

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών
ΣΕΤ ΑΣΚΗΣΕΩΝ #2 ΚΑΙ ΛΥΣΕΙΣ (V1.0)

FLEX-JFLEX

Γκόγκος Χρήστος

5/11/2019

Ερώτημα 1

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα που για την είσοδο την οποία θα δέχεται, σε όλες τις περιπτώσεις εμφάνισης των λέξεων alice και bob θα ανταλλάσει την μια με την άλλη λέξη. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα καθώς και την εντολή εκτέλεσης του κώδικα με είσοδο το αρχείο data1.txt.

Λύση

```
%%  
bob      printf("alice");  
alice    printf("bob");  
%%  
  
int main()  
{  
    yylex();  
}
```

Listing 1: example1.lex

```
$ flex -o example1.c example1.lex  
$ gcc -o example1 example1.c -lfl  
$ ./example1 < data1.txt
```

Listing 2: Εντολές δημιουργίας εκτελέσιμου κώδικα

```
ή  
$ flex example1.lex  
$ gcc lex.yy.c -lfl  
$ ./a.out < data1.txt
```

Listing 3: Εναλλακτικές εντολές δημιουργίας εκτελέσιμου κώδικα

Παρατήρηση: Σε περίπτωση που χρησιμοποιείται υπολογιστής Mac οι εντολές μεταγλώττισης με το gcc θα πρέπει η βιβλιοθήκη που συνδέεται να είναι η l και όχι η fl.

Ερώτημα 2

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα που για την είσοδο την οποία θα δέχεται, θα επιστρέφει τον αριθμό από χαρακτήρες και τον αριθμό από γραμμές του κειμένου. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα καθώς και την εντολή εκτέλεσης του κώδικα με είσοδο το αρχείο data2.txt.

Λύση

```
%{  
    int chars=0;  
    int lines=0;  
}%  
%%
```

```
.    { chars++;}
\n   { lines++;chars++;}

%%

int main()
{
    yylex();
    printf("characters: %i, lines: %i\n", chars, lines);
}
```

Listing 4: example2.lex

```
$ flex -o erotima2.c erotima2.lex
$ gcc -o erotima2 erotima2.c -lfl
$ ./erotima2 < data2.txt
```

Listing 5: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 3

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα που για την είσοδο την οποία θα δέχεται, θα επιστρέφει τον αριθμό λέξεων του κειμένου. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα καθώς και την εντολή εκτέλεσης του κώδικα με είσοδο το αρχείο data3.txt.

Λύση

```
SPACE    [\t\r\n]
WORD     [^ \t\r\n]+
```

```
%{
    int words=0;
}%

%%
{SPACE} { }
{WORD}  words++;
%%

int main()
{
    yylex();
    printf("words=%i\n", words);
}
```

Listing 6: erotima3.lex

```
$ flex -o erotima3.c erotima3.lex
$ gcc -o erotima3 erotima3.c -lfl
$ ./erotima3 < data3.txt
```

Listing 7: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 4

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα που για την είσοδο την οποία θα δέχεται, θα επιστρέφει το ίδιο κείμενο με κάθε αριθμό που εντοπίζει αυξημένο κατά 1. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα καθώς και την εντολή εκτέλεσης του κώδικα με είσοδο το αρχείο data1.txt.

Λύση

```
DIGIT    [0-9]
```

```
%%
{DIGIT}+ {
    int i = atoi(yytext);
    printf("%i", i+1);
}

%%

int main()
{
    yylex();
}
```

Listing 8: erotima4.lex

```
$ flex -o erotima4.c erotima4.lex
$ gcc -o erotima4 erotima4.c -lflex
$ ./erotima4 < data4.txt
```

Listing 9: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 5

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα που για την είσοδο την οποία θα δέχεται, θα αντιστρέφει όλες τις λέξεις του κειμένου. Για παράδειγμα για το κείμενο “the quick brown fox” θα πρέπει να επιστρέφει “eht kciuq nworb xof”. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα καθώς και την εντολή εκτέλεσης του κώδικα με είσοδο το αρχείο data5.txt.

