ESERCIZI CON BISON

Lunedì 3 Dicembre

Esercizi

1. Scrivere un programma in Bison che valuti un'espressione in forma postfissa.

2. Scrivere un programma in Bison che converta in notazione postfissa, le espressioni in forma infissa corretta.

3. Scrivere un programma in Bison che converta in notazione infissa, le espressioni in forma postfissa corretta.

4. Scrivere un programma che testi se una parola è palindroma

Attributi multitype

- In molti programmi è necessario avere differenti tipi di per gli attributi dei token e dei simboli non terminali .
- Per tale motivo si deve specificare l'intera collezione di tipi con la dichiarazione %union e poi scegliere di volta in volta il tipo piu' utile per i token (con %token) e con i simboli non terminali (con %type).
- %union {

  int intero;

  float reale;

  }

Funziona come l'union in C, ovvero le due definizioni condividono la stessa area di memoria.

- Per indicare in Flex un attributo di tipo intero si userà yylval.intero, altrimenti yylval.reale

File in Bison

```
%{
#include <stdio.h>
void yyerror(char*);

%}

%union {
  int intero;
  float reale;
}

%left '+'
%left '*'

%token <intero> NUM_INT
%token <reale> NUM_REAL
%type <reale> exp

%%
input:      /* empty */
            | input line ;

line:       '\n'
            | exp '\n'
{printf("Valore=%f\n",$1);};
exp : exp '+' exp {$$=$1+$3;}
    | exp '*' exp {$$=$1*$3;}
  | NUM_INT {$$=(float)$1;}
  | NUM_REAL {$$=$1;}
  ;
%%

void yyerror(char* s) {
  fprintf(stderr,"%s\n",s);
}

int main(void) {
  yyparse();
  return 0;
}
```

File in Flex

```
%{
#include "eseunion.tab.h"
%}

%option noyywrap

%%
[ \t]+ ;
[0-9]+          {sscanf(yytext,"%d",&yylval.intero); return NUM_INT;
            /*oppure yylval.intero=atoi(yytext);*/
[0-9]+"."[0-9]* {sscanf(yytext,"%f",&yylval.reale); return NUM_REAL;}
.|\n            return yytext[0];
%%
```