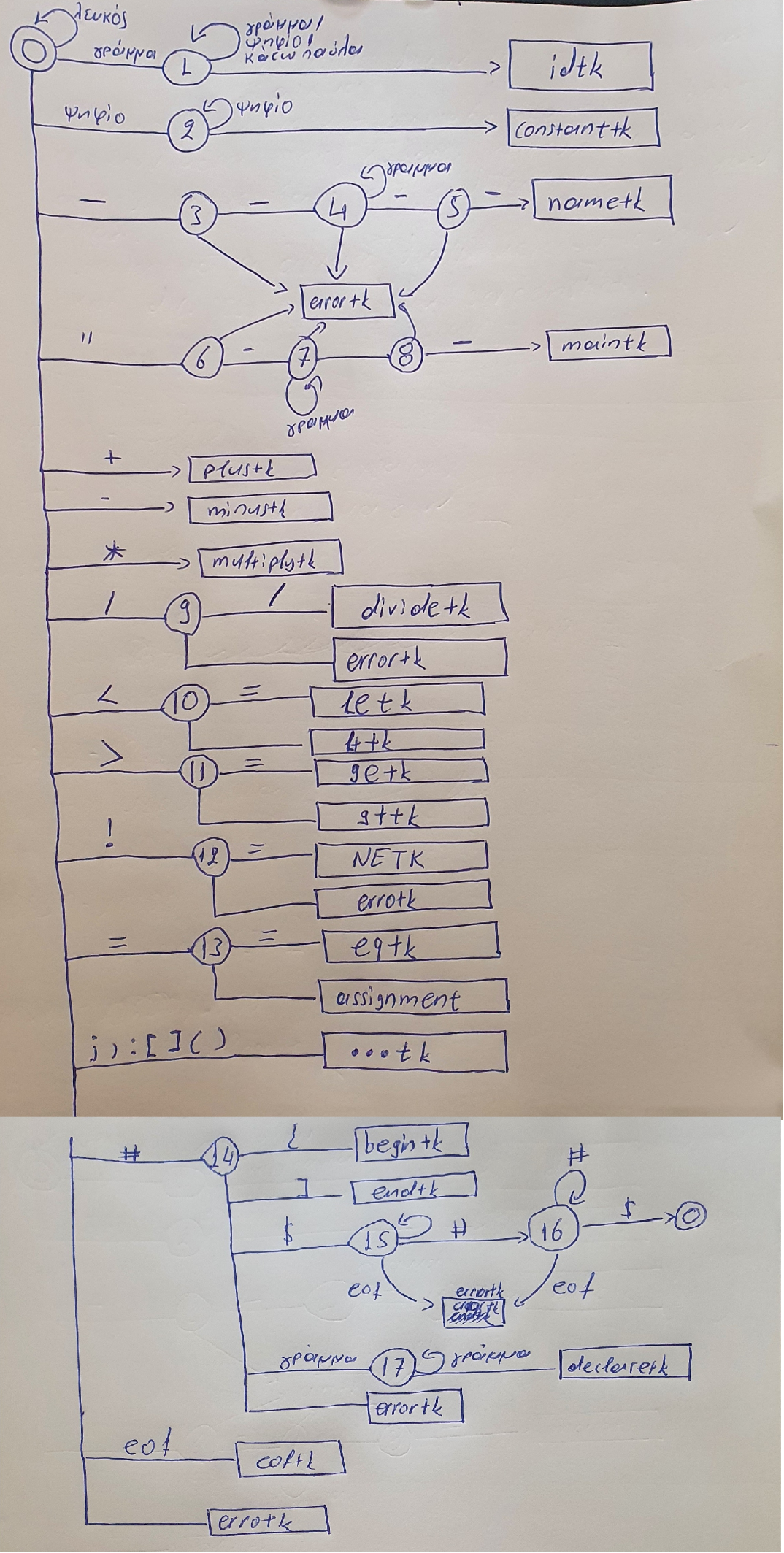
Σταλίκας Δημήτριος Α.Μ.: 3079

**Μεταφραστές Αναφορά**

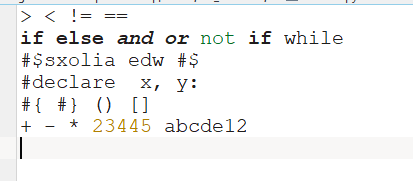
Το παρόν έγγραφο αποτελεί την αναφορά για την λειτουργία του μεταγλωττιστή της γλώσσας CutePy ο οποίος συντάχθηκε για την εργαστηριακή άσκηση του μαθήματος των μεταφραστών. Στην συνέχεια περιγράφεται εν συντομία η λειτουργία κάθε τμήματος του μεταγλωττιστή και μέσα από παραδείγματα εξηγείται η ορθή λειτουργία του.

**Λεκτικός αναλυτής**

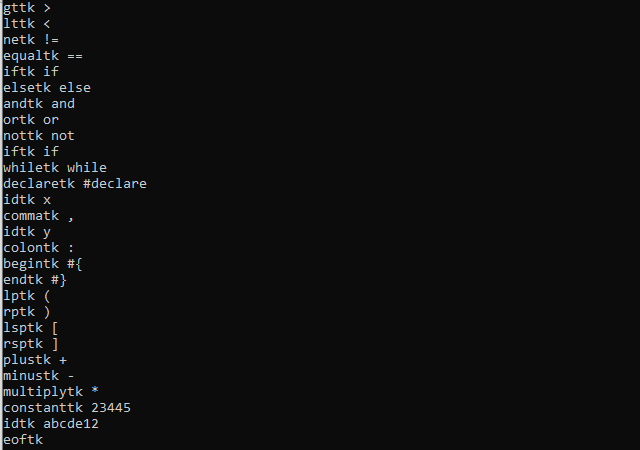
Η λειτουργία του λεκτικού αναλυτή αφορά στην εύρεση των λεκτικών μονάδων ενός προγράμματος. Κάθε λεκτική μονάδα αποτελείται από την λέξη και τον κωδικό στον οποίο αντιστοιχεί. Ο λεκτικός αναλυτής βασίζεται στο παρακάτω αυτόματο που δημιουργήθηκε με βάση τα δεδομένα της εκφώνησης.



