

1

P40

MIGUEL ÂNGELO RIBEIRO SOUSA SILVA  
Nº 30226

05-05-2009

List → '[' ']'  
| '[' Conteúdo ']'

Conteúdo → NUM | ElemList

ElemList → ε

| '[' Conteúdo

Yacc

listas.y

```
% {  
%} #include "TADList.h"
```

```
% token ABRE FECHA VIRG NUM
```

```
% start List  
% union {
```

```
int nlex;  
List l;  
}  
% type <nlex> NUM
```

```
% type <l> List Conteúdo
```

CONTINUA →

TADList.h

```
# ifndef LISTAS  
# define LISTAS
```

```
typedef struct slist
```

```
{  
int num;  
struct slist * seg;  
} * List, * Nodolist;
```

List → '[' ']'  
<sub><list></sub> | '[' Conteúdo

Conteúdo → Conteúdo ',' NUM  
<sub><list></sub> | NUM <nlex>

Gramática Abstrata

List → ~~Conteúdo~~

List → List NUM

| NUM

```
%%  
S : List { showlist ($1); printf ("%d",  
sumlist ($1)); } ;  
%%  
List : ABRE FECHA { $$ = NULL; }  
| ABRE CONTEUDO FECHA { $$ = $2; }  
;  
Conteúdo : Conteúdo VIRG NUM  
| NUM { $$ = conslist ($1, NULL); }  
;  
Conteúdo : Conteúdo VIRG NUM { $$ = conslist ($3, $1); }  
| NUM { $$ = conslist ($1, NULL); }  
;
```

listas.l ← listas.y  
TADlistas.h → y.tad.h  
TADlistas.c