```
get first Input_Character
if Input_Character in ['A'..'Z'] then
    begin
      while Input_Character in ['A'..'Z', '0'..'9'] do
        begin
            get next Input_Character
            if Input_Character = '_' then
                begin
                    get next Input_Character
                    Last_Char_Is_Underscore := true
                end  {if '_'}
            else
                Last_Char_Is_Underscore := false
        end  {while}
      if Last_Char_Is_Underscore then
        return (Token_Error)
      else
        return (Valid_Token)
    end  {if first in ['A'..'Z']}
else
    return (Token_Error)
```

(a)

State	A-Z	0-9	_	
1	2			{starting state}
2	2	2	3	{final state}
3	2	2		

(b)

Figure 5.10 Token recognition using (a) algorithmic code and (b) tabular representation of finite automaton.