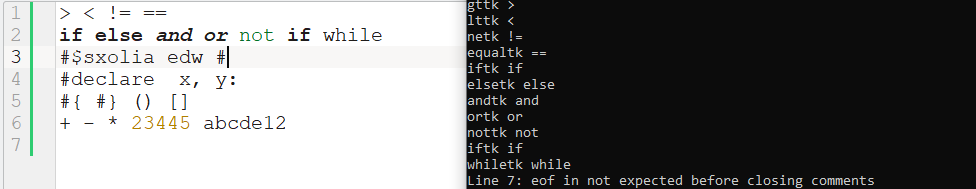
Το αυτόματο μετατράπηκε σε έναν πίνακα μεταβάσεων όπου με βάση την κατάσταση στην οποία βρίσκεται ο αναλυτής και τον χαρακτήρα που έχει διαβάσει μετακινείται σε επόμενη κατάσταση μέχρι να βρεθεί σε τελική. Ο κώδικας που ακολουθεί περιέχει διάφορες λεκτικές μονάδες ώστε να ελέγξουμε ότι ο λεκτικός αναλυτής επιστρέφει σωστά τις μονάδες αυτές.



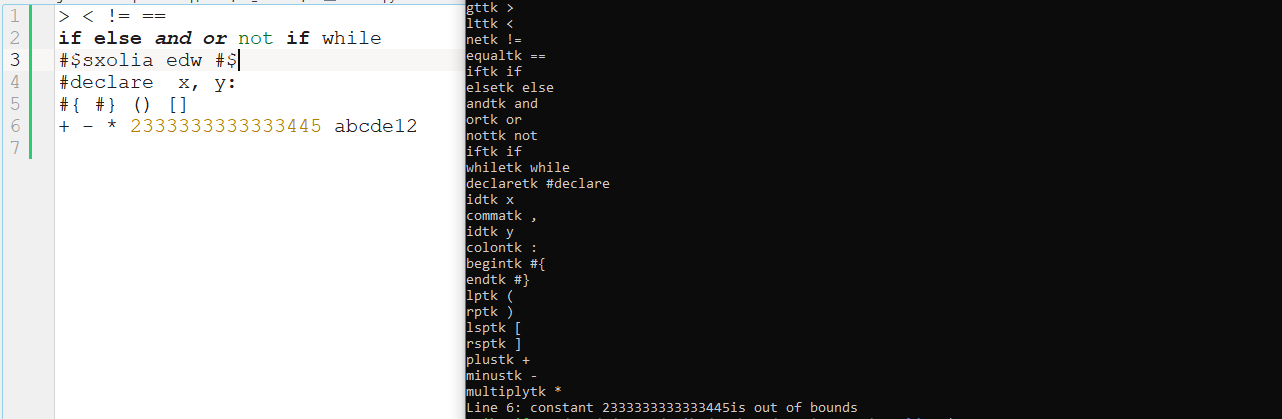
Η έξοδος δίνεται στην συνέχεια



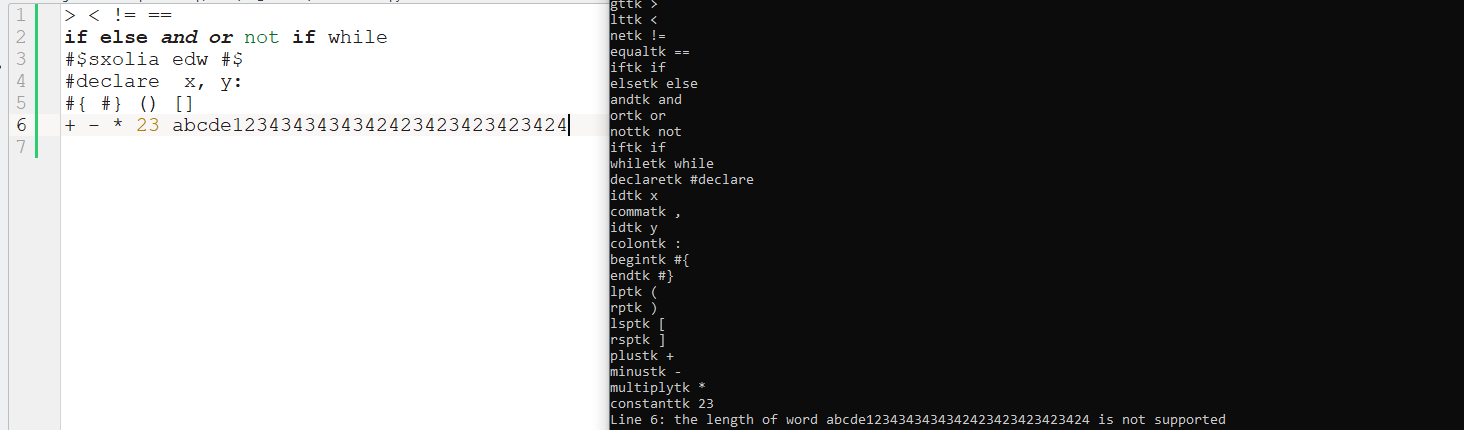
Το πρόγραμμα στην φάση αυτή μπορεί να αναγνωρίσει σφάλματα σχετικά με τα σχόλια, εάν δηλαδή δεν κλείνουν σωστά



σφάλματα σχετικά με το μέγεθος των σταθερών



ή με το μήκος των αναγνωριστικών

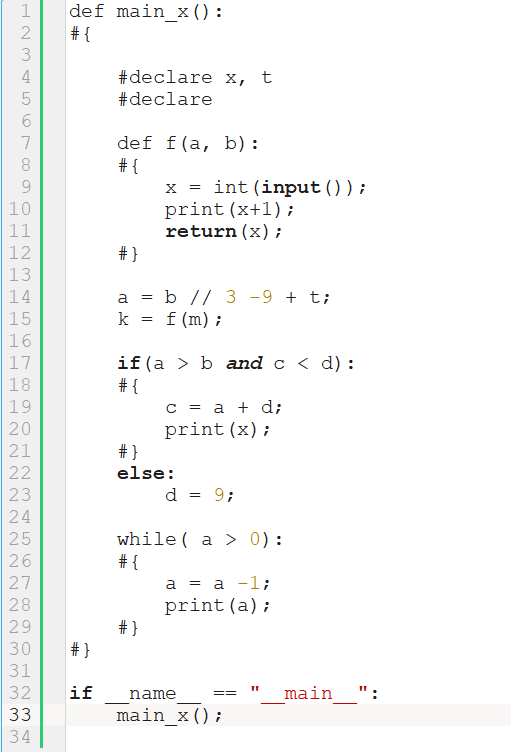


Επίσης αναγνωρίζονται λάθη σχετικά με τις λεκτικές μονάδες #declare, “\_\_main\_\_” και \_\_name\_\_ τα οποία όμως παραλείπονται από την αναφορά ώστε αυτή να παραμείνει σε λογικό μέγεθος

