MUNIB-UL-HASSAN

CS 19-037

"A"

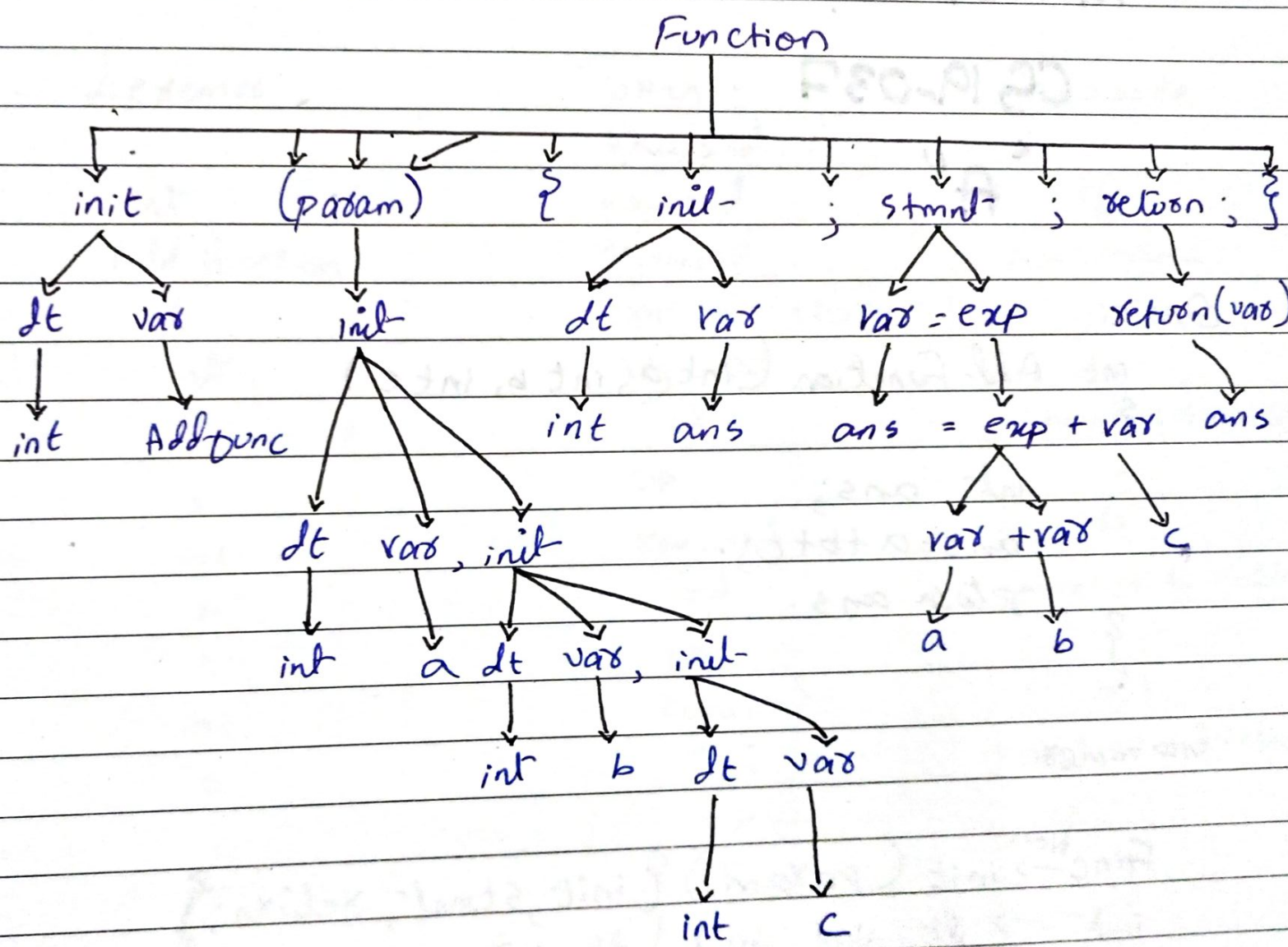
Q no 01.

```
int Add_Function (int a, int b, int c)
{
    int ans;
    ans=a+b+c;
    return ans.
}
```

Grammer:

Function → init (param) {init, stmnt; return;}
init → dt var, init | dt var
param → init
stmnt → var = exp
exp → exp | exp+ var | var + var
return → return var
dt → int | void | float

Parse Tree


Function
init
(param)
{
init-
;
stmnt-
;
return ;
}
dt
var
int
AddFunc
init
dt
var
int
ans
var = exp
return (var)
ans = exp + var
ans
dt
var , init
int
a
dt
var , init-
int
b
dt
var
int
c
var +var
c
a
b

Date: _____

Symbol Table

Lexemes	Token	Attribute
	keyword.	
int	keyword	dt
Add-function	keyword	func-name.
(	operator (op)	-
int	keyword	dt
a	id	pointer to table
,	op	-
int	keyword	dt
b	id	pointer to table
,	op	-
int	keyword	dt
c	id	pointer to table
,	op	-
)	op	-
{	op	-
int	keyword	dt
ans	id	pointer to table
;	op	-
ans	id	pointer to table
=	relop	EQ
a	id	pointer to table
+	Add-op	-
b	id	pointer to table
+	Add-op	-
c	id	pointer to table
;	op	-

return	keyword	—
(	op	—
ans	id	pointer to t
)	op	—
;	op	—
}	op	—