Atividade Prática - Analisador Sintático

1)

<comando> ::= PRINT <texto>

| ADD <numero> <numero>

| SUB <numero> <numero>

<texto> ::= STRING

<numero> ::= NUMBER

2) a)

<comando>

|

PRINT

|

<texto>

|

STRING

|

"Hello, World!"

b)

<comando>

|

SUB

/ \

<numero> <numero>

| |

NUMBER NUMBER

| |

15 4

c)

<comando>

|

ADD

/ \

<numero> <numero>

| |

NUMBER NUMBER

| |

5 10

3) comandos.l:

%{

#include "comandos.tab.h"

%}

%%

"PRINT" { return PRINT; }

"ADD" { return ADD; }

"SUB" { return SUB; }

"\".\*\"" { yylval.string\_val = strdup(yytext+1); yylval.string\_val[strlen(yylval.string\_val)-1] = '\0'; return STRING; }

[0-9]+ { yylval.int\_val = atoi(yytext); return NUMBER; }

[ \t\n] ; // Ignore whitespace

. { return yytext[0]; }

%%

int yywrap() {

return 1;

}

comandos.y:

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

%}

%token PRINT ADD SUB STRING NUMBER

%%

comando : PRINT texto { printf("Comando PRINT com texto: %s\n", $2); }

| ADD numero numero { printf("Comando ADD com números: %d, %d\n", $2, $3); }

| SUB numero numero { printf("Comando SUB com números: %d, %d\n", $2, $3); }

;

texto : STRING { $$ = $1; }

;

numero : NUMBER { $$ = $1; }

;

%%

int main() {

yyparse();

return 0;

}

Compilação e Execução:

flex comandos.l

bison -d comandos.y

gcc lex.yy.c comandos.tab.c -o analisador -lfl -ly

./analisador