**Συντακτικός αναλυτής**

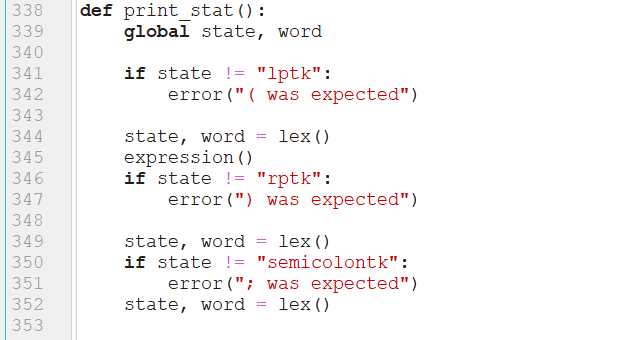
Ο συντακτικός αναλυτής ελέγχει εάν το πρόγραμμα ακολουθεί τους συντακτικούς κανόνες της γραμματικής της γλώσσας. Η γραμματική περιγράφεται από ένα πλήθος κανόνων. Για κάθε κανόνα έχει δημιουργηθεί μια συνάρτηση η οποία ελέγχει εάν ο κανόνας ακολουθείται από το πρόγραμμα. Αυτό γίνεται με την βοήθεια του λεκτικού αναλυτή. Αρχικά καλείται ο λεκτικός αναλυτής και ελέγχεται εάν η λεκτική μονάδα που έχει επιστραφεί είναι η αναμενόμενη. Εάν όχι τυπώνεται κατάλληλο μήνυμα και γίνεται τερματισμός. Εάν όχι τότε ο συντακτικός αναλυτής προχωράει στην επόμενη λεκτική μονάδα.

Το πρόγραμμα που ακολουθεί περνάει την συντακτική ανάλυση χωρίς κάποιο σφάλμα.



Το πρόγραμμα αυτό είναι πλήρες γιατί περιέχει μια κυρίως συνάρτηση που περιέχει μια εμφωλευμένη τοπική. Και οι δύο δηλώνουν τοπικές μεταβλητές. Το πρόγραμμα έχει από μια κλήση των statements της γλώσσας, για είσοδο, για έξοδο, για επιστροφή τιμής, μία if και μια while. Τέλος γίνονται όλες οι πράξεις και υπάρχουν συνθήκες που ενώνονται με το and. Αξίζει να σημειωθεί πως έχουν γίνει και έλεγχοι με προγράμματα που περιέχουν και άλλους λογικούς τελεστές.

Ο συντακτικός αναλυτής, όπως προαναφέρθηκε, σε περίπτωση μη αναμενόμενης λεκτικής μονάδας εμφανίζει κατάλληλο μήνυμα λάθους. Έχει γίνει έλεγχος για την λειτουργία όλων των συναρτήσεων, στην συνέχεια όμως θα δείξουμε για μία επιλεγμένη για να φανεί η λειτουργία των ελέγχων. Μία συνάρτηση που έχει αρκετούς ελέγχους αλλά είναι σχετικά μικρή, για οικονομία χώρου, είναι η print\_stat



Η συνάρτηση περιμένει μετά από το *print* μια παρένθεση



μετα την παρένθεση μια έκφραση



στην συνέχεια κλείσιμο παρένθεσης



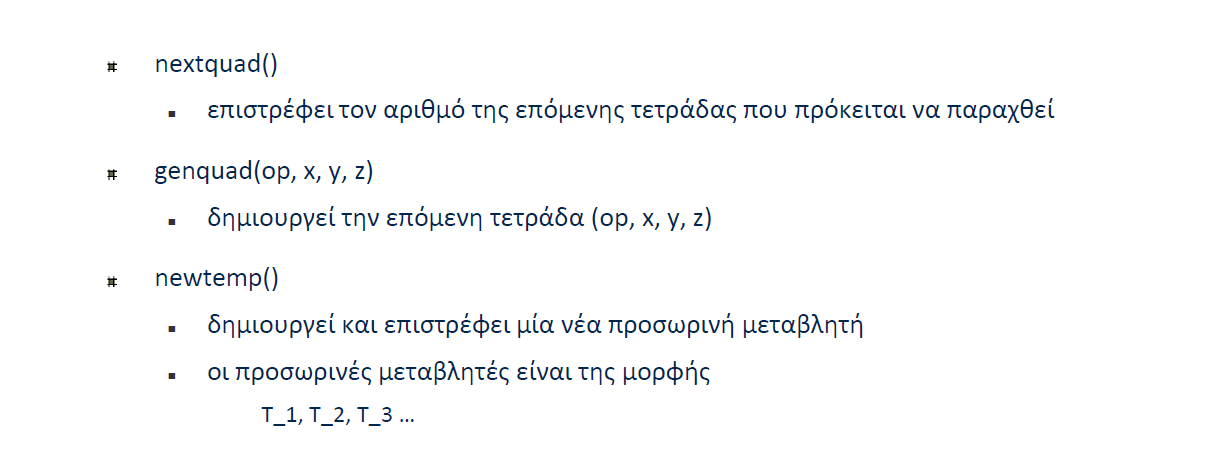
και τέλος το ερωτηματικό

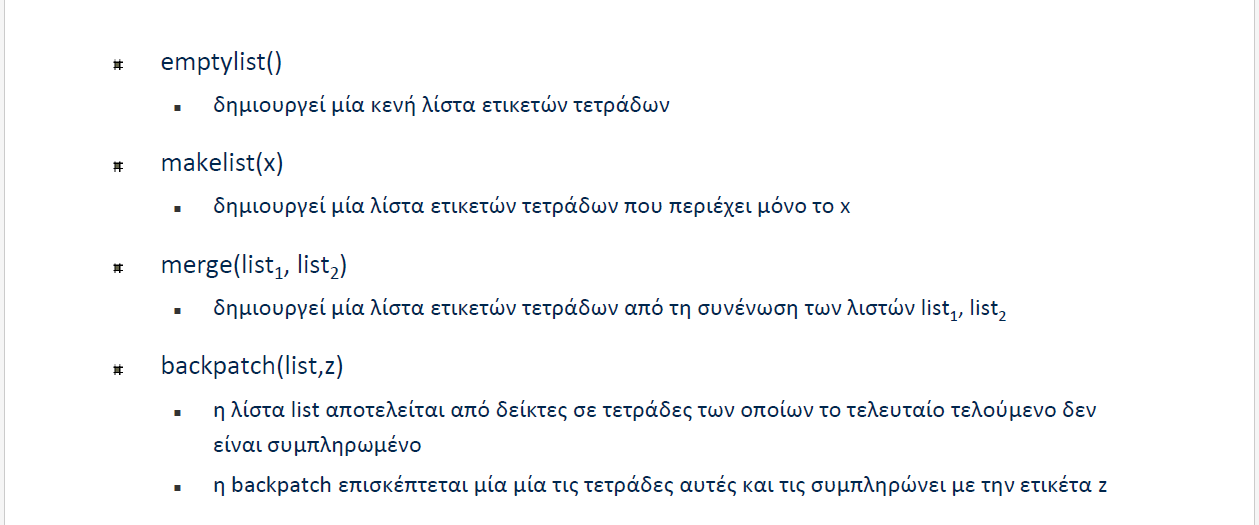


Τα παραπάνω γίνονται σε όλες τις συναρτήσεις και επομένως ένα πρόγραμμα με συντακτικά λάθη εμφανίζει κατάλληλο μήνυμα ώστε να διορθωθεί το λάθος

