1. result = oldsum - value / 100;

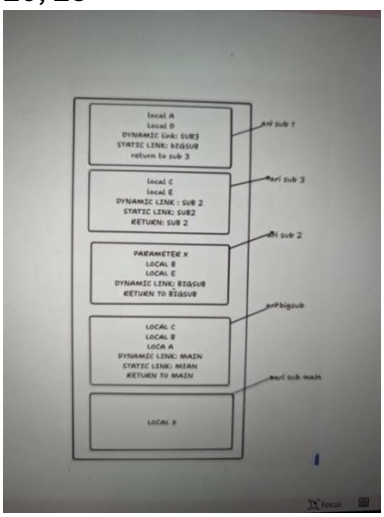
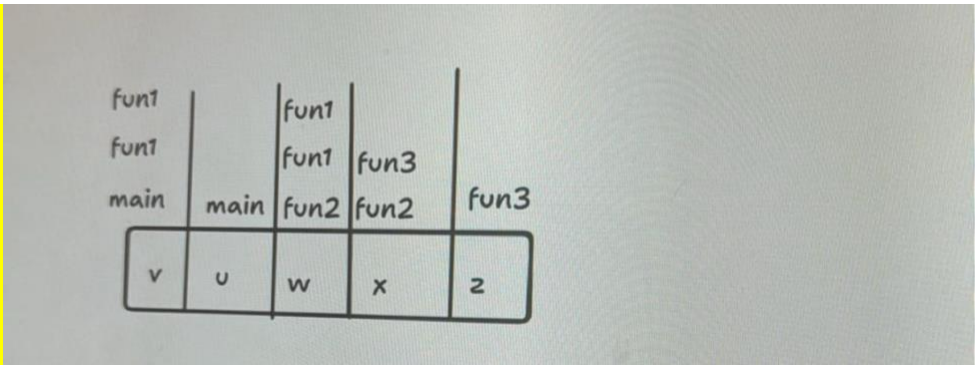
Lexemes and tokens

result IDENT
= ASSIGN_OP
oldsum IDENT
- SUB_OP
value IDENT
/ DIVISION_OP
100 INT_LIT
; SEMICOLON

2. (id * id) * id

Stack	Input Buffer	Action
0	(id*id)*id$	Shift
0(	Id*id)*id$	Shift
0(id	*id)*id$	Reduce id -> F
0(F	*id)*id$	Reduce F -> T
0(T	*id)*id$	Reduce T -> E
0(E	*id)*id$	Shift
0(E*	Id)*id$	Shift
0(E*id	)*id$	Reduce id -> F
0(E*F	)*id$	Reduce F -> T
0(E*T	)*id$	Reduce E * T -> E
0(E	)*id$	Shift
0(E)	*id$	Reduce (E) -> F
0F	*id$	Reduce F -> T
0T	*id$	Shift
0T*	Id$	Shift
0T*id	$	Reduce id -> F
0T*F	$	Reduce T * F -> T
0T	$	Recude T -> E
0E	$	Accept

3. 1. x, y, z from main;
   X, from main; a, y, z from sub1;
   - From main; b from sub1; a, b, z from sub2;
   - - from main; b from sub1; b, z form sub2; a, x, w from sub3
   -

4. 20, 25
5. 
local A
local D
DYNAMIC Link: SUB3
STATIC LINK: BIGSUB
return to sub 3
local C
local E
DYNAMIC LINK : SUB 2
STATIC LINK: SUB2
RETURN: SUB 2
PARAMETER x
LOCAL B
LOCAL E
DYNAMIC LINK: BIGSUB
RETURN TO BIGSUB
LOCAL C
LOCAL B
LOCA A
DYNAMIC LINK: MAIN
STATIC LINK: MIAN
RETURN TO MAIN
LOCAL x
6. 
fun1
fun1
main
main
fun1
fun1
fun2
fun3
fun2
fun3
v
u
w
x
z

partII

1. a
2. d
3. b
4. c
5. c
6. c
7. b
8. b
9. c
10. b