Questão Aula Tipo - Flex+Bison

Uma nova rede social de *microblogging*, estilo *Twitter*, pretende receber mensagens em bloco, usando um ficheiro de texto. Para isso, começou por definir os formatos disponíveis para envio de mensagens. Cada mensagem é uma única linha de texto, com a seguinte estrutura em formato EBNF (em que os campos opcionais são apresentados entre [] e os que se repetem 0 ou mais vezes entre { }):

<remetente> [<destinatário>{,<destinatário>}] <messagem>
[<hashtag>{,<hashtag>}] <sigilo>

Em que o formato de cada campo é o seguinte:

- **<remetente> e <destinatário>** – String de caracteres alfanuméricos, iniciada por @, com um comprimento máximo de 15 carateres;
- **<mensagem>** – String de caracteres alfanuméricos limitada por aspas (pode conter espaços);
- **<hashtag>** – String com minúsculas ou algarismos que inicia com # e tem um comprimento máximo de 10 carateres ;
- **<sigilo>** – Inteiro entre 1 e **10** ;

As mensagens sem destinatário(s) são públicas e não têm nível de sigilo. As mensagens com destinatário(s) são privadas e têm obrigatoriamente um nível de sigilo.

Defina a gramática para o ficheiro anteriormente descrito, e crie utilizando o Flex e o Bison um programa que:

- Reconheça a validade do ficheiro;
- Produza resultados com o seguinte formato:

<remetente>: <pública ou privada> <sigilo se privada> N:<total_hashtag>\n

- No final, apresente o nível de sigilo da mensagem privada com maior número de destinatários

Exemplo:

Ficheiro de entrada:

@eumesmo "Os D.A.M.A. vêm à queima!" #queima2016,#festa

@eumesmo "O Anselmo vêm à queima!" #queima2016,#festa,@anselmoralph

@outroTipo1 @eumesmo "Não te esqueças do Quim!" #quimbarreiros 3

@eumesmo @outroTipo1,@outroTipo2,@outroTipo3 "Eu não vou ver o Quim!" 5

@vendedor "Quem quer bilhetes para a queima?"

@outroTipo2 @eumesmo "Tens que vir ver o Quim!" #festabrava,#quimbarreiros 1

@comprador @vendedor "Qual é o preço?" #queima2016 10

Resultado produzido

@eumesmo: publica N:2

@eumesmo: publica N:3

@outroTipo1: privada 3 N:1

@eumesmo: privada 5 N:0

@vendedor: publica N:0

@outroTipo2: privada 1 N:2

@comprador: privada 10 N:1

A mensagem com mais destinatários tem sigilo 5.

Bison

```
%{
  #include <stdio.h>
  #include <string.h>

int maxDestinatarios=-1, maxSigilo = -1;
void processaLinha( char *remetente, int ndestinatarios, int nhashtags, int sigilo );

%}

%union {
   char *string;
   int inteiro;
}

%token <inteiro> SIGILO
%token <string> ID MENSAGEM HASH

%type <inteiro> listaDestinatarios listaHash listaHashOpcional

%%
```

Bison

```
inicio: /* vazio */
    | inicio linha '\n'
    ;

linha: ID MENSAGEM listaHashOpcional { processaLinha( $1, 0, $3, 0 ); }
      | ID listaDestinatarios MENSAGEM listaHashOpcional SIGILO
                                                { processaLinha( $1, $2, $4, $5 ); }
    ;

listaDestinatarios: ID { $$ = 1; }
    | listaDestinatarios ',' ID { $$ = $1 + 1; }
    ;

listaHashOpcional: /* vazio */ { $$ = 0; }
    | listaHash { $$ = $1; }
    ;

listaHash: HASH { $$ = 1; }
    | listaHash ',' HASH { $$ = $1 + 1; }
    ;

%%
```

```
<remetente> [<destinatário>{,<destinatário>}] <messagem>
            [<hashtag>{,<hashtag>}] <sigilo>
```

Bison

```
int main() {
  yyparse();
  printf("A mensagem com mais destinatários tem sigilo %d.\n",maxSigilo);
  return 0;
}

void processaLinha( char *remetente, int ndestinatarios, int nhashtags, int sigilo )
{
  printf("%s: ",remetente);
  if (ndestinatarios==0) printf("publica "); else printf("privada %d ",sigilo);
  printf( "N: %d\n", nhashtags);
  if (ndestinatarios > maxDestinatarios) {
   maxDestinatarios = ndestinatarios;
   maxSigilo = sigilo;
  }
}

int yyerror(char *s)
{
  printf("ERRO SEMANTICO: %s\n",s);
}
```

Flex

```
%{
  #include <string.h>
  #include "qa2016b.tab.h"
%}

ALFA [A-Za-z0-9]
MIN [a-z0-9]
SIGILO ([0-9]|10)

%%

@{ALFA}{1,15}   yylval.string=strdup(yytext); return ID;
\".+\"          return MENSAGEM;
#{ALFA}{1,15}   return HASH;
{SIGILO}        yylval.inteiro=atoi(yytext); return SIGILO;
[,\n]           return yytext[0];
.

%%
```