



## POSSÍVEL SOLUÇÃO DA PROVA... (disponível na Internet no mesmo site indicado acima)

- Analise este programa e procure verificar qual a similaridade com a sua solução, buscando entender como funciona (discuta com seus colegas)

### ANALISADOR LÉXICO (standalone – funciona isoladamente)

```
%{
/* Author Fernando Osorio */
/* Created: April 2006 */

#include "y.tab.h"
// extern int yylval;
// extern int flag_user1;
// extern int flag_user2;
// extern int flag_text;

#define TESTE 1
int flag_user1;
int flag_user2;
int flag_text;

}%

%%
#DATA { return INIDATA; }
#User1 {
    flag_user1=1;
    return INIUSR1;
}
#User2 {
    flag_user2=1;
    return INIUSR2;
}
#ENDATA { return FIMDATA; }
#TEXT {
    flag_text=1;
    return INITEXT;
}
#ENDTEXT { return FIMTEXT; }
$chairemail { return CEMAIL; }
$chairname { return CNAME; }
$conference { return CONFNAME; }
$confsite { return CONFSITE; }
$email { return UEMAIL; }
$title { return UTITLE; }
$givenname { return UGNAME; }
$surname { return USNAME; }
$loginname { return ULOGIN; }
$password { return UPASSWD; }
$number { return PNUMBER; }
$papertitle { return PTITLE; }
\[^\\" data-bbox="134 178 609 917"/>
```

```

%%

#ifdef TESTE

#include <stdio.h>

main(argc,argv)
int argc;
char *argv[];
{
    int val;

    flag_user1=0;
    flag_user2=0;
    flag_text=0;

    while (val = yylex())
        printf("Token: %d - [%s]\n",val,yytext);
}

#endif

```

---

### Saída:

Token: 258 - [#DATA]	Token: 275 - [\$papertitle]
Token: 264 - [\$chairmail]	Token: 280 - [=]
Token: 280 - [=]	Token: 277 - ["I no say write inglishi"]
Token: 277 - ["knuth@stanford.edu"]	Token: 274 - [\$number]
Token: 265 - [\$chairname]	Token: 280 - [=]
Token: 280 - [=]	Token: 277 - ["002"]
Token: 277 - ["Donald Knuth"]	Token: 269 - [\$title]
Token: 266 - [\$conference]	Token: 280 - [=]
Token: 280 - [=]	Token: 277 - ["Tequinico em TI"]
Token: 277 - ["IEEE World CompSci Education"]	Token: 268 - [\$email]
Token: 267 - [\$confsite]	Token: 280 - [=]
Token: 280 - [=]	Token: 277 - ["sergio@hell.com.br"]
Token: 277 - ["www.acm.org"]	Token: 270 - [\$givenname]
Token: 259 - [#User1]	Token: 280 - [=]
Token: 268 - [\$email]	Token: 277 - ["Sergio"]
Token: 280 - [=]	Token: 271 - [\$surname]
Token: 277 - ["ana@acm.org"]	Token: 280 - [=]
Token: 274 - [\$number]	Token: 277 - ["Liso"]
Token: 280 - [=]	Token: 272 - [\$loginname]
Token: 277 - ["001"]	Token: 280 - [=]
Token: 269 - [\$title]	Token: 277 - ["a002"]
Token: 280 - [=]	Token: 273 - [\$password]
Token: 277 - ["Prof. Dra."]	Token: 280 - [=]
Token: 270 - [\$givenname]	Token: 277 - ["qwerty"]
Token: 280 - [=]	Token: 261 - [#ENDATA]
Token: 277 - ["Ana"]	Token: 262 - [#TEXT]
Token: 271 - [\$surname]	Token: 259 - [#User1]
Token: 280 - [=]	Token: 278 - [To]
Token: 277 - ["Ferro"]	Token: 281 - [:]
Token: 275 - [\$papertitle]	Token: 268 - [\$email]
Token: 280 - [=]	Token: 278 - [From]
Token: 277 - ["How to be a good teacher"]	Token: 281 - [:]
Token: 272 - [\$loginname]	Token: 264 - [\$chairmail]
Token: 280 - [=]	Token: 278 - [Subject]
Token: 277 - ["a001"]	Token: 281 - [:]
Token: 273 - [\$password]	Token: 266 - [\$conference]
Token: 280 - [=]	Token: 281 - [-]
Token: 277 - ["xpto"]	Token: 278 - [Paper]
Token: 260 - [#User2]	Token: 274 - [\$number]

Token: 278 - [Decision]  
Token: 278 - [Notification]  
Token: 278 - [Dear]  
Token: 269 - [\$title]  
Token: 270 - [\$givenname]  
Token: 271 - [\$surname]  
Token: 281 - [:]  
Token: 278 - [Congratulations]  
Token: 281 - [!]  
Token: 278 - [Your]  
Token: 278 - [paper]  
Token: 275 - [\$papertitle]  
Token: 278 - [for]  
Token: 266 - [\$conference]  
Token: 278 - [has]  
Token: 278 - [been]  
Token: 278 - [accepted]  
Token: 278 - [The]  
Token: 278 - [reviews]  
Token: 278 - [are]  
Token: 278 - [can]  
Token: 278 - [be]  
Token: 278 - [found]  
Token: 278 - [at]  
Token: 267 - [\$confsite]  
Token: 281 - [.]  
Token: 278 - [Your]  
Token: 278 - [login]  
Token: 278 - [and]  
Token: 278 - [password]  
Token: 278 - [are]  
Token: 281 - [:]  
Token: 272 - [\$loginname]  
Token: 281 - [-]  
Token: 273 - [\$password]  
Token: 278 - [Best]  
Token: 278 - [Regards]  
Token: 281 - [.]  
Token: 265 - [\$chairname]  
Token: 278 - [Conference]  
Token: 278 - [Chair]  
Token: 260 - [#User2]  
Token: 278 - [To]  
Token: 281 - [:]  
Token: 268 - [\$email]  
Token: 278 - [From]  
Token: 281 - [:]  
Token: 264 - [\$chairemail]  
Token: 278 - [Subject]  
Token: 281 - [:]