**Ενδιάμεσος κώδικας**

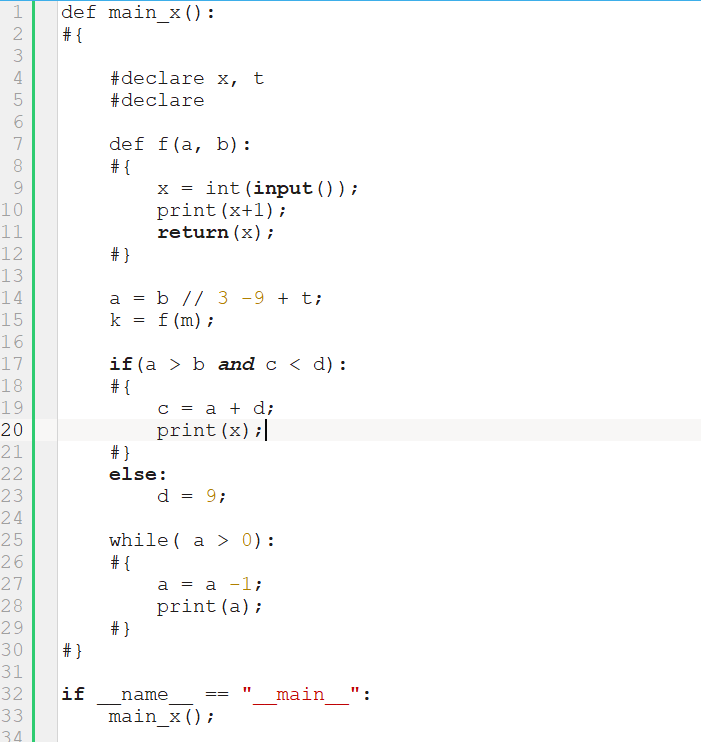
Ο ενδιάμεσος κώδικας παράγεται παράλληλα με την συντακτική ανάλυση του κώδικα. Στόχος του είναι να μετατρέψει τον κώδικα της γλώσσας CutePy σε τετράδες ώστε να είναι ευκολότερη η μετατροπή του σε γλώσσα μηχανής. Ο ενδιάμεσος κώδικας βασίζεται στις παρακάτω 7 συναρτήσεις:



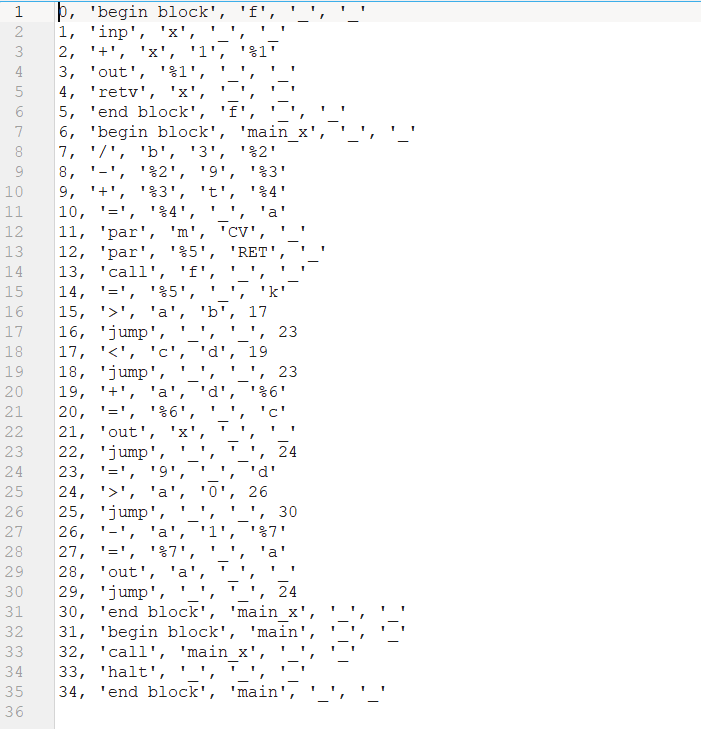


Οι παραπάνω συναρτήσεις καλούνται στα κατάλληλα σημεία του συντακτικού αναλυτή για να παράγουν τις σωστές τετράδες. Ο ενδιάμεσος κώδικας παράγεται πρώτα για την συνάρτηση που βρίσκεται στο μεγαλύτερο βάθος φωλιάσματος και μετά για τις εξωτερικές της. Ο ενδιάμεσος κώδικας αποθηκεύεται σε αρχείο με την κατάληξη .int.

Ο κώδικας που ακολουθεί έχει αρκετά στοιχεία για να ελεγχθεί η ορθότητα του ενδιάμεσου κώδικα



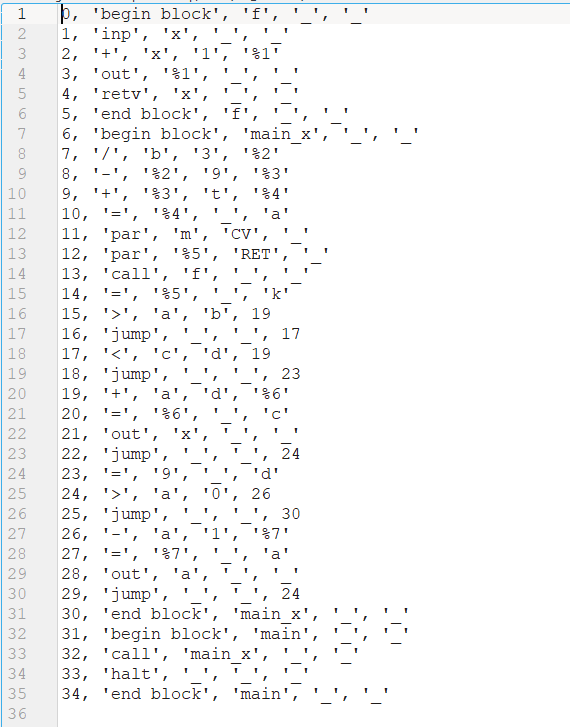
Το αποτέλεσμα για τον ενδιάμεσο κώδικα είναι:



Απο το παραπάνω παράδειγμα συμπεραίνουμε ότι ο ενδιάμεσος κώδικας λειτουργεί σωστά γιατί:

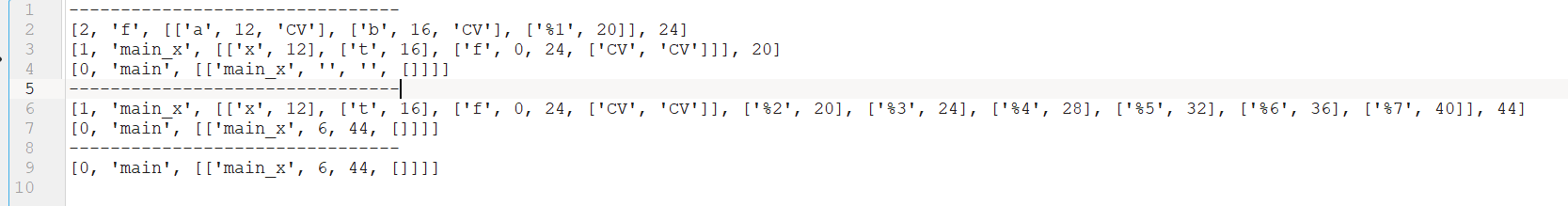
1. ξεκινάει με τη πιο εσωτερική συνάρτηση
2. κάθε συνάρτηση έχει δικά της begin και end block τα οποία δεν εμπλέκονται με άλλη συνάρτηση
3. Δεν υπάρχουν τετράδες σε συνθήκες που δεν είναι συμπληρωμένες
4. Το if παράγει ένα Jump για να μην εκτελέσει το else.
5. το while επιστρέφει στην αρχή του για να ελέγξει ξανά την συνθήκη του
6. το πρόγραμμα έχει ως προτελευταία εντολή το halt
7. Πριν την κλήση τοπικής συνάρτησης υπάρχουν οι παράμετροι και μια παράμετρος τύπου RET
8. Η κλήση της κύριας συνάρτησης δεν εμπλέκει παραμέτρους
9. Το λογικό AND δουλεύει σωστά

Για το τελευταίο, παρατηρούμε πως εάν αλλάξουμε το and με or υπάρχει και η ανάλογη αλλαγή στον ενδιάμεσο κώδικα



**Πίνακας συμβόλων**

Ο πίνακας συμβόλων παραγεται παράλληλα με την συντακτική ανάλυση και ως επι τον πλείστων στα σημεία όπου δεν παράγεται ενδιάμεσος κώδικας. Ο πίνακας συμβόλων διατηρεί πληροφορία για τις μεταβλητές και τις συναρτήσεις που δηλώνονται στο πρόγραμμα. Οι πληροφορίες αυτές αφορούν το βάθος φωλιάσματος, το μήκος(framelength) και την πρώτη τετράδα (start quad) για μια συνάρτηση ενώ για μεταβλητές και παραμέτρους το offset. Για τις παραμέτρους διατηρείται και ο τύπος της μεταβλητής. Στον πίνακα συμβόλων εισάγεται μια γραμμή όταν ξεκινάει η δήλωση μια συνάρτησης και διαγράφεται από αυτόν όταν τερματίζει η δήλωση της συνάρτησης. Στην συνέχεια ακολουθεί ο πίνακας συμβόλων που παράγεται για το πρόγραμμα που ελέγχθηκε ο ενδιάμεσος. Χρησιμοποιείται ακριβως το ίδιο παράδειγμα μιας και ήταν ζητούμενο στην ίδια παράδοση. Και ο πίνακας συμβόλων αποθηκεύεται σε αρχείο



Ο πίνακας τυπώνεται σε 3 μορφές του, πριν από κάθε μια διαγραφή. Παρατηρούμε ότι και αυτός είναι σωστός καθώς για κάθε entity κρατάει τα σωστά στοιχεία. Επίσης παρατηρούμε, και αναμένουμε, τα στοιχεία για κάθε συνάρτηση για το entity που τους αντιστοιχεί στο scope του πατέρα ότι ενημερώνονται λίγο πριν την διαγραφή του scope τους.

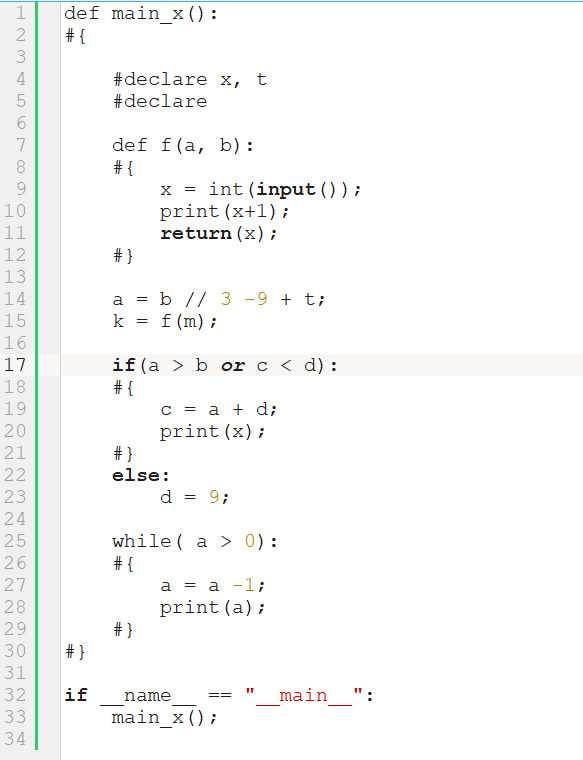
**Σημασιολογική ανάλυση**

Η σημασιολογική ανάλυση αφορά στην ορθή χρήση των μεταβλητών και συναρτήσεων που έχουν δηλωθεί στο πρόγραμμα καθώς και στην χρήση δηλωμένων μεταβλητών και συναρτήσεων. Για την σημασιολογική ανάλυση χρησιμοποιείται κατα βάση ο πίνακας συμβόλων. Οι περιπτώσεις που ελέγχονται είναι:

