ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος

Μεταγλωττιστές

Άσκηση	***Β)***

Εκφώνηση εργασίας

Β) Υλοποιήστε σε γλώσσα προγραμματισμού της επιλογής σας μια γεννήτριασυμβολοσειρών για τη γραμματική που ορίζεται με τους εξής κανόνες παραγωγής: <έκφραση>::=(<υποέκφραση>)
<υποέκφραση>::=<στοιχείο1><στοιχείο2>
<στοιχείο1>::=v|<έκφραση>
<στοιχείο2>::=-<υποέκφραση>|+<υποέκφραση>|ε
Το πρόγραμμά σας θα πρέπει να τυπώνει τα βήματα της εκάστοτε παραγωγής.

Ανάλυση του προγράμματος:

Στην εργασία αυτή, το ζητούμενο είναι η κατασκευή μιας γεννήτριας σε γλώσσα της επιλογής μας για την παραπάνω γραμματική. Ταυτόχρονα το πρόγραμμα θα πρέπει να τυπώνει τα βήματα της κάθε παραγωγής. Το πρόγραμμα είναι γραμμένο σε Python 3.7 και τώρα θα εξηγήσουμε πως λειτουργεί:

Αρχικά κάθε κανόνας είναι και μία συνάρτηση: πχ
Ο πρώτος κανόνας: <έκφραση>::=(<υποέκφραση>) αντιστοιχεί στην def kanonas1:

```
def kanonas1(result,counter):
    print('<έκφραση>::=(<υποέκφραση>)');
    result = result+'(';
    print(result);
    result,counter = kanonas2(result,counter);
    return result,counter;
```

Ο δεύτερος κανόνας <υποέκφραση>::=<στοιχείο1><στοιχείο2> αντιστοιχεί στην def kanonas2:

```
def kanonas2(result,counter):
    print('<υποέκφραση>::=<στοιχείο1><στοιχείο2>');
    ran = (random.randint(0,1));
    if (ran==0):
        counter +=1;
        result,counter = kanonas3a(result,counter);
    elif (ran==1):
        counter +=1;
        result,counter = kanonas3b(result,counter);
    result,counter = kanonas4(result,counter);
    return result,counter;
```

Κανόνες όπου εμπεριέχουν pipes (|) (ή) χωρίζονται σε υποκανόνες , δηλαδή:
Στον 3ο κανόνα <στοιχείο1>::=v|<έκφραση> τα δυνάτα αποτελέσματα είναι δύο το «v» ή το «<έκφραση>» , οπότε χωρίζεται σε kanonas3a και kanonas3b ανάλογα με το αποτέσμα. Ο κανόνας 3a:

```
def kanonas3a(result,counter):
    print('<στοιχείο1>::=v');
    result = result+'v';
    print(result);
    return result,counter;
```

Και ο 3b:

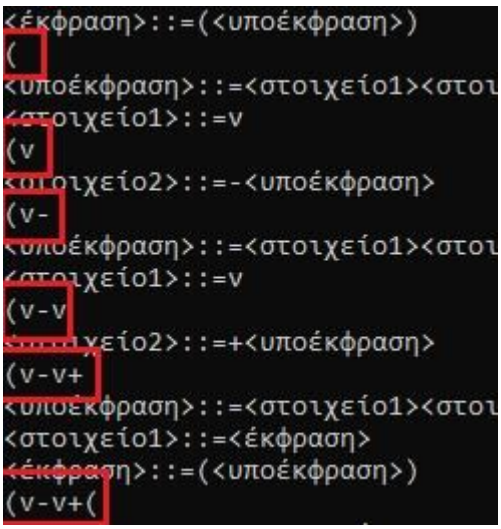
```
def kanonas3b(result,counter):
    print('<στοιχείο1>::=<έκφραση>');
    result,counter = kanonas1(result,counter);
    return result,counter;
```

Τώρα που καταλάβαμε την αντιστοιχία ώρα να αναλύσουμε τις λεπτομέρειες:

Στο πρόγραμμα έχουμε τρεις κύριες μεταβλητές :

1. **result**
2. **counter**
3. **ran (random)**

result : Είναι η μεταβήτη στην οποία αποθηκεύεται η τρέχουσα κατάσταση της στοίβας:



