

# *ESERCIZI CON FLEX*

Giovedì 17 Ottobre

## PRIMI ESERCIZI

- ◉ Scrivere un programma in Flex che :
  1. Converta tutte le lettere maiuscole di un file in minuscole, ad eccezione di quelle racchiuse tra " e "
  2. Restituisca il più piccolo dei numeri naturali presenti nel testo
  3. Conti le occorrenze dei numeri pari e di quelli dispari

## ESERCIZIO

- Scrivere un programma in Flex che converta un file attraverso un cifrario a sostituzione. Esso consiste nel triplicare ogni vocale con l'aggiunta di una g e una s tra le occorrenze: per esempio, a diventa agasa, e diventa egese, e così via (quindi "ciao" diventa cigisiagasaogoso). Assumiamo che il file sia una sequenza di parole separati da spazi.



## ESERCIZIO

- Scrivere un programma in Flex che converta un file attraverso il Cifrario di Cesare.  
Il cifrario di Cesare è uno dei più antichi algoritmi crittografici di cui si abbia traccia storica. È un cifrario a sostituzione monoalfabetica in cui ogni lettera del testo in chiaro è sostituita nel testo cifrato dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto. Nel nostro caso useremo uno spostamento di 3 posizioni. Ovvero 'A' sarà sostituito con 'D', 'B' con 'E' e così via. La cifratura è ciclica, nel senso che 'X' è sostituito con 'A', 'Y' con 'B' e 'Z' con 'C'.



# SOLUZIONE

```
%{  
    int offset = 0;  
}%  
  
%option noyywrap  
%%  
[a-z] {fprintf(yyout,"%c",(*yytext - 97 + offset) % 26 + 97);}   
[A-Z] {fprintf(yyout,"%c",(*yytext - 65 + offset) % 26 + 65);}   
%%  
int main(int argc, char* argv[])  
{  
    if(argc > 1)  
    {  
        offset = atoi(argv[1]);  
        if (argc>2) yyin = fopen(argv[2],"r");  
        if (argc>3) yyout = fopen(argv[3],"w");  
    }  
    else  
    {  
        offset = 3;  
    }  
  
    yylex();  
    return 0;  
}
```

## ESERCIZIO

- (da Wikipedia) Un rudimentale sistema di cifratura basato sul cifrario di Cesare è stato usato anche da Bernardo Provenzano per proteggere informazioni rilevanti scritte nei suoi famosi pizzini, i piccoli foglietti di carta con i quali il boss della mafia, durante la sua latitanza, riceveva informazioni e impartiva ordini. Il sistema scelto da Provenzano era abbastanza semplice: si trattava di sostituire ad ogni lettera il numero corrispondente alla posizione nell'alfabeto sommato a 3 e di comporre così un singolo, lungo numero. Ad esempio, i numeri "512151522 191212154" nascondono il nome di "Binnu Riina": infatti,  $5 = 2$  (posizione della B) più 3;  $12 = 9$  (posizione della I) più 3; ecc...

Sebbene non molto edificante, scrivere un programma in Flex che simuli tale codifica.

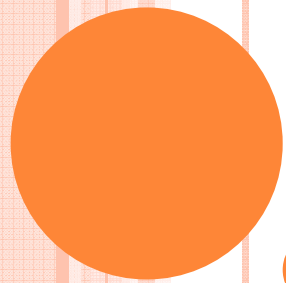
## ESERCIZIO

- Scrivere un programma in Flex che, preso in input un file di testo, conti tutte le parole sull'alfabeto {a, b, c} in cui la prima occorrenza del simbolo 'b' deve essere sempre preceduta da almeno un'occorrenza di 'a';
- Scrivere un programma in FLEX che aggiunga alla fine di ogni parola una 'a' per ogni occorrenza di 'he' nella parola e nell'ordine una 'b' per ogni occorrenza di 'the' e una 'c' per ogni occorrenza di 'she'.

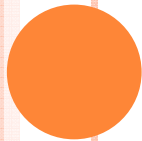
Per esempio:

shethenths -> shethenthsaabc





# START CONDITIONS





## START CONDITIONS

- Flex fornisce un meccanismo di regole che si attivano in modo condizionato.
- Le condizioni START sono dichiarate nella I sezione con %s o %x (condizioni inclusive o esclusive)
- Una condizione START è attivata usando l'azione BEGIN (lo scanner si trova nella cond START);
- Se la cond. è inclusiva le regole che non si trovano in nessuna delle condizioni START saranno attive.
- Se la cond. è esclusiva le sole regole ad essere attive saranno quelle qualificate con la cond. START attivata.



## CONDIZIONI INCLUSIVE

**Problema: Copiare l'input cambiando la parole "magic" in "first" in tutte le linee che cominciano per a, in "second" in quelle che cominciano per b e in "third" in quelle che cominciano per c**

```
%s AA BB CC
%%
^a {ECHO; BEGIN AA;}
^b {ECHO; BEGIN BB;}
^c {ECHO; BEGIN CC;}
\n {ECHO; BEGIN 0;}
<AA>magic {printf("first");}
<BB>magic {printf("second"); }
<CC>magic {printf("third");}
. ;
```



## CONDIZIONI ESCLUSIVE

**Problema: Scanner che riconosce e scarta i commenti mantenendo un contatore delle linee.**

%x comment

%%

int linea=1;

"/\*\*"

BEGIN(comment);

<comment>[^\*\n] ; /\* rimuove tutto ciò che non è '\*' \*/

<comment>\\*[^\*/\n]\* ;

<comment>\n ++linea;

<comment> "\*\*/" BEGIN(INITIAL);

. printf("sono fuori dal commento\n");

***BEGIN(0) o BEGIN(INITIAL) sono equivalenti e riportano lo scanner allo stato originale in cui nessuna condizione start è attiva.***



## ESERCIZI

- ◉ Scrivere un programma in FLEX che trasformi in maiuscole tutte le parole delle righe che cominciano con una parola di caratteri tutti minuscoli, in minuscole tutte le parole delle righe che cominciano con una parola di caratteri tutti maiuscoli. Le altre siano lasciate inalterate.

