# Εισαγωγή

…

# Άσκηση Α

## Εισαγωγή

Η πρώτη άσκηση έχεις ως στόχο την υλοποίηση ενός προγράμματος που δέχεται ως είσοδο ένα αρχείο πηγαίου κώδικα και μέσω μιας συγκεκριμένης διαδικασίας αναγνωρίζει τον αριθμό των παρενθέσεων και αποφαίνεται για την ορθότητά τους.

Το πρόγραμμα σε αρχικό στάδιο δέχεται ως όρισμα τη διαδρομή του αρχείο που θα λάβει ως είσοδο ώστε να αρχίσει την επεξεργασία. Έπειτα, διαβάζει τα περιεχόμενά του και αφαιρεί όλους τους χαρακτήρες εκτός των παρενθέσεων. Το πρόβλημα που καλείται το πρόγραμμα να αντιμετωπίσει έγκειται στον τρόπο με τον οποίο θα αναγνωριστούν οι ανοιχτές και κλειστές παρενθέσεις λαμβάνοντας υπ’ όψη τη σειρά με την οποία βρίσκονται.

## Επίλυση

Με τη χρήση ενός Ντετερμινιστικού Αυτόματου Στοίβας (ΝΑΣ) το ζήτημα αντιμετωπίζεται αρκετά πιο εύκολα. Σύμφωνα με τον ορισμό ενός ΝΑΣ, στη δική μας περίπτωση ορίζεται ως εξής:

**Μ = (K, T, V, p, k1, A1, F)**

**Όπου:**

Σύνολο Καταστάσεων (K)  
{ **k0**, **k1**, **k2** }

Αλφάβητο Εισόδου (T)  
{ **(**, **)** }

Αλφάβητο Συμβόλων (V)  
{ **(**, **)**, **$** }

Συνάρτηση Στοίβας (p)  
f(**k0**, **ε**, **ε**) = { (**k1**, **$**) }  
f(**k1**, **ε**, **(**) = { (**k1**, **(**) }  
f(**k1**, **(**, **)**) = { (**k1**, **ε**) }  
f(**k1**, **$**, **ε**) = { (**k2**, **ε**) }

Αρχική Κατάσταση (k1)  
**k0**

Αρχικό Σύμβολο Στοίβας (A1)  
**$**

Σύνολο Τελικών Καταστάσεων (F)  
{ **t2** }

Με άλλα λόγια, κατά την έναρξη του προγράμματος, αφού διαβαστούν τα περιεχόμενα του αρχείου, η στοίβα σύμφωνα με τη συνάρτησή της, αρχικοποιείται και μεταβαίνει στην κατάσταση **k1** με το σύμβολο **$** στο πυθμένα της στοίβας. Στη συνέχεια, εάν δεχτεί η στοίβα το σύμβολο **(** εισάγει τη παρένθεση και παραμένει στην κατάσταση **k1** μέχρις ότου το σύμβολο **)** να διαβαστεί. Σε αυτή τη περίπτωση, αφαιρείται η προηγούμενη παρένθεση **(** από τη στοίβα. Συνεχίζουμε αυτή τη διαδικασία μέχρι να καταναλωθεί η συμβολοσειρά από παρενθέσεις και να μεταβούμε στη τελική κατάσταση **k2**.

Κατά την κατανάλωση της συμβολοσειράς, εάν διαπιστωθεί πως δεν υπάρχει καθορισμένη μετάβαση σύμφωνα με τη συνάρτηση της στοίβας, η διαδικασία σταματάει και οι παρενθέσεις δεν αναγνωρίζονται.

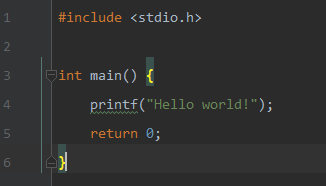
## Εκτέλεση

Το πρόγραμμα έχει υλοποιηθεί στη **Python 3** και δέχεται **1** όρισμα.

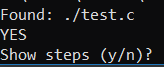
Ενδεικτικά, μέσω της γραμμής εντολών το πρόγραμμα εκτελείται ως εξής:

* Κάνουμε **cd compilers-project**
* Κάνουμε **cd exercise-1**
* Εκτελούμε το πρόγραμμα **python parentheses.py ./test.c**

Αν υποθέσουμε ότι το αρχείο **test.c** περιέχει τον πηγαίο κώδικα για ένα από hello world πρόγραμμα στη C και βρίσκεται στον ίδιο φάκελο με το αρχείο **parentheses.py**:



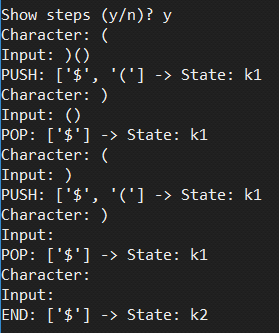
Τότε παρατηρούμε ότι το πρόγραμμά μας ανταποκρίνεται ως εξής:



Το πρόγραμμα βρήκε επιτυχώς το αρχείο **test.c** και αφού ανέλυσε τα περιεχόμενα και τις παρενθέσεις, τυπώνει «**YES**» στη γραμμή εντολών. Η ένδειξη αυτή σημαίνει πως οι παρενθέσεις που βρίσκονται στο πηγαίο κώδικα του test.c είναι σωστές.

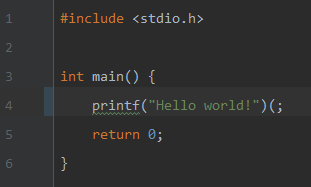
Πράγματι αφού έχουμε τις εξής παρενθέσεις: **()()**

Στη συνέχεια, παρατηρούμε πως το πρόγραμμα δίνει την επιλογή στο χρήστη να του εκτυπώσει αναλυτικά όλα τα βήματα που οδήγησαν σε αυτό το αποτέλεσμα.

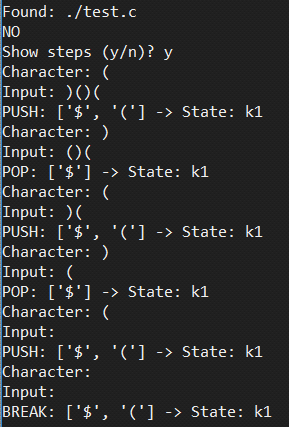


Στο τελευταίο βήμα, μπορούμε να διακρίνουμε πως η στοίβα περιέχει το σύμβολο **$** και βρίσκεται στην κατάσταση **k2**, άρα η συμβολοσειρά αναγνωρίζεται.

Σε μια άλλη περίπτωση που οι παρενθέσεις στο αρχείο του πηγαίου κώδικα δεν ήταν ισορροπημένες:



Το αποτέλεσμα θα ήταν «**ΝΟ**» αφού πράγματι θα είχαμε τις εξής παρενθέσεις: **()()(**



Έτσι, διαπιστώνουμε ότι η στοίβα παρέμεινε στη κατάσταση **k1** με μια εκκρεμής ανοιχτή παρένθεση.

# Άσκηση Β

…

# Άσκηση Γ

…

# Άσκηση Δ

…

# Άσκηση Ε

…