

# ΤΕΚΜΗΡΙΩΣΗ

Το ζητούμενο της άσκησης αυτής είναι η κατασκευή του συντακτικού διαγράμματος, η περιγραφή BNF, η περιγραφή EBNF και το σχετικό πρόγραμμα Flex για τον ορισμό μιας κανονικής έκφρασης (χωρίς την χρήση παρενθέσεων) σε μια γλώσσα προγραμματισμού δεδομένου ότι ορίζεται ως εξής:

- (1) Στην αρχή πρέπει να εμφανίζεται το όνομα μιας μεταβλητής.
- (2) Στη συνέχεια, θα πρέπει να ακολουθεί το σύμβολο “=”.
- (3) 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.
- (4) Το (3) μπορεί να επαναληφθεί όσες φορές επιθυμούμε.
- (5) Η κανονική έκφραση τελειώνει με τον χαρακτήρα “;”

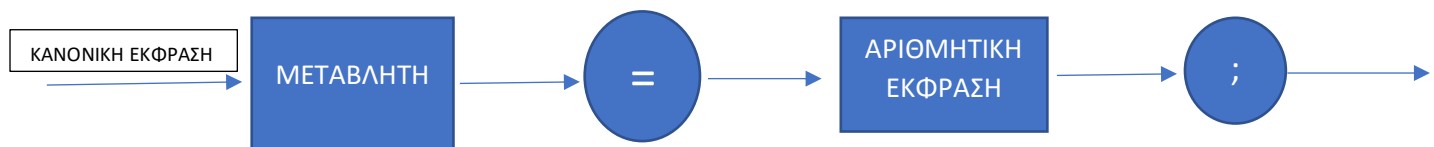
Με μια περαιτέρω ανάλυση προκύπτει ότι ο ορισμός μιας κανονικής έκφρασης (χωρίς την χρήση παρενθέσεων) σε μια γλώσσα προγραμματισμού:

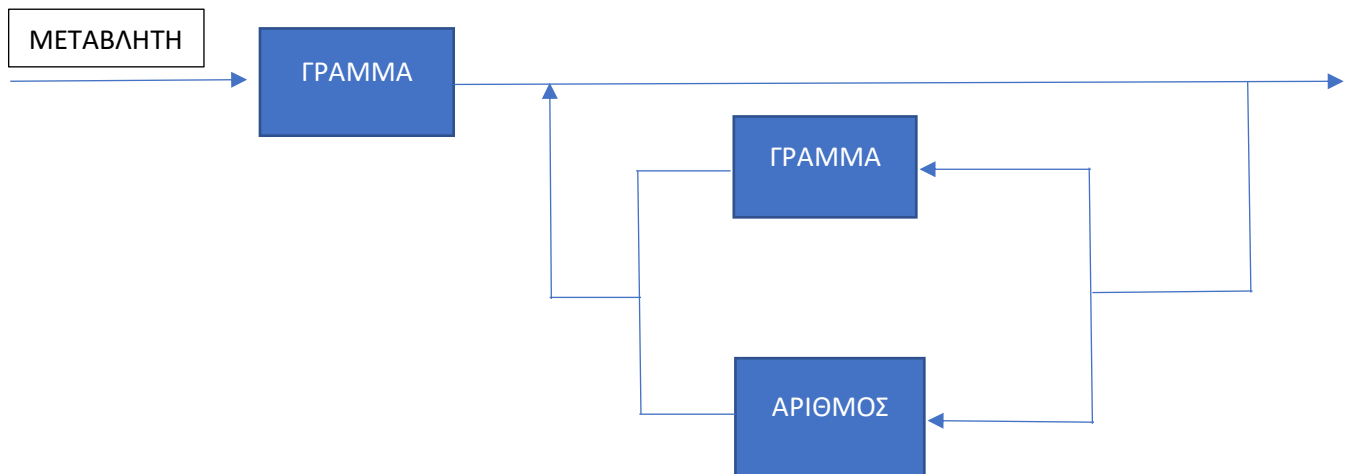
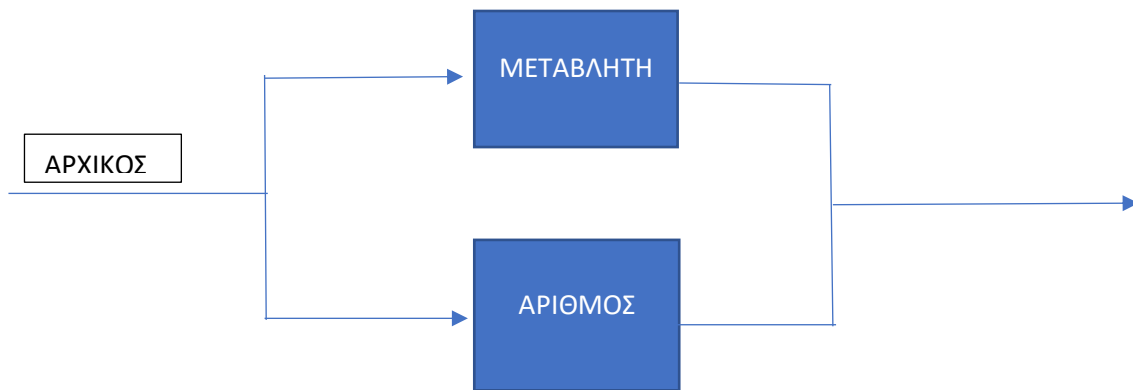
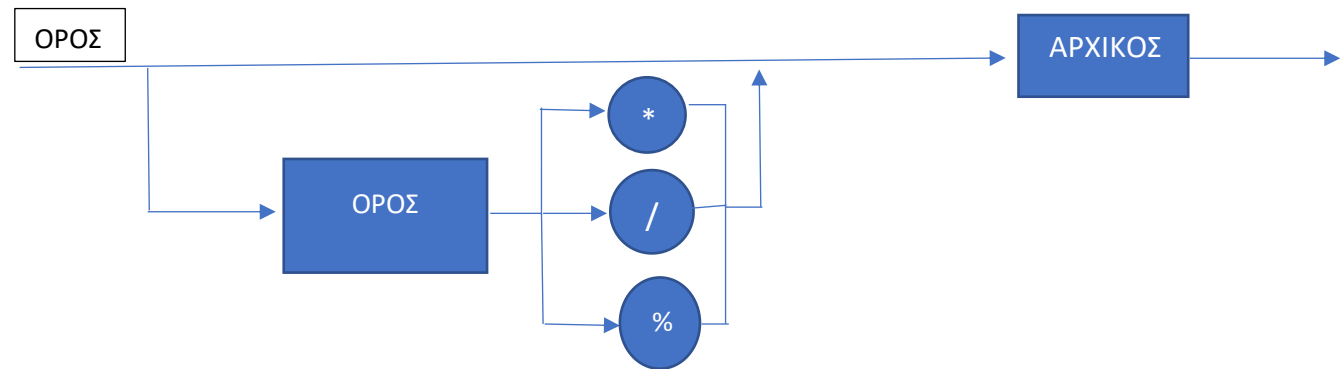
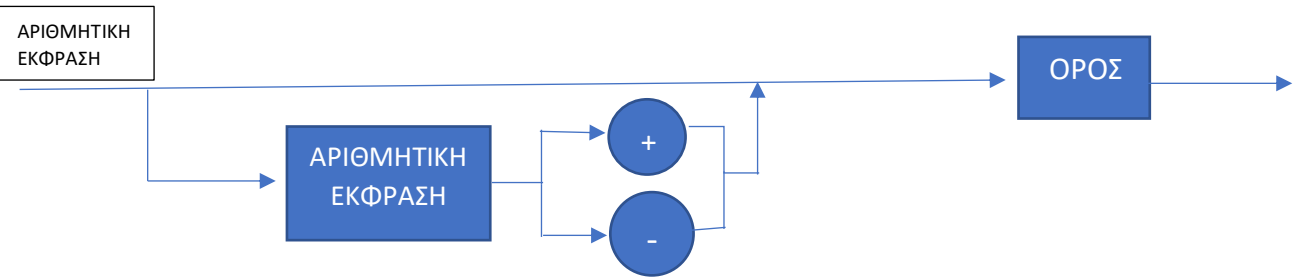
- Αποτελείται από μια μεταβλητή ακολουθούμενη από το τερματικό σύμβολο “=”, μια αριθμητική έκφραση και τέλος το τερματικό σύμβολο “;”.
- Αντίστοιχα μια μεταβλητή ξεκινάει από ένα γράμμα και μπορεί να ακολουθούν άλλα γράμματα και αριθμοί.
- Επίσης μια αριθμητική έκφραση αποτελείται από προσθαφαιρέσεις όρων.
- Ο όρος αποτελείται από πολλαπλασιασμούς, διαιρέσεις και υπόλοιπα διαιρέσεων(modulo) αρχικών όρων.
- Και τέλος ένας αρχικός όρος αποτελείται από μια μεταβλητή ή έναν αριθμό.