Κάθε κανόνας που εκτελείται , τροποποιεί και την result αντίστοιχα. Συγκεκριμένα:
Όποτε έχουμε εκτέλεση του κανόνα:
kanonas1) έχουμε πρόσθεση χαρακτήρα από δεξιά το σύμβολο (

kanoans3a) έχουμε πρόσθεση χαρακτήρα από δεξιά το σύμβολο v

kanonas4a) έχουμε πρόσθεση χαρακτήρα από δεξιά το σύμβολο –

kanonas4b) έχουμε πρόσθεση χαρακτήρα από δεξιά το σύμβολο +

kanonas4c) έχουμε πρόσθεση χαρακτήρα από δεξιά το σύμβολο) Γιατί προσθέτουμε τον χαρακτήρα) στον κανόνα που βγάζει αποτέσμα κενό; Αν παρατηρήσουμε προσέκτηκα τους κανόνες θα παρατηρήσουμε ότι όποτε έχουμε εκτέλεση του κανόνα 4c που οδηγεί σε κένο , ακριβώς μετά πρέπει να κλείνει και η παρένθεση που άνοιξε με την (<υποέκφραση>) στον κανόνα 1)

counter: Είναι ίσως η πιο περίεργη μεταβητή:
Πρέπει να παρατηρήσουμε ότι όποτε εκτελούμε τον κανόνα 2 , πρέπει μετά να βρούμε και το στοιχείο 1 και το στοιχείο 2 , ξεκινόντας από το στοιχείο 1 αριστερά, η counter λοιπόν μετράει πόσες φορές έχουμε γράψει στοιχείο 1 , ώστε μετά , με την σωστη σειρά να γράψει τα αντίστοιχα στοιχεία 2.

```
def kanonas2(result,counter):
    print('<υποέκφραση>::=<στοιχείο1><στοιχείο2>');
    ran = (random.randint(0,1));
    if (ran==0):
        counter +=1;
        result,counter = kanonas3a(result,counter);
    elif (ran==1):
        counter +=1;
        result,counter = kanonas3b(result,counter);
    result,counter = kanonas4(result,counter);
    return result,counter;
```

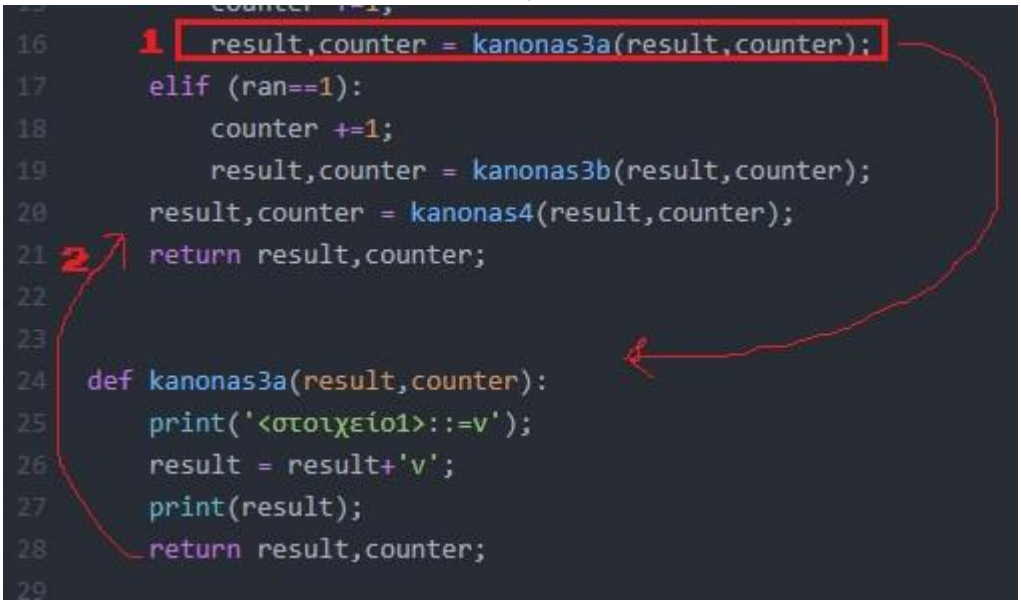
Όποτε λοιπόν εκτελούμε την περίπτωση 3a ή 3b , η counter αυξάνεται κατά ένα.
Μετά όποτε εκτελούμε τον αντίστοιχο κώδικα για το στοιχείο 2 , η counter μειώνεται κατά ένα.

```
def kanonas4(result,counter):
    ran = (random.randint(0,2));
    counter -=1;
    if (ran==0):
        result,counter = kanonas4a(result,counter);
    elif (ran==1):
        result,counter = kanonas4b(result,counter);
    elif (ran==2):
        result,counter = kanonas4c(result,counter);
    return result,counter;
```

