

$S \rightarrow cAt$
 $A \rightarrow a \mid \epsilon$

(a) Construct First sets and Follow sets for the nonterminals in the grammar

$First(S) = \{c\}$
 $First(A) = \{a, \epsilon\}$

 $Follow(S) = \{\$ \}$
 $Follow(A) = \{t\}$

(b) Construct a recursive-decent parser (pseudo code) for the grammar.

```
const int a = 1 , c = 2 , t = 3;
int token = scanner();

void match(int t ){
    if(token == t){
        token = scanner();
    }
    else
        error();
}

void S(){
    switch(token){
        case c : match(c); A(); match(t);
                break;
        default: error();
    }
}

void A(){
    switch(token){
        case a : match(a);
                break;
        case t : break;
        default : error();
    }
}
```

(c) Construct the LL(1) parsing table for the grammar.

	c	a	t	\$
S	$S \rightarrow cAt$			
A		$A \rightarrow a$	$A \rightarrow \epsilon$	

(d) Show the behavior of the parser on the sentence “cat”.