1. Να μην έχει δηλωθεί μια entity δύο φορές στο ίδιο scope
2. Να έχει δηλωθεί οτιδήποτε χρησιμοποιείται
3. Να χρησιμοποιείται μια συνάρτηση σαν συνάρτηση και όχι σαν μεταβλητή
4. Να χρησιμοποιείται μια μεταβλητή/παράμετρος σαν μεταβλητή/παράμετρος και όχι σαν συνάρτηση
5. Μια κύρια συνάρτηση να ξεκινάει το όνομα της με main\_
6. Μια κύρια συνάρτηση να μην μπορεί να κληθεί από άλλη κύρια
7. Μια κύρια συνάρτηση να μην επιστρέφει τιμή
8. Μια τοπική συνάρτηση να επιστρέφει τιμή

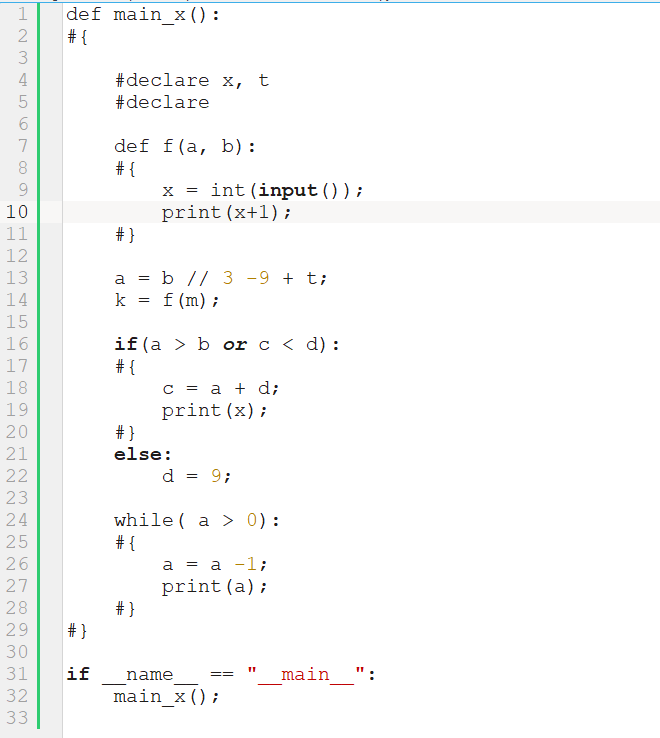
Στην συνέχεια δείχνουμε 2 παραδείγματα από τα παραπάνω

Μια μεταβλητή που δεν έχει δηλωθεί





και μια συνάρτηση χωρίς return





**Τελικός κώδικας**

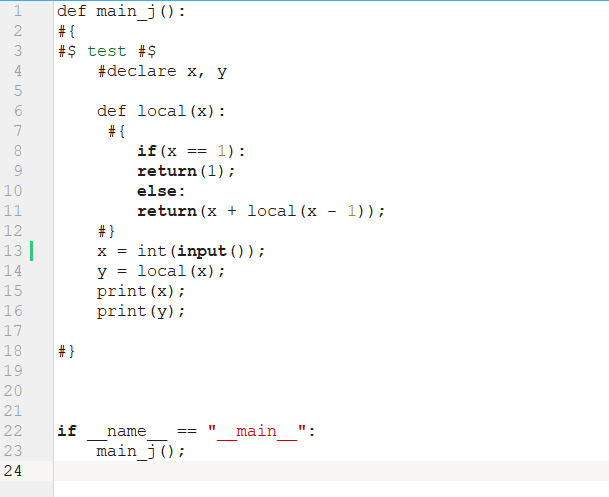
Στην φάση του τελικού κώδικα μετατρέπονται οι τετράδες του ενδιάμεσου σε γλώσσα assembly. Η μετατροπή γίνεται αντλώντας πληροφορία από τον πίνακα συμβόλων και για τον λόγο αυτό η μετατροπή πρέπει να γίνει στο τέλος κάθε συνάρτησης πριν διαγραφεί το scope της από τον πίνακα συμβόλων

Τα παραδείγματα που ακολουθούν ελέγχουν την σωστή λειτουργία του τελικού κώδικα. Σε αυτό ελέγχονται διάφορα όπως:

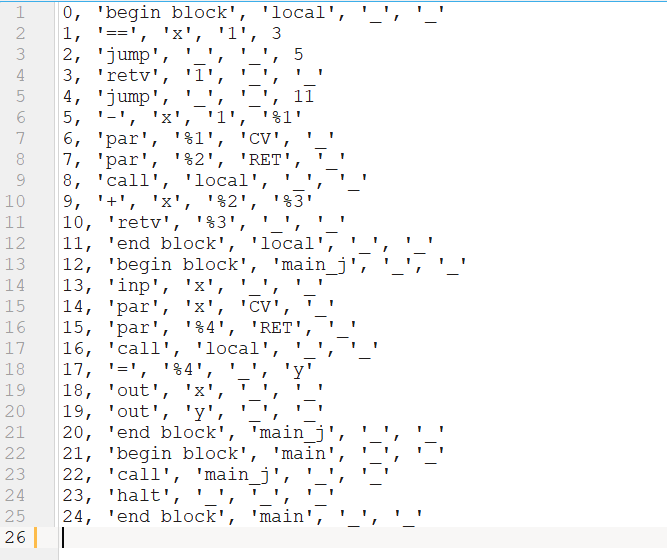
1. Η είσοδος τιμής
2. Η εκτύπωση
3. Η κλήση συνάρτησης
4. Η σωστή λειτουργία του περάσματος παραμέτρων
5. Η λειτουργία της επιστροφής τιμής από συνάρτηση
6. Η λειτουργία του if
7. H λειτουργία του while

Η λειτουργία ελέγχθηκε αρχικά ελέγχοντας μια προς μια τις παραγόμενες εντολές και στην συνέχεια με το rars