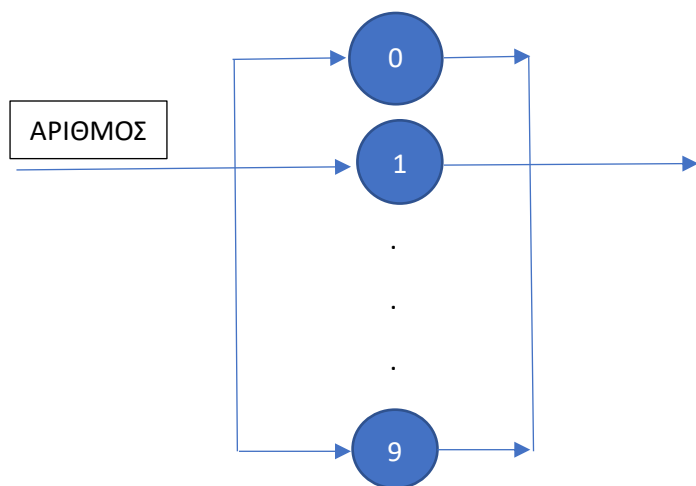
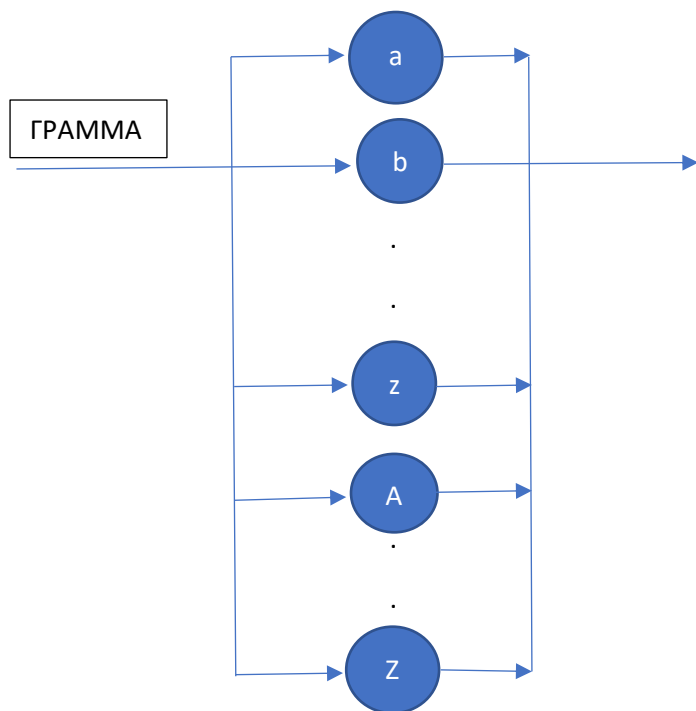
## ΕΡΩΤΗΜΑ 1: ΣΥΝΤΑΚΤΙΚΟ ΔΙΑΓΡΑΜΜΑ

- Μοιάζει με BNF αλλά είναι γραφικός τρόπος.
- Αυτός ο τρόπος έχει χρησιμοποιηθεί για να παραστήσει πολλές γλώσσες προγραμματισμού.
- Ο πλήρης συντακτικός ορισμός φτιάχνεται από ένα σύνολο συντακτικών διαγραμμάτων.
- Τα σύμβολα που χρησιμοποιούνται είναι τετράγωνα και κύκλοι.
- Κάθε κουτί (■) περιέχει το όνομα ενός μη-τερματικού συμβόλου το οποίο ορίζεται από άλλο συντακτικό διάγραμμα.
- Κάθε στρογγυλό (●) περιέχει τερματικά σύμβολα.
- Οποιαδήποτε διαδρομή μέσα στο συντακτικό διάγραμμα που ακολουθεί τα βέλη, αναλύοντας τα μη-τερματικά σύμβολα όπως προκύπτουν μας δίνουν μια συντακτικά σωστή παραγωγή της γλώσσας.

