

5COP093 - Lista de Exercícios 08

1. Considere a gramática a seguir:

```
 $S \rightarrow \text{if } E \text{ then } S \text{ else } S$   
 $S \rightarrow \text{begin } S \ L$   
 $S \rightarrow \text{print } E$   
 $L \rightarrow \text{end}$   
 $L \rightarrow ; \ S \ L$   
 $E \rightarrow \text{num} = \text{num}$ 
```

Abaixo existe um pseudo-código que implementa um analisador sintático descendente recursivo para a gramática acima:

```
int IF=1, THEN=2, ELSE=3, BEGIN=4, END=5, PRINT=6, SEMI=7, NUM=8, EQ=9;
```

```
int token = getToken();  
void advance() {token=getToken();}  
void eat(int t) {if (token==t) advance(); else error();}  
  
void S(){  
    switch(token) {  
        case IF: eat(IF); E(); eat(THEN); S(); eat(ELSE); S(); break;  
        case BEGIN: eat(BEGIN); S(); L(); break;  
        case PRINT: eat(PRINT); E(); break;  
        default: error(); }  
}  
void L(){  
    switch(token) {  
        case END: eat(END); break;  
        case SEMI: eat(SEMI); S(); L(); break;  
        default: error(); }  
}  
void E(){ eat(NUM); eat(EQ); eat(NUM); }
```

Implemente o código acima em C, de modo a torná-lo funcional. Observe que será necessário implementar um analisador léxico para os símbolos terminais da gramática. Se a cadeia de entrada for aceita, imprima a mensagem:

CADEIA ACEITA.

Se a cadeia for recusada, imprima a mensagem:

CADEIA RECUSADA.