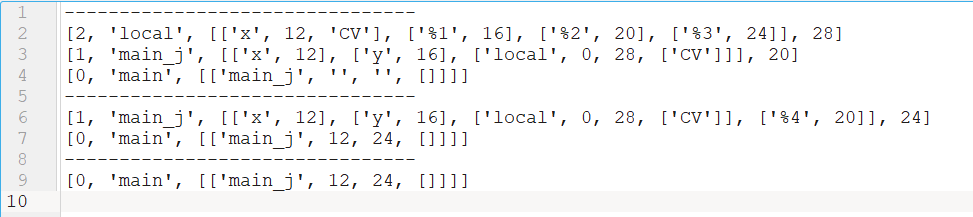
Ο κώδικας σε CutePy



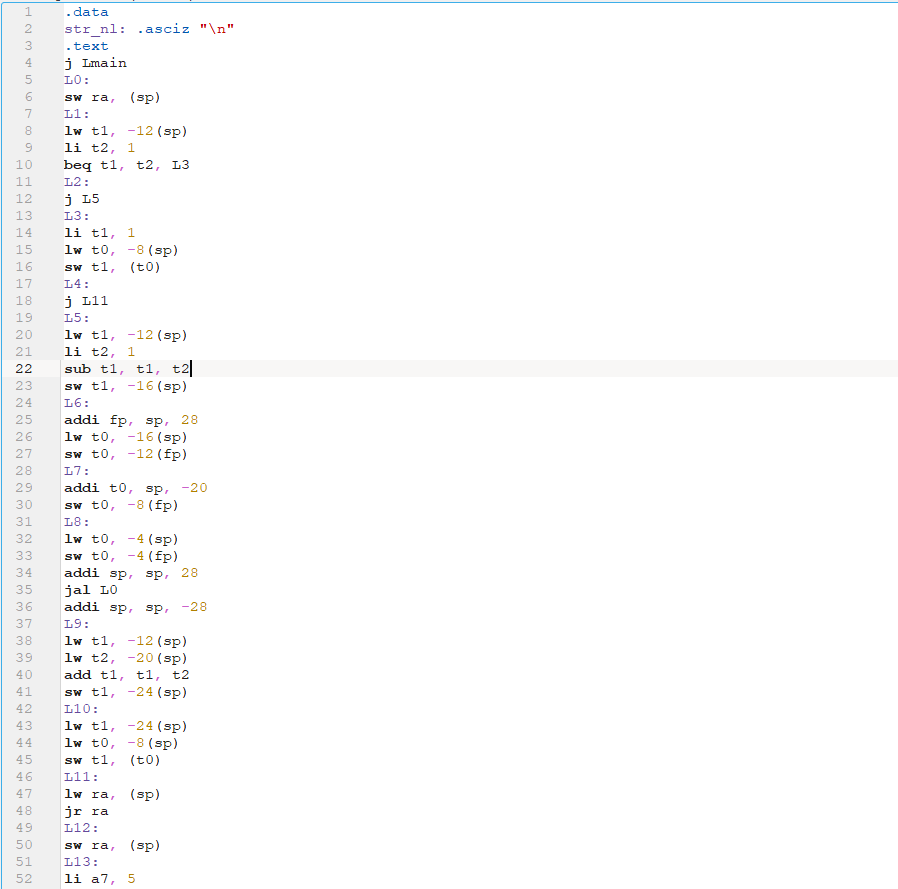
Ο ενδιάμεσος κώδικας

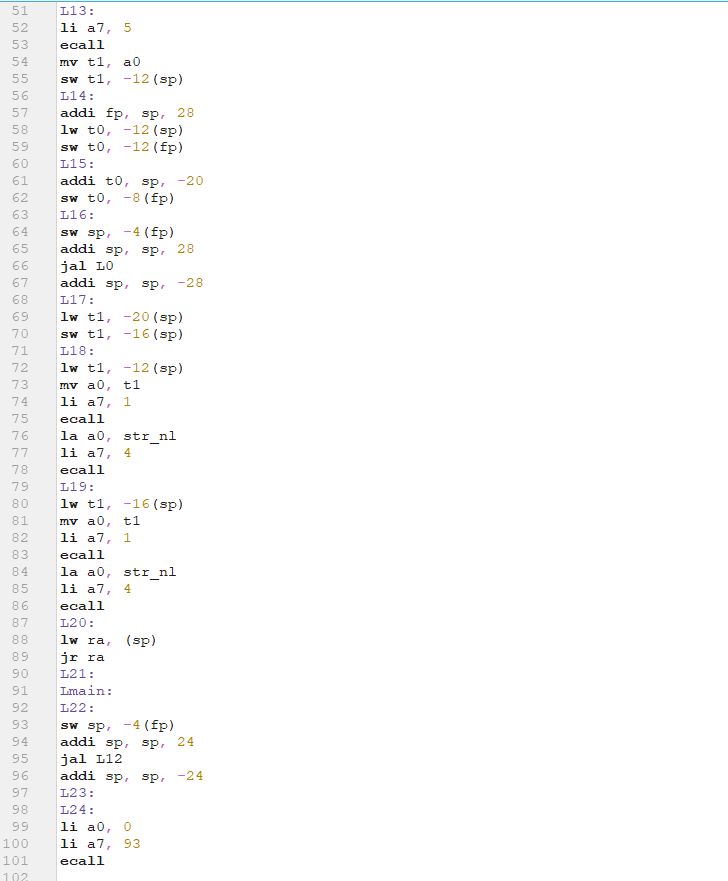


Ο πίνακας συμβόλων

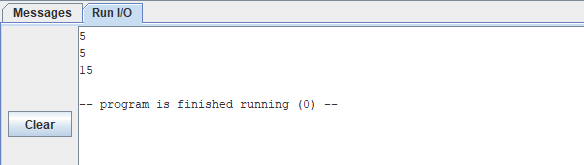


Ο τελικός κώδικας





και τέλος το αποτέλεσμα από τον rars με είσοδο 5

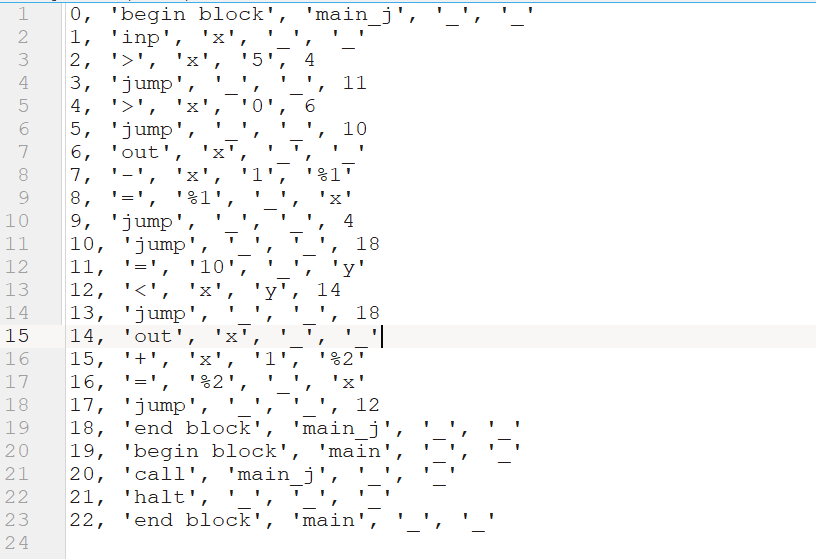


Το πρόγραμμα ζητάει μια τιμή από τον χρήστη και στην συνέχεια υπολογίζει αναδρομικά το άθροισμα των τιμών από το χ έως το 1. Εκτυπώνουμε την αρχική τιμή μετά την κλήση της αναδρομικής συνάρτησης και το αποτέλεσμα και περιμένουμε 5 και 15 αντίστοιχα εάν δουλεύει σωστά ο τελικός κώδικας. Οι τιμές που πήραμε είναι αυτές που περιμέναμε επομένως όλα είναι εντάξει.

