

Um algoritmo para simulação de um AP pode ser implementado da maneira como segue, onde a variável Pilha corresponde a pilha do autômato e terá o valor VAZIO quando esta estiver vazia, SimbEntrada corresponde ao simbolo da fita de entrada para o qual a cabeça de entrada está apontando no momento, SimbTopo corresponde ao simbolo que esta no topo da pilha (terminal ou não terminal). Aceita é um flag que será usado para verificar se uma determinada string é aceita ou não, isto é, se seu valor final for SIM, ok, NAO caso contrário.

```

01.#define NAO      0
02.#define SIM      1
03.#define VAZIO    '\0'
04.#define FIMFITA '\0'
05.
06.Pilha = VAZIO;
07.Empilha(S, Pilha);
08.SimbEntrada = Proximo(FitaEntrada);
09.Aceita = SIM;
10.while (Aceita && (SimbEntrada != FIMFITA ||
11.    Pilha != VAZIO))
12.{
13.  SimbTopo = Desempilha(Pilha);
14.
15.  if(SimbTopo in T) { // é terminal ?
16.    if(SimbTopo != SimbEntrada){
17.      Aceita = NAO;
18.      break;          }
19.    else {
20.      reconhece(SimbEntrada);
21.
22.      SimbEntrada = Proximo(FitaEntrada);
23.    }
24.  }
25.  else
26.    if(SimbTopo in N) { // é não terminal
27.      selecione uma produção A -> X1...Xn;
28.      //Substitui o não terminal pelo lado
29.      // direito da produção adequada
30.      Empilha(X1...Xn, Pilha);
31.    }
32.    else Aceita = NAO;
33. }
34. retorne Aceita;

```

Faça um teste de mesa para a seguinte gramática, utilizando a string **acbcc**.

```

S-> ABC
A->aC
B->bC
C->c

```