Σύμφωνα με τα παραπάνω προκύπτει το συντακτικό διάγραμμα για μια κανονική έκφραση:







## ΕΡΩΤΗΜΑ 2: ΠΕΡΙΓΡΑΦΗ BNF (Backus-Naur Form)

- Είναι μια μεταγλώσσα που χρησιμοποιείται συχνά για τον ορισμό σύνταξης μιας γλώσσας προγραμματισμού.
- Είναι μια τεχνική για να παριστάνουμε κανόνες που χρησιμοποιούνται για την παραγωγή προτάσεων της γλώσσας.
- Εάν το σύνολο των κανόνων είναι πεπερασμένο τότε αυτό το σύνολο κανόνων μπορεί να αποτελέσει έναν αυστηρό ορισμό της σύνταξης της γλώσσας.
- Η BNF αντιστοιχεί στις γλώσσες χωρίς συμφραζόμενα.

### Αντιστοιχία συμβολισμού BNF με τον ορισμό των γραμματικών

- Γωνιακές παρενθέσεις  $\langle \rangle$  περικλείουν τα μη-τερματικά σύμβολα.
- Το  $::=$  αντιστοιχεί στο  $\rightarrow$ . Δηλ. μπορεί να ακολουθείται από οσαδήποτε δεξιά μέλη συντακτικών κανόνων.
- Το  $|$  χωρίζει τα δεξιά μέλη μεταξύ τους όταν σε ένα αριστερό μέλος αντιστοιχούν περισσότεροι από ένας συντακτικοί κανόνες.

## Σύμφωνα με τα παραπάνω προκύπτει η περιγραφή BNF για μια κανονική έκφραση:

$\langle \text{ΚΑΝΟΝΙΚΗ ΕΚΦΡΑΣΗ} \rangle ::= \langle \text{ΜΕΤΑΒΛΗΤΗ} \rangle = \langle \text{ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ} \rangle;$

$\langle \text{ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ} \rangle ::= \langle \text{ΟΡΟΣ} \rangle \mid \langle \text{ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ} \rangle + \langle \text{ΟΡΟΣ} \rangle \mid$   
 $\langle \text{ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ} \rangle - \langle \text{ΟΡΟΣ} \rangle$

$\langle \text{ΟΡΟΣ} \rangle ::= \langle \text{ΑΡΧΙΚΟΣ} \rangle \mid \langle \text{ΟΡΟΣ} \rangle * \langle \text{ΑΡΧΙΚΟΣ} \rangle \mid \langle \text{ΟΡΟΣ} \rangle / \langle \text{ΑΡΧΙΚΟΣ} \rangle \mid \langle \text{ΟΡΟΣ} \rangle \%$   
 $\langle \text{ΑΡΧΙΚΟΣ} \rangle$

$\langle \text{ΑΡΧΙΚΟΣ} \rangle ::= \langle \text{ΜΕΤΑΒΛΗΤΗ} \rangle \mid \langle \text{ΑΡΙΘΜΟΣ} \rangle$

$\langle \text{ΜΕΤΑΒΛΗΤΗ} \rangle ::= \langle \text{ΓΡΑΜΜΑ} \rangle \mid \langle \text{ΜΕΤΑΒΛΗΤΗ} \rangle \langle \text{ΓΡΑΜΜΑ} \rangle \mid$   
 $\langle \text{ΜΕΤΑΒΛΗΤΗ} \rangle \langle \text{ΑΡΙΘΜΟΣ} \rangle$

$\langle \text{ΓΡΑΜΜΑ} \rangle ::= a \mid b \mid c \mid d \mid \dots \mid A \mid B \mid C \mid D \mid \dots$

$\langle \text{ΑΡΙΘΜΟΣ} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

## **ΕΡΩΤΗΜΑ 3: ΠΕΡΙΓΡΑΦΗ EBNF (Extended BNF)**

- ❖ Η μεγαλύτερη διαφορά από την BNF είναι η χρήση μετασυμβόλων, δηλαδή συμβόλων που χρησιμοποιούνται στη μεταγλώσσα για να σημαίνουν κάτι συγκεκριμένο.

