

A)

Στην εργασία αυτήν μας ζητήθηκε να υλοποιήσουμε με την χρήση ενός αυτόματου στοίβας ένα πρόγραμμα το οποίο θα δέχεται ως είσοδο ένα αρχείο και θα το ελέγχει ως προς την συντακτική ορθότητα των παρενθέσεων.

Το πρόγραμμα υλοποιήθηκε σε C++ ως εξής:

Το πρόγραμμα δέχεται ως είσοδο ένα αρχείο είτε ως όρισμα

```
./exercise1 test1.txt
```

είτε ζητώντας το από τον χρήστη σε περίπτωση που το ξεχάσει.

```
Insert filename: test1.txt
```

Αρχικά, εμφανίζονται οι κανόνες του ΝΑΣ.

```
M = (K, T, V, ρ, k1, $, {k2})  
K = {k1, k2}  
T = {"(", ")", ε}  
V = {A, $}
```

Στην συνέχεια αρχικοποιείται η κατάσταση με την τιμή 1, δημιουργείται μια στοίβα και τοποθετείται σε αυτήν το '\$'. Έπειτα το πρόγραμμα διαβάζει το αρχείο και κάθε φορά που συναντάει τον χαρακτήρα '(' ή ')' τον προσθέτει σε μια λίστα (σύμβολα εισόδου). Το πρόγραμμα διαβάζει κάθε χαρακτήρα από την λίστα και αν είναι '(' προσθέτει στην στοίβα τον χαρακτήρα 'A' αλλιώς αν είναι ')' αφαιρεί τον χαρακτήρα 'A' αν υπάρχει, διαφορετικά η ανάγνωση σταματά. Ο αριθμός των 'A' αντιπροσωπεύει τον αριθμό των αριστερών παρενθέσεων '(' που δεν έχουν ταιριάξει με μια δεξιά παρένθεση ')'.

Επομένως, υπάρχουν 3 περιπτώσεις:

- Να διαβαστούν όλα τα σύμβολα εισόδου αλλά να υπάρχουν ακόμα 'Α' στην στοίβα, πράγμα το οποίο σημαίνει ότι υπάρχουν αριστερές παρενθέσεις '(' που δεν έχουν κλείσει με μια δεξιά παρένθεση ')'. Σε αυτήν την περίπτωση εκτυπώνεται η ένδειξη 'NO'.

Παράδειγμα: (aaaabbbbccccc)(d)((

```
M = (K, T, V, p, k1, $, {k2})  
K = {k1, k2}  
T = {"(", ")", "\n", ε}  
V = {A, $}  
  
NO  
  
Print steps? (y/n): y  
  
Stack: $           State: k1           Input: ()()()  
Stack: $A          State: k1           Input: )()()  
Stack: $            State: k1           Input: ()()  
Stack: $A           State: k1           Input: )()  
Stack: $             State: k1           Input: (  
Stack: $A            State: k1           Input: (  
Stack: $$A           State: k1           Input: )  
Stack: $A            State: k1           Input: ε
```

- Το σύμβολο εισόδου να είναι ‘)’ ενώ δεν υπάρχουν ‘(’ στην στοίβα, πράγμα που σημαίνει ότι έχουν διαβαστεί περισσότερες δεξιές παρενθέσεις ‘)’ από ότι αριστερές ‘(’ . Σε αυτήν την περίπτωση διαβάζεται ο χαρακτήρας ‘\$’ από την στοίβα, η ανάγνωση διακόπτεται και εκτυπώνεται η ένδειξη ‘NO’ .

Παράδειγμα: () (aaaabbbbcccc) (d) ()

```
M = (K, T, V, p, k1, $, {k2})
K = {k1, k2}
T = {"(", ")", "ε"}
V = {A, $}

NO

Print steps? (y/n): y

Stack: $           State: k1           Input: ())()()((
Stack: $A          State: k1           Input: ))()()((
Stack: $           State: k1           Input: ))()()((
```

- Να διαβαστούν όλα τα σύμβολα εισόδου και στην στοίβα να υπάρχει μόνο ο χαρακτήρας '\$', πράγμα το οποίο σημαίνει ότι το αρχείο είναι συντακτικά ορθό ως προς τις παρενθέσεις. Σε αυτήν την περίπτωση εκτυπώνεται η ένδειξη 'YES'.

Παράδειγμα: $(())(aaaa(bbbb)cccc)(d)((e))$

```
M = (K, T, V, p, k1, $, {k2})
K = {k1, k2}
T = {"(", ")", "ε"}
V = {A, $}

YES

Print steps? (y/n): y

Stack: $           State: k1           Input: (()())()((()
Stack: $A          State: k1           Input: ())()()((()
Stack: $AA         State: k1           Input: ))()()()((()
Stack: $A          State: k1           Input: ))()()()((()
Stack: $           State: k1           Input: ())()()((()
Stack: $A          State: k1           Input: ())()()((()
Stack: $AA         State: k1           Input: ))()()()((()
Stack: $A          State: k1           Input: ))()()()((()
Stack: $           State: k1           Input: ())()()((()
Stack: $A          State: k1           Input: ))()()()((()
Stack: $           State: k1           Input: ())()()((()
Stack: $A          State: k1           Input: ))()()()((()
Stack: $AA         State: k1           Input: ))()()()((()
Stack: $A          State: k1           Input: ))()()()((()
Stack: $           State: k1           Input: ε
Stack: $           State: k2           Input: ε
```

Τέλος, δίνεται η επιλογή να εκτυπωθεί η αλληλουχία βημάτων που οδήγησε το ΝΑΣ στην αναγνώριση ή την απόρριψη της έκφρασης.

```
Print steps? (y/n):
```