Λύση

```
WORD     [^ \t\r\n]+

%%
{WORD}   {
    for(int i=yytext[0]; i<yytext[1]; i++){
        printf("%c", yytext[i]);
    }
}

%%

int main()
{
    yylex();
}
```

```
}
```

Listing 10: erotima5.lex

```
$ flex -o erotima5.c erotima5.lex
$ gcc -o erotima5 erotima5.c -lfl
$ ./erotima5 < data5.txt
```

Listing 11: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 6

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα το οποίο να δέχεται επαναληπτικά είσοδο μέχρι να εισαχθεί η λέξη bye. Στα δεδομένα που εισάγει ο χρήστης να εντοπίζει τις ακέραιες τιμές, να τις προσμετρά και να εμφανίζει το άθροισμα από όλες τις τιμές που έχουν εισαχθεί. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα.

Λύση

```
%{
int sum=0;
int c=0;
}%

%%
0|[1-9][0-9]* {
    int x = atoi(yytext);
    sum = sum + x;
    c++;
    printf("number=%d, count=%d, sum=%d\n", x, c, sum);
    return 1;
}
. { } // do nothing
bye {return 0;}
%%

int main()
{
    int token;
    while ((token = yylex()) != 0) {
        printf("%s was processed", yytext);
    }
    return 0;
}
```

Listing 12: erotima6.lex

```
$ flex -o erotima6.c erotima6.lex
$ gcc -o erotima6 erotima6.c -lfl
```

Listing 13: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 7

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα το οποίο να αναγνωρίζει στην είσοδο του 32 bit δεκαεξαδικούς αριθμούς, δηλαδή να συμφωνούν με το πρότυπο `0[xX]([0-9a-fA-F]){1,8}`. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα.

Λύση

```
digit      [0-9]
alpha      [a-zA-F]
hextail    ({digit}|{alpha}){1,8}
hex        0[xX]{hextail}

%%
{hex}      printf("Found a HEX number %s !", yytext);
.          printf("");
%%

int main()
{
    printf("Give me your input:\n");
    yylex();
}
```

Listing 14: erotima7.lex

```
$ flex -o erotima7.c erotima7.lex
$ gcc -o erotima7 erotima7.c -lfl
```

Listing 15: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 8

Γράψτε κωδικοποίηση flex που να δημιουργεί ένα πρόγραμμα το οποίο να αντικαθιστά στην είσοδό του τις λέξεις TODAY, TOMORROW, YESTERDAY, TODAY+n, TODAY-n ως εξής:

1. TODAY: αντικατάσταση με τη σημερινή ημερομηνία.
2. YESTERDAY: αντικατάσταση με τη χθεσινή ημερομηνία.
3. TOMORROW: αντικατάσταση με τη αυριανή ημερομηνία.
4. TODAY+n: αντικατάσταση με τη σημερινή ημερομηνία μετά από n ημέρες.
5. TODAY-n: αντικατάσταση με τη σημερινή ημερομηνία πριν από n ημέρες.

Μπορείτε να χρησιμοποιήσετε τον ακόλουθο κώδικα:

```
#include <time.h>

void printDay(int offset){
    char date[32];
    time_t t = time(NULL) + offset * 86400;
```

```

    strftime(date, 32, "%d %B", localtime(&t));
    printf("%s", date);
}

```

Listing 16: εμφάνιση ως λεκτικό της τρέχουσας ημερομηνίας συν/πλην έναν αριθμό ημερών

Λύση

```

POSITIVE_INTEGER    [1-9][0-9]*
TODAY_PLUS_N        TODAY\+{POSITIVE_INTEGER}
TODAY_MINUS_N       TODAY\-{POSITIVE_INTEGER}

%{
#include <time.h>

void printDay(int offset){
    char date[32];
    time_t t = time(NULL) + offset * 86400;
    strftime(date, 32, "%d %B", localtime(&t));
    printf("%s", date);
}

}%

%%
TODAY printDay(0);
YESTERDAY printDay(-1);
TOMORROW printDay(1);
{TODAY_PLUS_N} {
    char* c = strchr(yytext, '+');
    printDay(atoi(c));
}
{TODAY_MINUS_N} {
    char* c = strchr(yytext, '-');
    printDay(atoi(c));
}

%%
int main()
{
    yylex();
}

```

Listing 17: erotima8.lex

```

$ flex -o erotima8.c erotima8.lex
$ gcc -o erotima8 erotima8.c -lfl

```

Listing 18: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 9

Γράψτε κωδικοποίηση flex που για είσοδο που περιέχει πραγματικούς αριθμούς να εμφανίζει το ακέραιο και το δεκαδικό τμήμα ως ακεραίους εκτός και εάν η είσοδος ξεκινά με το κείμενο "expect-floats" οπότε θα εμφανίζει την τιμή ως πραγματικό αριθμό. Δώστε τις εντολές που δημιουργούν τον εκτελέσιμο κώδικα.