Τέλος μόλις κλείσει μία παρένθεση το πρόγραμμα θα μπει στη λούπα:

```
while (counter >0):
    try:
        result,counter = kanonas4(result,counter);
    except:
```

Εφόσον το counter είναι μεγαλύτερο του 0 , σημάινει πως πρέπει να γραφούν counter στοιχεία2 ακόμα. Οπότε τρέχει και τον ανάλογο κώδικα. Όσο για την σωστή σειρά , το στοιχείο 2 γράφεται ανάλογα με το αποτέλεσμα του στοιχείου 1 , ή ακριβώς μετά το v:



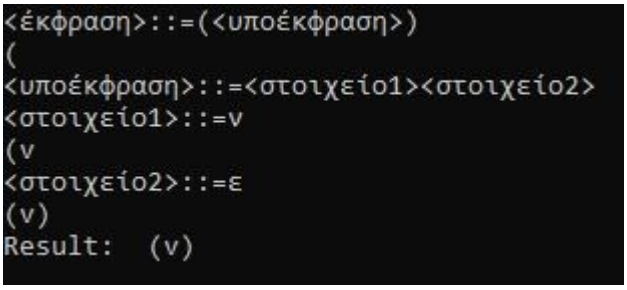
ή ακριβώς μόλις κλείσει παρένθεση και το counter είναι > 0:

```
while (counter >0):
    try:
        result,counter = kanonas4(result,counter);
```

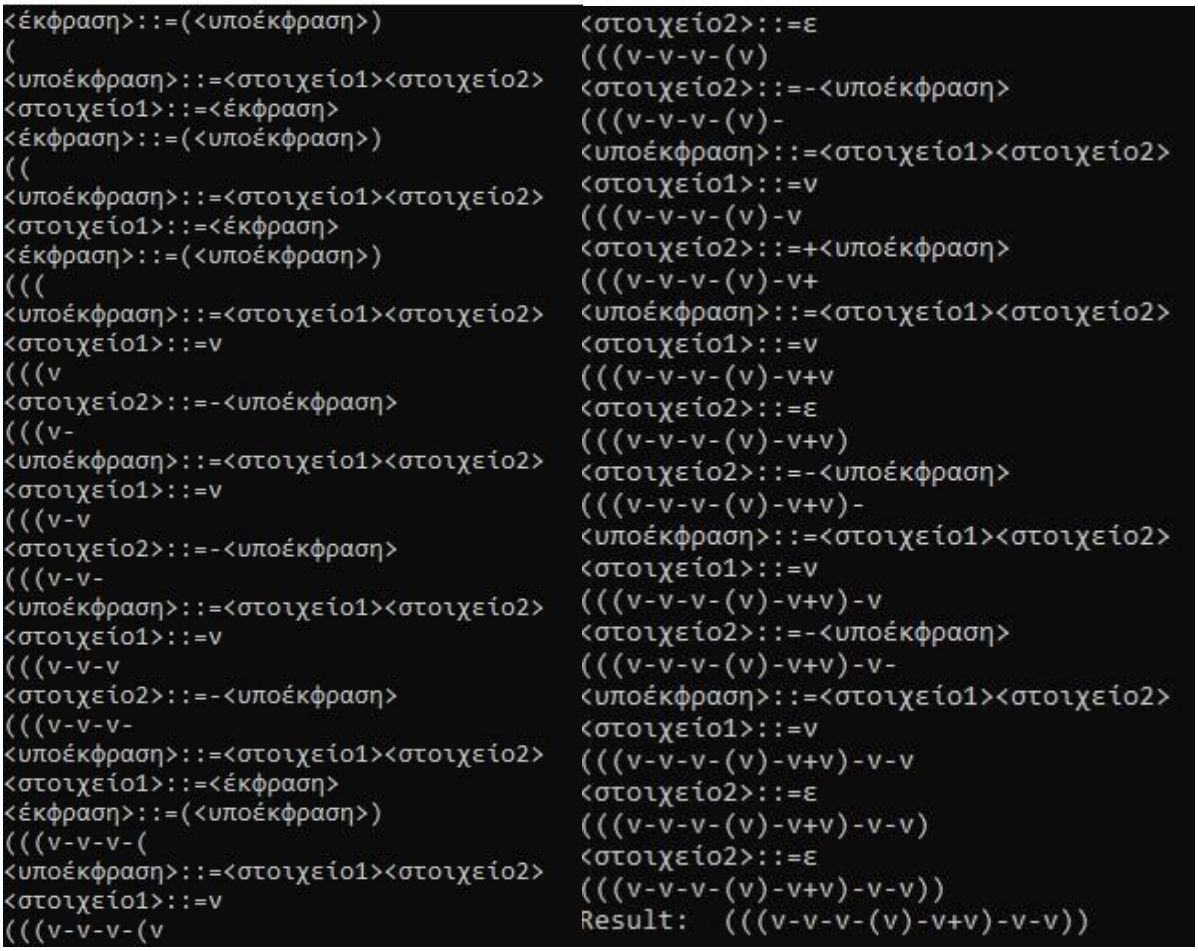
ran: ή αλλιώς random είνα η μεταβλητή που αποθηκεύεται ο αντίστοιχος τυχαίος αριθμός ώστε να γίνει επιλογή του κανόνα όποτε είμαστε ανάμεσα σε πολλούς.

