

Εργαστήριο #3: Δεύτερη Ομάδα στις Ασκήσεις Λεξικής Ανάλυσης με προτεινόμενες λύσεις

Άσκηση 3-1.

Να γίνει λεξικός αναλυτής με το Flex που να εμφανίζει οποιοδήποτε χαρακτήρα εισάγεται από το χρήστη με εξαίρεση τη λέξη "username". Όταν εμφανισθεί η λέξη αυτή θα πρέπει να εμφανίζεται αντί αυτής της λέξης το όνομα χρήστη. Το όνομα χρήστη εμφανίζεται με τη κλήση της συνάρτησης "getlogin()". Σημειώστε πως το flex όταν δεν έχει κάποιο κανόνα για έναν χαρακτήρα απλά τον εμφανίζει.

```
%option noyywrap
%{
#include <stdio.h>
%}
%%
"username" {printf("%s", getlogin());}
%%
int main(int argc, char *argv[])
{
yylex();
return 0;
}
```

Άσκηση 3-2.

Να γίνει λεξικός αναλυτής με το Flex που αναγνωρίζει αριθμούς ακέραιους και πραγματικούς. Χρησιμοποιείτε και δήλωση DIGIT στο τμήμα των ορισμών για τον ορισμό ενός ψηφίου (0-9). Στη συνέχεια στο τμήμα των κανόνων χρησιμοποιείτε τη δήλωση DIGIT για τον κανόνα αναγνώρισης ακεραίων αλλά και πραγματικών αριθμών. Όταν αναγνωρίζει έναν ακέραιο ή έναν πραγματικό αριθμό θα πρέπει να εμφανίζει τον δεκαπλάσιο αριθμό από αυτόν που αναγνώρισε ως εξής:

```
10
0 δεκαπλάσιος ακέραιος: 10 (100)
10.23
0 πραγματικός: 10.23 (102.3)
something else
<δεν εμφανίζει τίποτε και αγνοείται η φράση 'something else'>
```

Σημείωση 1: Μετατροπή του αλφαριθμητικού s σε ακέραιο --> atoi(s)

Σημείωση 2: Μετατροπή του αλφαριθμητικού s σε πραγματικό --> atof(s)

```
%option noyywrap
%{
/*
χρειάζεται η math.h για τη κλήση της atoi() & atof()
που μετατρέπει ένα αλφαριθμητικό σε float
*/
#include <stdio.h>
#include <math.h>
%}
DIGIT [0-9]
%%
{DIGIT}+ {
printf( "0 δεκαπλάσιος ακέραιος: %s (%d)\n", yytext,
10*atoi( yytext ) );
```

```

    }
    {DIGIT}+("."{DIGIT}+)? { printf( "Ο πραγματικός: %s (%g)\n",
                                   yytext, 10*atof( yytext ) );
    }
. { }
%%
int main( int argc, char **argv )
{
    yylex();
}

```

Άσκηση 3-3.

Να γίνει λεξικός αναλυτής με το Flex που αναγνωρίζει:

Ακεραίους: σειρές ψηφίων 0-9 (π.χ. 123),

Πραγματικούς: σειρές ψηφίων 0-9 αλλά και με τελεία ανάμεσα από το ακέραιο και δεκαδικό μέρος (π.χ. 12.24)

Τις δεσμευμένες λέξεις: if, then, begin, end, procedure, function

Αναγνωριστικά: λέξεις που ξεκινάνε από μικρό γράμμα και ακολουθούνται από μηδέν ή περισσότερα μικρά γράμματα ή ψηφία.

Αγνοούνται τα κενά, tab και οι αλλαγές γραμμών

Για οποιοδήποτε άλλο χαρακτήρα εμφανίζεται το μήνυμα "Άγνωστος χαρακτήρας"

```

10
0 ακέραιος: 10 (10)
10.23
0 πραγματικός: 10.23 (10.23)
if
Λέξη-κλειδί: if
a101
Ένα αναγνωριστικό: a101
dd
Ένα αναγνωριστικό: dd

```

```

%option noyywrap
%{
/*
χρειάζεται η math.h για τη κλήση της atoi() & atof()
που μετατρέπει ένα αλφαριθμητικό σε float
*/
#include <math.h>
%}
DIGIT [0-9]
ID [a-z][a-z0-9]*
%%
{DIGIT}+ {
    printf( "Ο ακέραιος: %s (%d)\n",
           yytext,
           atoi( yytext ) );
}
{DIGIT}+("."{DIGIT}+)? {
    printf( "Ο πραγματικός: %s (%g)\n",
           yytext,
           atof( yytext ) );
}
if|then|begin|end|procedure|function {
    printf( "Λέξη-κλειδί: %s\n", yytext );
}

```

```
{ID} { printf( "Ένα αναγνωριστικό: %s\n", yytext ); }  
[ \t\n]+ /* αγνόησε τα κενά */  
. printf( "Άγνωστος χαρακτήρας: %s\n", yytext );  
%%  
int main( int argc, char **argv )  
{  
  yylex();  
}
```