Λύση

```
%{
#include <math.h>
%}

%s expect

%%

expect-floats BEGIN(expect);

<expect>[0-9]+\.[0-9]+ {
    printf("found a float = %f\n", atof(yytext));
}
<expect>\n {
    BEGIN(INITIAL);
}
[0-9]+ {
    printf("found an integer = %d\n", atoi(yytext));
}
"." {
    printf("found a dot\n");
}

%%

int main(){
    yylex();
}
```

Listing 19: erotima9.lex

```
$ flex -o erotima9.c erotima9.lex
$ gcc -o erotima9 erotima9.c -lfl
```

Listing 20: Εντολές δημιουργίας εκτελέσιμου κώδικα

Ερώτημα 10

A) Το ερώτημα αυτό προϋποθέτει τη χρήση του λογισμικού JFlex. Συμπληρώστε το αρχείο erotima10.jflex (όπου υπάρχουν 3 συνεχόμενα ερωτηματικά) έτσι ώστε για κάθε ακέραιο αριθμό που εντοπίζει να τον αντικαθιστά με τον αντίστοιχο δυαδικό αριθμό. Επιπλέον, κάθε αλφαβητικό σύμβολο να το αντικαθιστά με το αντίστοιχο κεφαλαίο. Δώστε τις εντολές που θα δημιουργήσουν το scanner καθώς και την εντολή που θα εκτελέσει τον κώδικα για ένα υποθετικό αρχείο κειμένου με όνομα test.txt. Για παράδειγμα αν το περιεχόμενο του αρχείου data10.txt είναι:

This is a test using numbers 128, 64, 32, 16, 8, 4, 2 and 1.

τότε θα πρέπει να εξαχθεί το αποτέλεσμα:

THIS IS A TEST USING NUMBERS 10000000, 1000000, 100000, 10000, 1000, 100, 10 and 1.

```
%%
%class Dec2BinScanner
%standalone
    number = ???(1)
    letter = ???(2)
%%
{number}    {
                ???(3)
            }
{letter}    {
                ???(4)
            }
```

Listing 21: erotima10.jflex

B) Γράψτε ισοδύναμη κωδικοποίηση και για την περίπτωση που πρόκειται να χρησιμοποιηθεί το λογισμικό flex.

Λύση Α υποερωτήματος

```
%%
%class Dec2BinScanner
%standalone
    number = [0-9]+
    letter = [a-zA-Z]
%%
{number}    {
                int x = Integer.parseInt(yytext());
                System.out.printf("%s", Integer.toBinaryString(x));
            }
{letter}    {
                System.out.printf("%s", yytext().toUpperCase());
            }
```

Listing 22: erotima10.jflex

```
$ jflex erotima10.jflex
$ javac Dec2BinScanner.java
$ java Dec2BinScanner data10.txt
```

Listing 23: Εντολές για τη μεταγλώττιση και την εκτέλεση του κώδικα

Λύση Β υποερωτήματος

```
%{
#include <ctype.h>
%}

number [0-9]+
letter [a-zA-Z]
```

```
%%
{number}      {
    int c = atoi(yytext);
    int buffer[100]; // maximum 100 digits
    int i=0;
    while (c>0){
        buffer[i] = c%2;
        c /= 2;
        i++;
    }
    i--;
    while(i>=0){
        printf("%d", buffer[i]);
        i--;
    }
}
{letter}      {
    printf("%c", toupper(*yytext));
}
%%

int main()
{
    yylex();
}
```

Listing 24: erotimal0.jflex

```
$ flex -o erotimal0.c erotimal0.lex
$ gcc -o erotimal0 erotimal0.c -lfl
$ ./erotimal0 < data10.txt
```

Listing 25: Εντολές για τη μεταγλώττιση και την εκτέλεση του κώδικα

Αναφορές

Οι παραπάνω ασκήσεις γράφτηκαν συλλέγοντας πληροφορίες από διάφορες πηγές. Μερικές από αυτές είναι οι ακόλουθες:

- <https://www.cs.york.ac.uk/fp/lisa>
- <http://dinosaur.compilertools.net/flex/>

Ο κώδικας των λύσεων βρίσκεται στο https://github.com/chgogos/uoi_compilers/tree/master/exercises_set2.