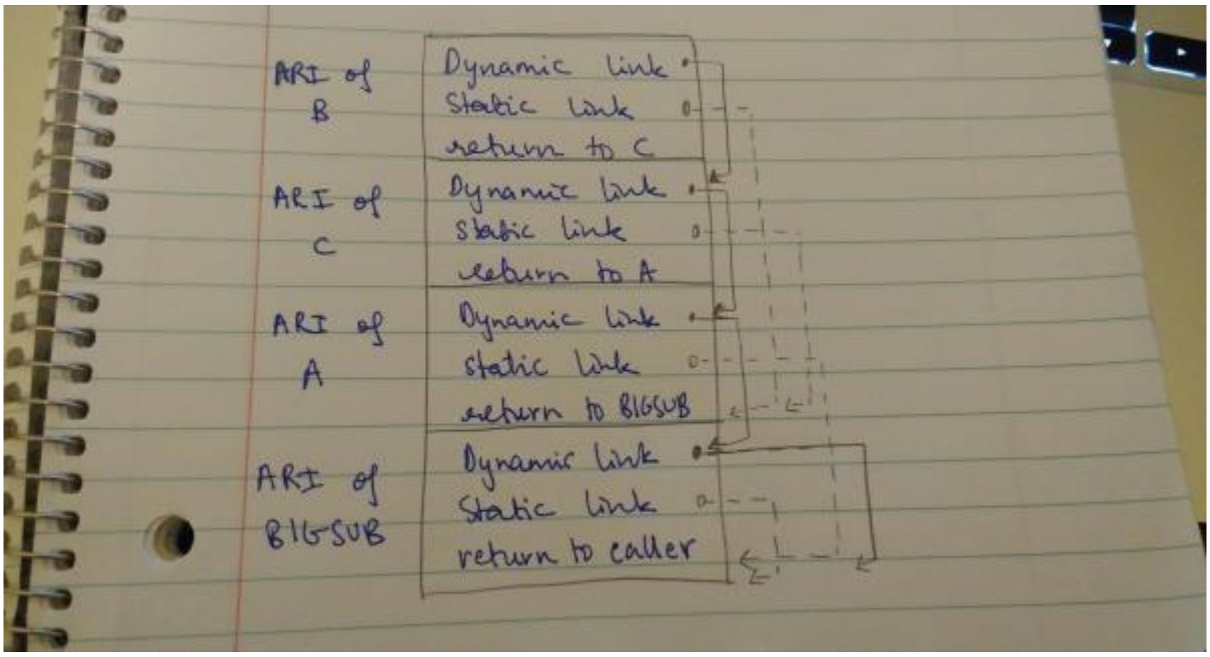
Assignment: Subprograms 2

1. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume Bigsub is at level 1.

```
procedure Bigsub is
procedure A is
procedure B is
begin  -- of B
. . .                          →1
end; -- of B
procedure C is
begin  -- of C
. . .
B;
. . .
end;  -- of C

begin  -- of A
. . .
C;
. . .
end;  -- of A
begin  -- of Bigsub
. . .
A;
. . .
end; -- of Bigsub
```

Answer:

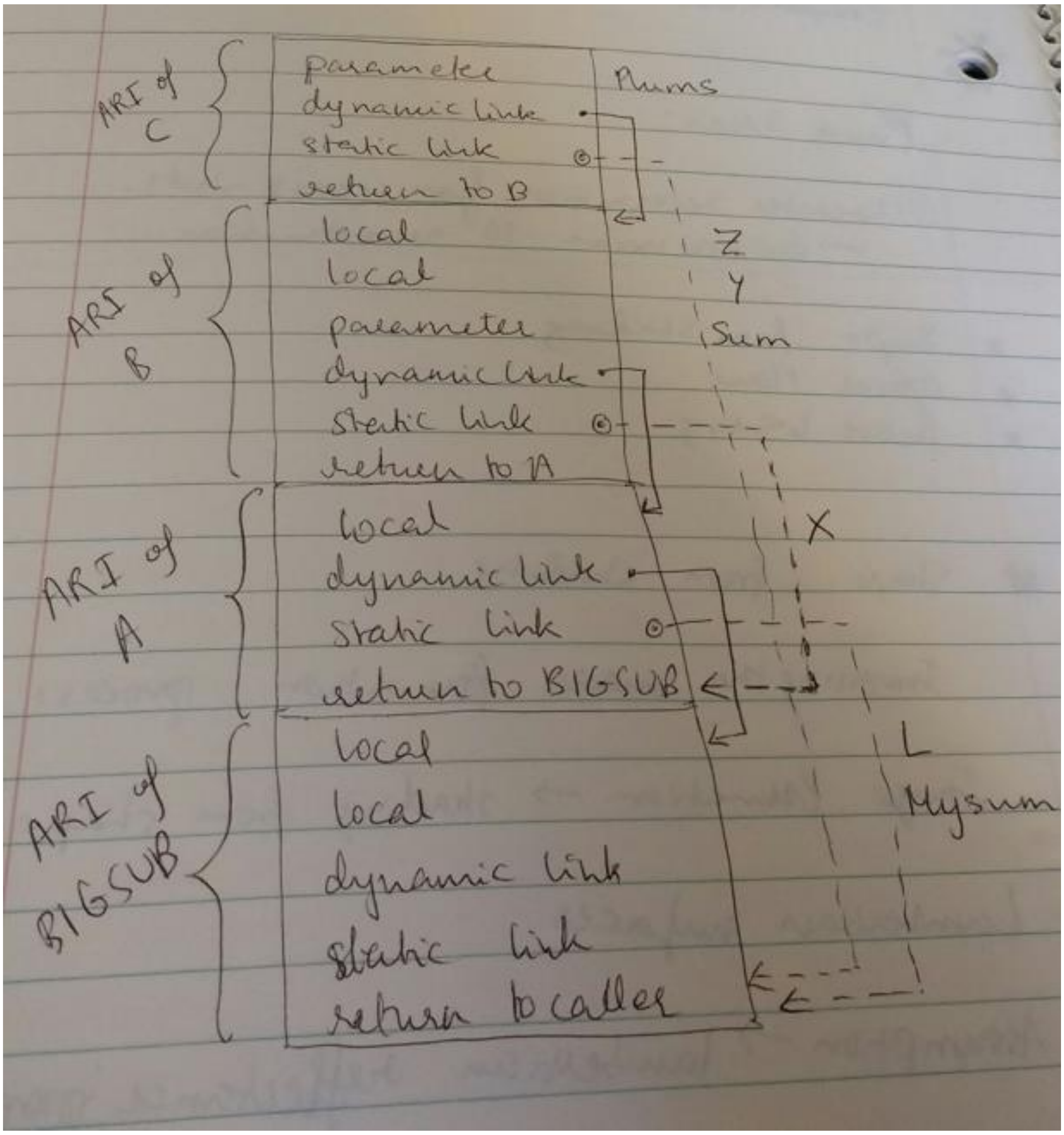


2. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal

program. Assume Bigsub is at level 1.

```
procedure Bigsub is
MySum : Float;
procedure A is
X : Integer;
procedure B(Sum : Float) is
Y, Z : Float;
begin  -- of B
. . .
C(Z)
. . .
end; -- of B
begin  -- of A
. . .
B(X);
. . .
end;  -- of A
procedure C(Plums : Float) is
begin  -- of C
. . .                              →1
end; -- of C
L : Float;
begin  -- of Bigsub
. . .
A;
. . .
end; -- of Bigsub
```

Answer:



3. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume Bigsub is at level 1.

```
procedure Bigsub is
procedure A(Flag : Boolean) is
procedure B is
. . .
A(false);
end; -- of B
begin -- of A
```

```
if flag
then B;
else C;
. . .
end; -- of A
procedure C is
procedure D is
. . .                              →1
end; -- of D
. . .
D;
end; -- of C
begin -- of Bigsub
. . .
A (true );
. . .
end; -- of Bigsub
```