## EBNF: Συμβολισμοί

- ❖ Τα τερματικά σύμβολα μπαίνουν σε “ ” (οπότε τα μη τερματικά σύμβολα δεν χρειάζεται να είναι σε <>).
- ❖ Υπάρχει τελεία στο τέλος κάθε συντακτικού κανόνα.
- ❖ Το σύμβολο ::= μπορεί να αντικατασταθεί με =.
- ❖ Σύμβολα που μπορούν να παραλειφθούν μπαίνουν σε αγκύλες [ ].
- ❖ Οι αγκύλες σημαίνουν ότι το περιεχόμενό τους μπορεί να εμφανιστεί 0 φορές ή 1 φορά (δηλ. μπορεί να παραλειφθεί).
- ❖ Σύμβολα που μπορούν να παραλειφθούν ή να επαναληφθούν οσοδήποτε φορές μπαίνουν σε άγκιστρα { }.
- ❖ Τα άγκιστρα σημαίνουν ότι το περιεχόμενό τους μπορεί να εμφανιστεί 0 φορές ή περισσότερες φορές.
- ❖ Παρενθέσεις επιτρέπονται για την ομαδοποίηση συμβόλων.

### Σύμφωνα με τα παραπάνω προκύπτει η περιγραφή EBNF για μια κανονική έκφραση:

ΚΑΝΟΝΙΚΗ ΕΚΦΡΑΣΗ = (ΜΕΤΑΒΛΗΤΗ “=” ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ “;” ) .

ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ = [ΑΡΙΘΜΗΤΙΚΗ ΕΚΦΡΑΣΗ (“+” | “-”)] ΟΡΟΣ.

ΟΡΟΣ = [ΟΡΟΣ (“\*” | “/” | “%”)] ΑΡΧΙΚΟΣ.

ΑΡΧΙΚΟΣ = ΜΕΤΑΒΛΗΤΗ | ΑΡΙΘΜΟΣ.

ΜΕΤΑΒΛΗΤΗ = ΓΡΑΜΜΑ {ΓΡΑΜΜΑ | ΑΡΙΘΜΟΣ}.

ΓΡΑΜΜΑ = “a” | “b” | “c” | “d” | ... | “A” | “B” | “C” | “D” | ... .

ΑΡΙΘΜΟΣ = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” .

## ΕΡΩΤΗΜΑ 4: ΠΡΟΓΡΑΜΜΑ FLEX

### Επισημάνσεις

✓ Στο τμήμα δηλώσεων έχουμε δηλώσει:

- Τη μεταβλητή NUMBER η οποία αποτελείται από έναν αριθμό από το 1 έως και το 9.
- Τη μεταβλητή VAR (δηλαδή το όνομα από το οποίο θα ξεκινάει μια μεταβλητή στην κανονική μας έκφραση) η οποία μπορεί να είναι οποιοδήποτε αλφαριθμητικό στοιχείο που ξεκινάει από γράμμα κεφαλαίο η πεζό και στη συνέχεια ακολουθούν αλλά γράμματα και αριθμοί.
- Επίσης έχουμε δηλώσει τον αρχικό όρο PRIMARY που θα είναι μια μεταβλητή(VAR) ή ένας αριθμός(NUMBER).
- Τη μεταβλητή TERM (δηλαδή τον όρο) που απαρτίζεται είτε από μόνο έναν αρχικό όρο PRIMARY (δηλαδή μια μεταβλητή ή έναν αριθμό) ή από πολλαπλασιασμούς ("\*"), διαιρέσεις("/") και υπόλοιπα διαιρέσεων ("%") αρχικών όρων PRIMARY.
- Τη μεταβλητή EXPRESSION δηλαδή την έκφραση που βρίσκεται στο δεξί μέλος (μετά το "=") και αποτελείται είτε από μόνο έναν όρο TERM ή από προσθαιρέσεις όρων TERM.

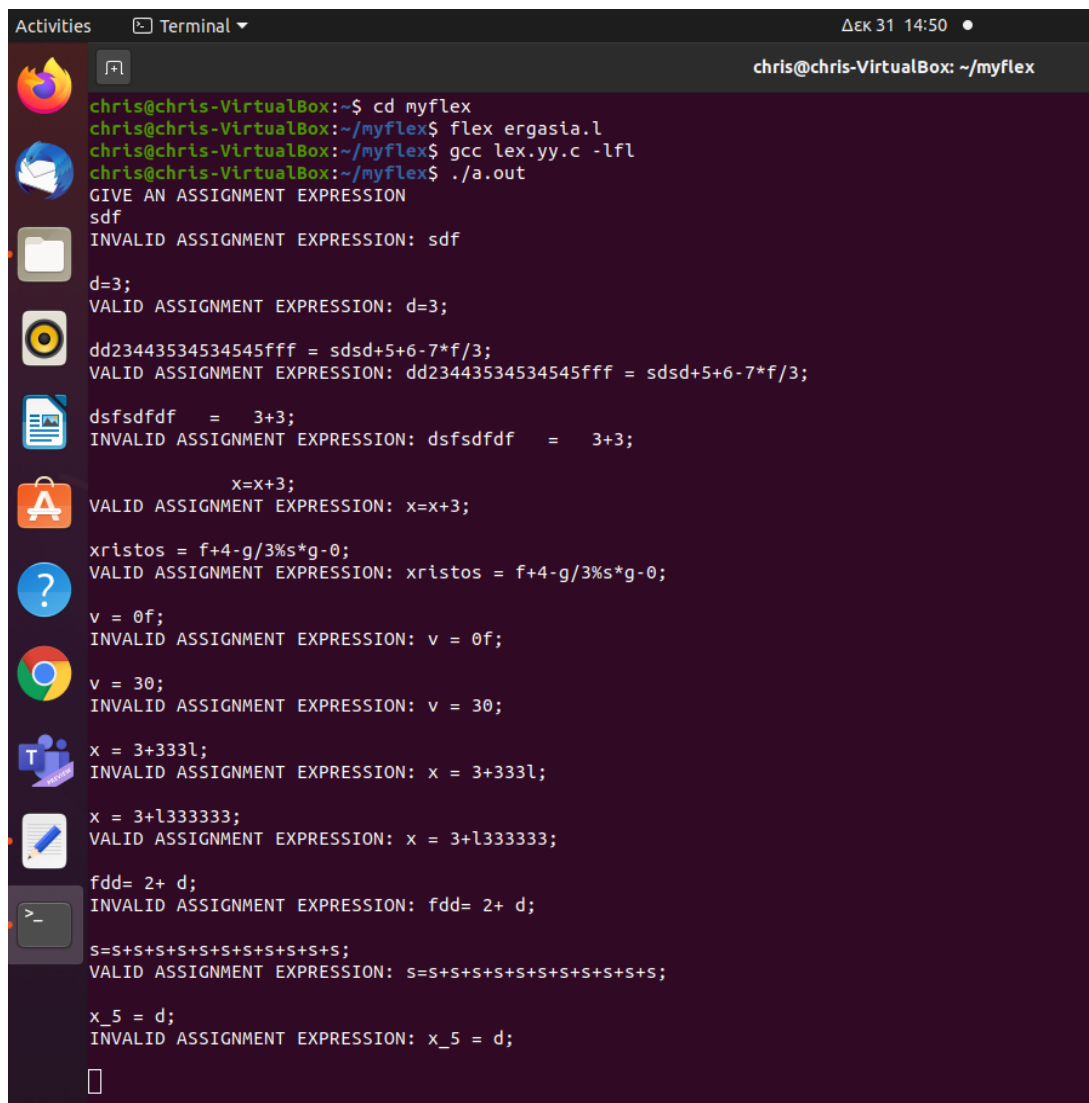


- Τέλος τη μεταβλητή ASSIGNMENT η οποία θα αναπαριστά μια έγκυρη κανονική έκφραση(ανάθεσης τιμής σε μια μεταβλητή). Στο αριστερό της μέρος (πριν το “=”) θα υπάρχει μια μεταβλητή δηλαδή μια VAR, στη συνέχεια, θα ακολουθεί το σύμβολο “=” και μετά στο δεξί της μέρος η έκφραση EXPRESSION όπως ορίστηκε προηγουμένως. Τέλος το σύμβολο “;” το οποίο σηματοδοτεί το τέλος της κανονικής έκφρασης. Πριν και μετά την δήλωση ASSIGNMENT μπορούν να υπάρχουν άπειρα κενά ενώ πριν και μετά το “=” μπορεί να υπάρχει το πολύ ένα κενό έτσι ώστε να υφίσταται ευελιξία στο πως θα δίνει την κανονική έκφραση ο χρήστης ως είσοδο.

#### ✓ Στο τμήμα κανόνων:

- Ο πρώτος κανόνας ενεργοποιείται όποτε βρεθεί κάποια έγκυρη κανονική έκφραση ανάθεσης τιμής, δηλαδή ένα ASSIGNMENT το οποίο αναγνωρίζει έγκυρες κανονικές εκφράσεις. Τότε εμφανίζουμε στον χρήστη το μήνυμα “VALID ASSIGNMENT EXPRESSION:” μαζί με την έγκυρη κανονική έκφραση που ανιχνεύθηκε.
- Ο δεύτερος κανόνας απαλείφει τα κενά και τις νέες γραμμές έτσι ώστε να είναι πιο εύκολη η αναγνώριση κάποιας έγκυρης κανονικής έκφρασης.
- Τέλος ο τρίτος κανόνας εφαρμόζεται όταν η είσοδος που δίνει ο χρήστης δεν είναι κάποια έγκυρη κανονική έκφραση και τότε εμφανίζουμε το μήνυμα “INVALID ASSIGNMENT EXPRESSION:” μαζί με την μη έγκυρη κανονική έκφραση που έδωσε ο χρήστης.

## ΠΑΡΑΔΕΙΓΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ:



```
chris@chris-VirtualBox: ~$ cd myflex
chris@chris-VirtualBox:~/myflex$ flex ergasia.l
chris@chris-VirtualBox:~/myflex$ gcc lex.yy.c -lfl
chris@chris-VirtualBox:~/myflex$ ./a.out
GIVE AN ASSIGNMENT EXPRESSION
sdf
INVALID ASSIGNMENT EXPRESSION: sdf
d=3;
VALID ASSIGNMENT EXPRESSION: d=3;
dd2344353453454545fff = sdsd+5+6-7*f/3;
VALID ASSIGNMENT EXPRESSION: dd2344353453454545fff = sdsd+5+6-7*f/3;
dsfsdfdf = 3+3;
INVALID ASSIGNMENT EXPRESSION: dsfsdfdf = 3+3;
x=x+3;
VALID ASSIGNMENT EXPRESSION: x=x+3;
xristos = f+4-g/3*s*g-0;
VALID ASSIGNMENT EXPRESSION: xristos = f+4-g/3*s*g-0;
v = 0f;
INVALID ASSIGNMENT EXPRESSION: v = 0f;
v = 30;
INVALID ASSIGNMENT EXPRESSION: v = 30;
x = 3+333l;
INVALID ASSIGNMENT EXPRESSION: x = 3+333l;
x = 3+l333333;
VALID ASSIGNMENT EXPRESSION: x = 3+l333333;
fdd= 2+ d;
INVALID ASSIGNMENT EXPRESSION: fdd= 2+ d;
s=s+s+s+s+s+s+s+s+s;
VALID ASSIGNMENT EXPRESSION: s=s+s+s+s+s+s+s+s+s;
x_5 = d;
INVALID ASSIGNMENT EXPRESSION: x_5 = d;
```

## Οδηγίες χρήσης

Μπορείτε να τρέξετε το αρχείο “ASKHSH4.l” σε περιβάλλον Linux αποθηκεύοντας το στον προσωπικό σας κατάλογο και τρέχοντας το στο terminal με τις παρακάτω εντολές:

```
flex ASKHSH.l
```

```
gcc lex.yy.c -lfl
```

```
./a.out
```