```
1: procedure NextWord()
2:     state ← s_0
3:     lexeme ← ''
4:     clear(stack)
5:     push(⟨bad, bad⟩)
6:     while state ≠ s_e do
7:         NextChar(char)
8:         InputPos ← InputPos + 1
9:         lexeme ← lexeme + char
10:        if Failed[state, InputPos] then
11:            break
12:        end if
13:        if state ∈ S_A then
14:            clear(stack)
15:        end if
16:        push(⟨state, InputPos⟩)
17:        cat ← CharCat[char]
18:        state ← δ[state, cat]
19:    end while
20:    while state ∉ S_A and state ≠ bad do
21:        Failed[state, InputPos] ← true
22:        ⟨state, InputPos⟩ ← pop()
23:        truncate(lexeme)
24:        RollBack()
25:    end while
26:    if state ∈ S_A then
27:        return TokenType[state]
28:    else
29:        return bad
30:    end if
31: end procedure
32: procedure InitializeScanner()
33:    InputPos ← 0
34:    for each state s in the DFA do
35:        for i = 0 to |input stream| do
36:            Failed[s, i] ← false
37:        end for
38:    end for
39: end procedure
```