Παραδείγματα:

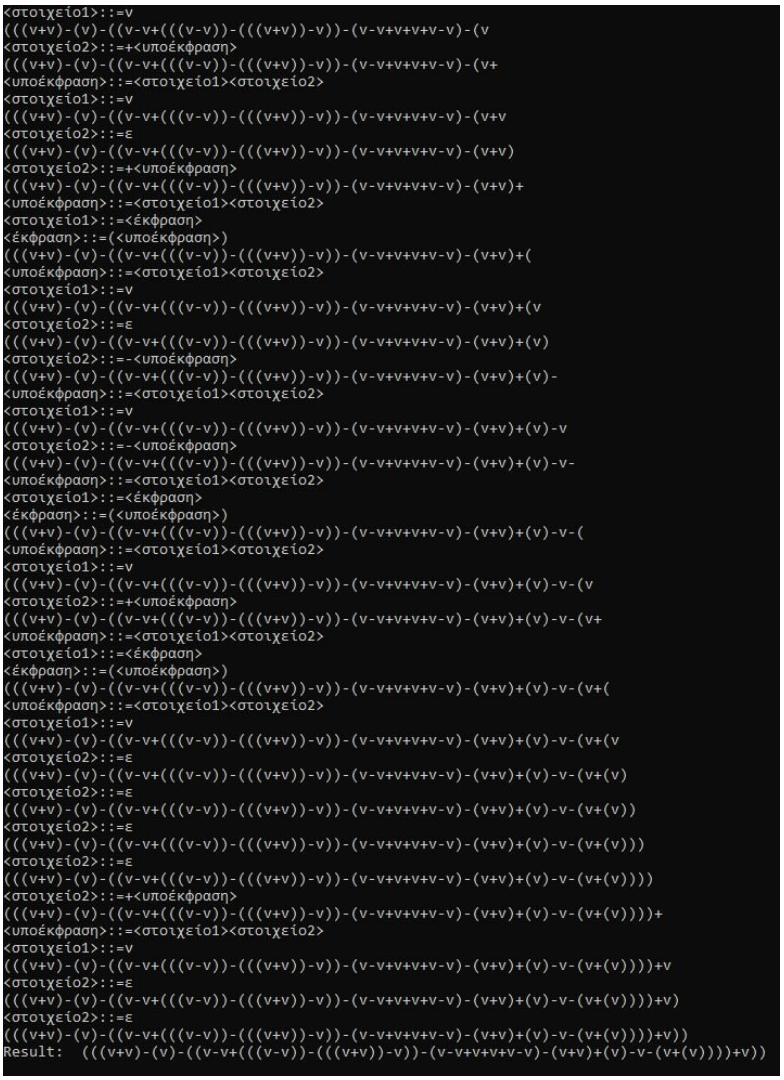
Σε κάθε βήμα τυπώνεται η εκάστοτε παραγωγή και η κατάσταση της στοίβας.



2ο παράδειγμα:



3ο παράδειγμα:



Το πρόγραμμα είναι ικανό να εκτελέσει μεγάλες παραγωγές αλλά μερικές φορές παραείναι μεγάλες: