

**(Α) – 1 μονάδα:** Υλοποιήστε σε γλώσσα προγραμματισμού της επιλογής σας, Ντετερμινιστικό Αυτόματο Στοιβάς (ΝΑΣ) που μοντελοποιεί εκφράσεις αποτελούμενες μόνο από τους χαρακτήρες 'x' και 'y', κατά τρόπον ώστε:

- όσοι χαρακτήρες 'x' εμφανίζονται συνολικά, άλλοι τόσοι χαρακτήρες 'y' εμφανίζονται.
- κοιτάζοντας την έκφραση από αριστερά προς τα δεξιά, οι χαρακτήρες 'y' δεν είναι ποτέ περισσότεροι από τους χαρακτήρες 'x'.

Για παράδειγμα, η έκφραση xxyxy αναγνωρίζεται από το ΝΑΣ, αλλά η xyxy δεν αναγνωρίζεται.

Το πρόγραμμά σας θα δέχεται είσοδο μία έκφραση κάθε φορά και θα επιστρέφει την ένδειξη «YES» ή «NO», ανάλογα με το αν η συμβολοσειρά αναγνωρίζεται ή όχι. Επίσης θα τυπώνεται η αλληλουχία βημάτων που οδήγησαν στην αναγνώριση (ή στην απόρριψη) της έκφρασης. Σε κάθε γραμμή εκτύπωσης θα παρουσιάζονται τα περιεχόμενα της στοιβάς, η τρέχουσα κατάσταση και τα υπόλοιπα σύμβολα εισόδου.

**(Β) – 1 μονάδα:** Υλοποιήστε σε γλώσσα προγραμματισμού της επιλογής σας μια γεννήτρια συμβολοσειρών για τη γραμματική που ορίζεται με τους εξής κανόνες παραγωγής:

$\langle \text{έκφραση} \rangle ::= (\langle \text{υποέκφραση} \rangle)$

$\langle \text{υποέκφραση} \rangle ::= \langle \text{στοιχείο1} \rangle \langle \text{στοιχείο2} \rangle$

$\langle \text{στοιχείο1} \rangle ::= v | \langle \text{έκφραση} \rangle$

$\langle \text{στοιχείο2} \rangle ::= - \langle \text{υποέκφραση} \rangle | + \langle \text{υποέκφραση} \rangle | \varepsilon$

Το πρόγραμμά σας θα πρέπει να τυπώνει τα βήματα της εκάστοτε παραγωγής.

**(Γ) – 1 μονάδα:** Δίνεται η γραμματική G:

S  $\rightarrow$  (X)

X  $\rightarrow$  YZ

Y  $\rightarrow$  a | β | S

Z  $\rightarrow$  \*X | -X | +X | ε

Εξετάστε αν η γραμματική είναι LL(1). Αν αυτό ισχύει, τότε κατασκευάστε σε γλώσσα προγραμματισμού C, C++ ή Java, συντακτικό αναλυτή top-down που αναγνωρίζει (παράγει) την εκάστοτε δοθείσα συμβολοσειρά (αλυσίδα εφαρμογής κανόνων) ή απαντά αρνητικά ως προς τη συντακτική της ορθότητα. Να γίνει επίδειξη για την έκφραση ((α-β)\*(β+α)). Αν θεωρείτε ότι η γραμματική δεν είναι LL(1), αποδείξτε το.

**(Δ) – 1 μονάδα:** Μια κανονική έκφραση (χωρίς την χρήση παρενθέσεων) σε μια γλώσσα προγραμματισμού ορίζεται ως εξής:

- Στην αρχή πρέπει να εμφανίζεται το όνομα μιας μεταβλητής.

- ii. Στη συνέχεια, θα πρέπει να ακολουθεί το σύμβολο "=".
- iii. Μετά ακολουθεί το όνομα μιας μεταβλητής ή ένας αριθμός από 1 έως και 9, ύστερα ένα σύμβολο από τα εξής "+", "-", "\*", "/", "%" και έπειτα ξανά ακολουθεί το όνομα μιας μεταβλητής ή ένας αριθμός από 1 έως και 9.
- iv. Το (iii) μπορεί να επαναληφθεί όσες φορές επιθυμούμε.
- v. Η κανονική έκφραση τελειώνει με τον χαρακτήρα ";".

Παραδείγματα έγκυρων κανονικών εκφράσεων με βάση τα παραπάνω είναι τα εξής:

$X = a + 3 * b;$

$Y = 5 \% 2;$

$Z = 10 / 2 - a * b;$

Για τον ορισμό που περιγράφηκε να γίνει:

- a) Το συντακτικό διάγραμμα.
- b) Η περιγραφή BNF
- c) Η περιγραφή EBNF
- d) Το πρόγραμμα Flex