

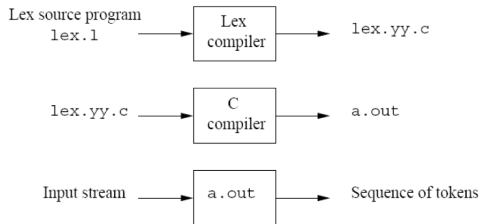
Laboratorio di Compilatori: 11 Aprile 2013

Michele Peroli



REsearch Group in Information Security
Department of Computer Science
University of Verona, Italy

April 17, 2013



flex name.l

gcc lex.yy.c -o name -lfl

./name [< test.txt]

- Modifica il programma Lex di Fig. 3.23 in modo da aggiungere la parola chiave `while` e permettere l'uso del simbolo “`_`” (underscore) come lettera aggiuntiva.
- Scrivere un programma Lex che copia un file sostituendo ogni sequenza non vuota di spazi bianchi con un singolo carattere vuoto (blank).
- Scrivere un programma Lex che copia un programma C sostituendo ogni istanza della parola chiave `float` con `double`.
- Scrivere un programma Lex che stampa tutti i tag HTML presenti in un file (per default Lex legge dallo standard input).
- Scrivere un programma Lex che converte un file trasformando ogni parola nel seguente modo:
 - se la prima lettera è una consonante, allora questa viene spostata alla fine e viene aggiunto `'ay'`;
 - se la prima lettera è una vocale, allora si riscrive la stessa parola aggiungendo alla fine `'ay'`.

Modifica il programma Lex di Fig. 3.23 in modo da aggiungere la parola chiave `while` e permettere l'uso del simbolo “`_`” (underscore) come lettera aggiuntiva.

Da fare a mano lo riprenderemo in futuro.

Scrivere un programma Lex che copia un file sostituendo ogni sequenza non vuota di spazi bianchi con un singolo carattere vuoto (blank).

```
%{  
%}  
%%  
[ ]+ {printf(" ");}  
%%  
int main(int argc, char **argv)  
{  
    yylex();  
    printf("\n");  
}
```

Scrivere un programma Lex che copia un programma C sostituendo ogni istanza della parola chiave float con double.

```
%{  
#include <stdio.h>  
#define FLOAT "float"  
%}  
%%  
float {printf("double");}  
%%  
int main(int argc, char **argv)  
{  
yylex();  
}
```

Scrivere un programma Lex che stampa tutti i tag HTML presenti in un file (per default Lex legge dallo standard input).

```
%{  
#include <stdio.h>  
%}  
%%  
"<"[^>]*> { printf("VALUE: %s\n", yytext); }  
.|\\n      ;  
%%  
main()  
{  
    yylex();  
}
```

Scrivere un programma Lex che converte un file trasformando ogni parola nel seguente modo:

- se la prima lettera è una consonante, allora questa viene spostata alla fine e viene aggiunto “ay”;
- se la prima lettera è una vocale, allora si riscrive la stessa parola aggiungendo alla fine “ay”.

TBA :-)

Questo esempio stampa a schermo il nome utente

```
%option main
%%
[ ];
username { printf("%s", getlogin() );}
.    { printf("Riconoscimento fallito.\n");}

%%
```

Per ciascuno dei seguenti problemi, scrivere un programma Lex che ne realizza una soluzione.

- Sostituire tutte le vocali sia minuscole che maiuscole che compaiono in un testo di input con occorrenze di vocali diverse da quelle lette.
- Contare il numero delle righe e dei caratteri in un testo.
- Invertire le lettere che compongono una data stringa in input.
- Stampare a schermo la stringa letta.

- Scrivere un riconoscitore per rispondere alle parole inserite.
- Realizzare uno scanner per un linguaggio simil-Pascal che contiene
 - identificatori, i.e. sequenze di lettere minuscole e cifre che cominciano con una lettera;
 - numeri interi e float;
 - parole chiave: if, then, begin, end, procedure, function;
 - operatori: +, -, *, /