Token: 266 - [\$conference]  
Token: 281 - [-]  
Token: 278 - [Paper]  
Token: 274 - [\$number]  
Token: 278 - [Decision]  
Token: 278 - [Notification]  
Token: 278 - [Dear]  
Token: 269 - [\$title]  
Token: 270 - [\$givenname]  
Token: 271 - [\$surname]  
Token: 281 - [:]  
Token: 278 - [Try]  
Token: 278 - [again]  
Token: 278 - [next]  
Token: 278 - [year]  
Token: 281 - [!]  
Token: 278 - [Your]  
Token: 278 - [paper]  
Token: 275 - [\$papertitle]  
Token: 278 - [for]  
Token: 266 - [\$conference]  
Token: 278 - [has]  
Token: 278 - [been]  
Token: 278 - [rejected]  
Token: 278 - [The]  
Token: 278 - [reviews]  
Token: 278 - [are]  
Token: 278 - [can]  
Token: 278 - [be]  
Token: 278 - [found]  
Token: 278 - [at]  
Token: 267 - [\$confsite]  
Token: 281 - [.]  
Token: 278 - [Your]  
Token: 278 - [login]  
Token: 278 - [and]  
Token: 278 - [password]  
Token: 278 - [are]  
Token: 281 - [:]  
Token: 272 - [\$loginname]  
Token: 281 - [-]  
Token: 273 - [\$password]  
Token: 278 - [Best]  
Token: 278 - [Regards]  
Token: 281 - [.]  
Token: 265 - [\$chairname]  
Token: 278 - [Conference]  
Token: 278 - [Chair]  
Token: 263 - [#ENDTEXT]

**ANALISADOR GRAMATICAL [Só a gramática – não inclui a saída final]  
(versão completa na Internet – ga-v3.l / ga-v3.y)**

```
/* PROVA.Y */
/* Author Fernando Osorio */
/* Created: April 2006 */

#include <stdio.h>
#include <string.h>

typedef struct {
    char nome[255];
    char email[255];
    char titulo[255];
    char site[255];
} conf_t;

typedef struct {
    char gnome[255];
    char snome[255];
    char tit[255];
    char email[255];
    char login[255];
    char pwd[255];
    char pn[255];
    char pt[255];
} user_t;

int flag_user=0;
int flag_text=0;
char texto[255];

conf_t conferencia;
user_t usuario[2];

%}

%token INIDATA
%token INIUSR1
%token INIUSR2
%token FIMDATA
%token INITEXT
%token FIMTEXT
%token CEMAIL
%token CNAME
%token CONFNAME
%token CONFSITE
%token UEMAIL
%token UTITLE
%token UGNAME
%token USNAME
%token ULOGIN
%token UPASSWD
%token PNUMBER
%token PTITLE
%token TAG
%token STRTXT
%token PALAVRA
%token NUMBER
%token EQUAL
%token THINGS

%start template

%%
```

```

template : sectdata secttext      { printf("Template\n"); }
;

sectdata :
    INIDATA subdata FIMDATA      { printf("Data Section:
%s\n",conferencia.titulo); }
;

subdata: seqdata subdata        { }
    | seqdata                    { }
;

seqdata : CEMAIL EQUAL STRTXT { strcpy(conferencia.email,texto); }
    | CNAME EQUAL STRTXT { strcpy(conferencia.nome,texto); }
    | CONFNAME EQUAL STRTXT { strcpy(conferencia.titulo,texto); }
    | CONFSITE EQUAL STRTXT { strcpy(conferencia.site,texto); }
    | UEMAIL EQUAL STRTXT { strcpy(usuario[flag_user].email,texto); }
    | UTITLE EQUAL STRTXT { strcpy(usuario[flag_user].tit,texto); }
    | UGNAME EQUAL STRTXT { strcpy(usuario[flag_user].gnome,texto); }
    | USNAME EQUAL STRTXT { strcpy(usuario[flag_user].snome,texto); }
    | ULOGIN EQUAL STRTXT { strcpy(usuario[flag_user].login,texto); }
    | UPASSWD EQUAL STRTXT { strcpy(usuario[flag_user].pwd,texto); }
    | PTITLE EQUAL STRTXT { strcpy(usuario[flag_user].pt,texto); }
    | PNUMBER EQUAL STRTXT { strcpy(usuario[flag_user].pn,texto); }
    | INIUSR1
    | INIUSR2
;

secttext : INITEXT texto_livre FIMTEXT { printf("Texto Section\n"); }
;

texto_livre : CEMAIL texto_livre { }
    | CNAME texto_livre { }
    | CONFNAME texto_livre { }
    | CONFSITE texto_livre { }
    | UEMAIL texto_livre { }
    | UTITLE texto_livre { }
    | UGNAME texto_livre { }
    | USNAME texto_livre { }
    | ULOGIN texto_livre { }
    | UPASSWD texto_livre { }
    | PTITLE texto_livre { }
    | PNUMBER texto_livre { }
    | PALAVRA texto_livre { }
    | THINGS texto_livre { }
    | INIUSR1 texto_livre { }
    | INIUSR2 texto_livre { }
    | PALAVRA
;

%%

#include <stdio.h>
extern FILE *yyin;

main ()
{
    flag_user=0;
    flag_text=0;

    do { yyparse(); }
    while (!feof(yyin));
}

```