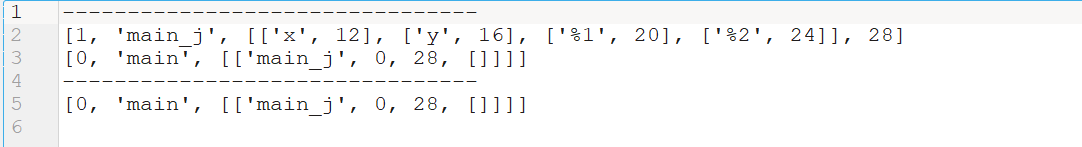
Ένα ακόμα παράδειγμα είναι



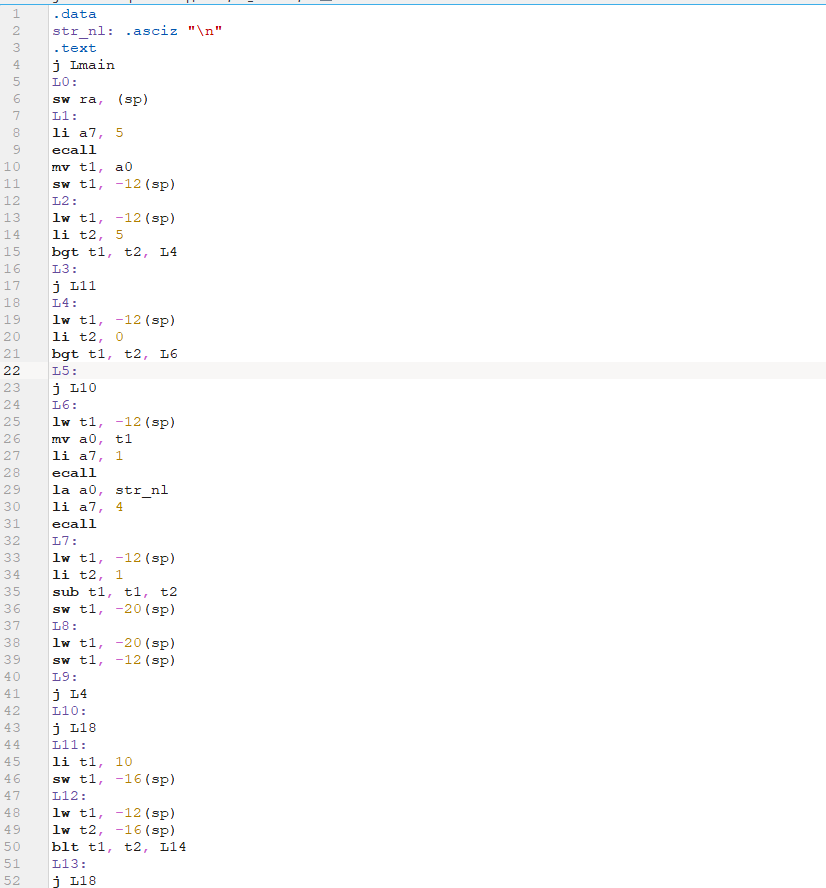
Ο ενδιάμεσος κώδικας είναι

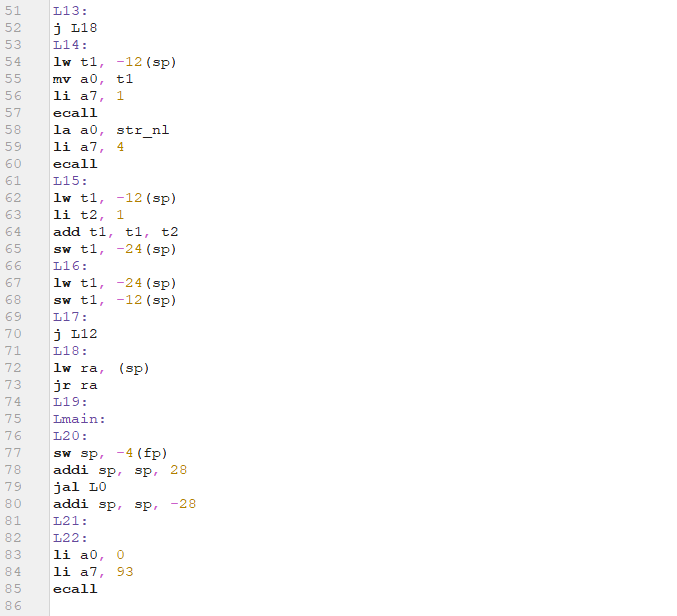


Ο πίνακας συμβόλων

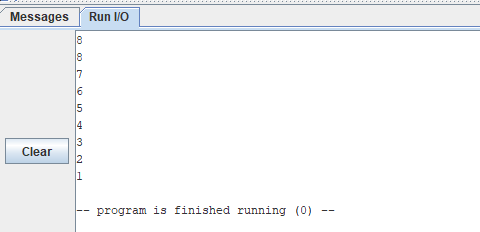


και ο τελικός κώδικας

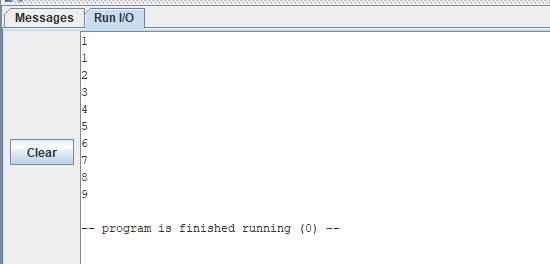




Στην πρώτη κλήση δόθηκε το 8 οπότε τύπωσε τους αριθμούς από το 8 μέχρι το 0 ανάποδα.



Στην δεύτερη κλήση δόθηκε το 1 όποτε τύπωσε τους αριθμούς 1 έως 9.



Τα παραπάνω είναι τα αναμενόμενα αποτελέσματα