The calling sequence for this program for execution to reach D is

Bigsub calls A

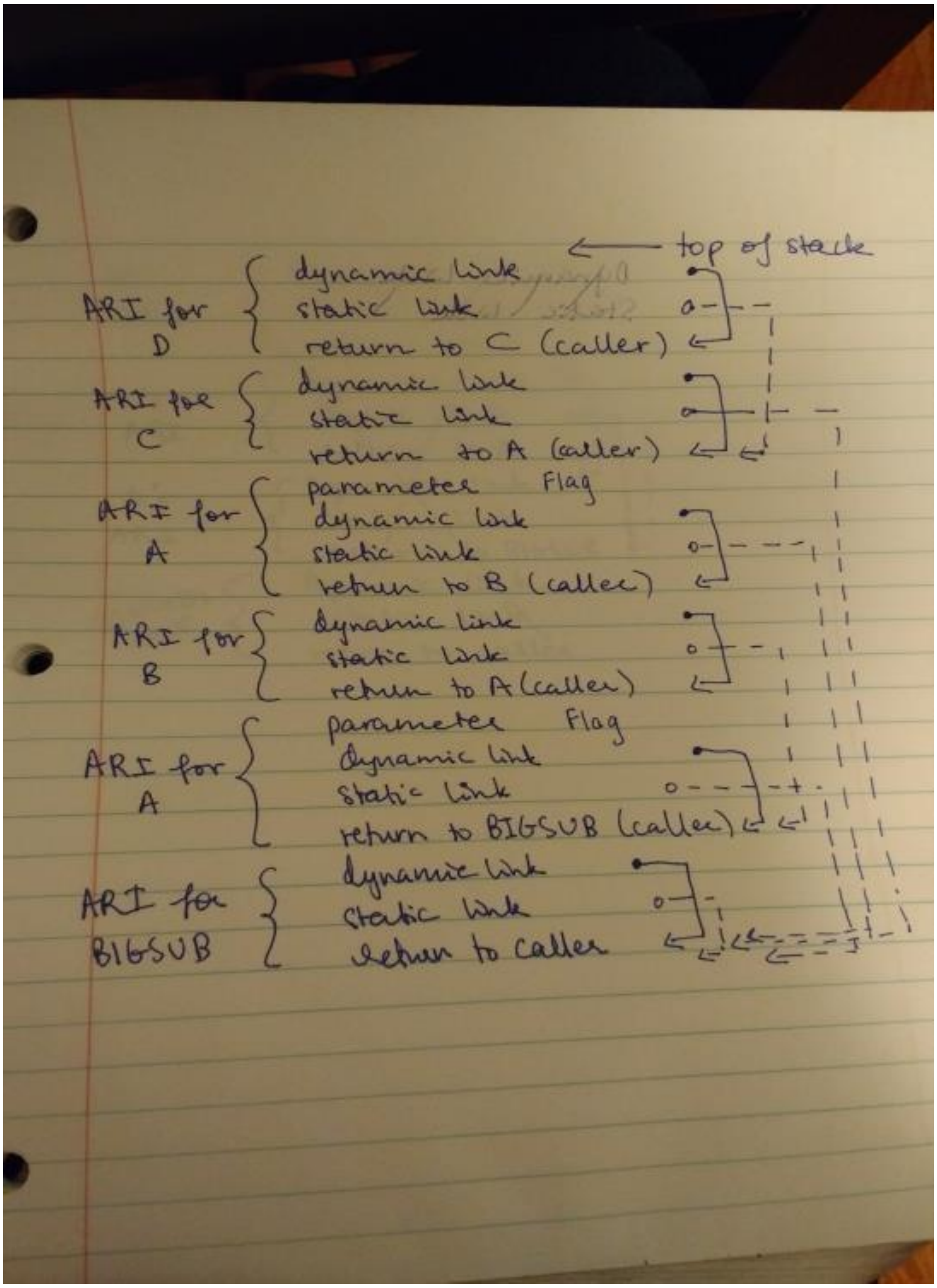
A calls B

B calls A

A calls C

C calls D

Answer:



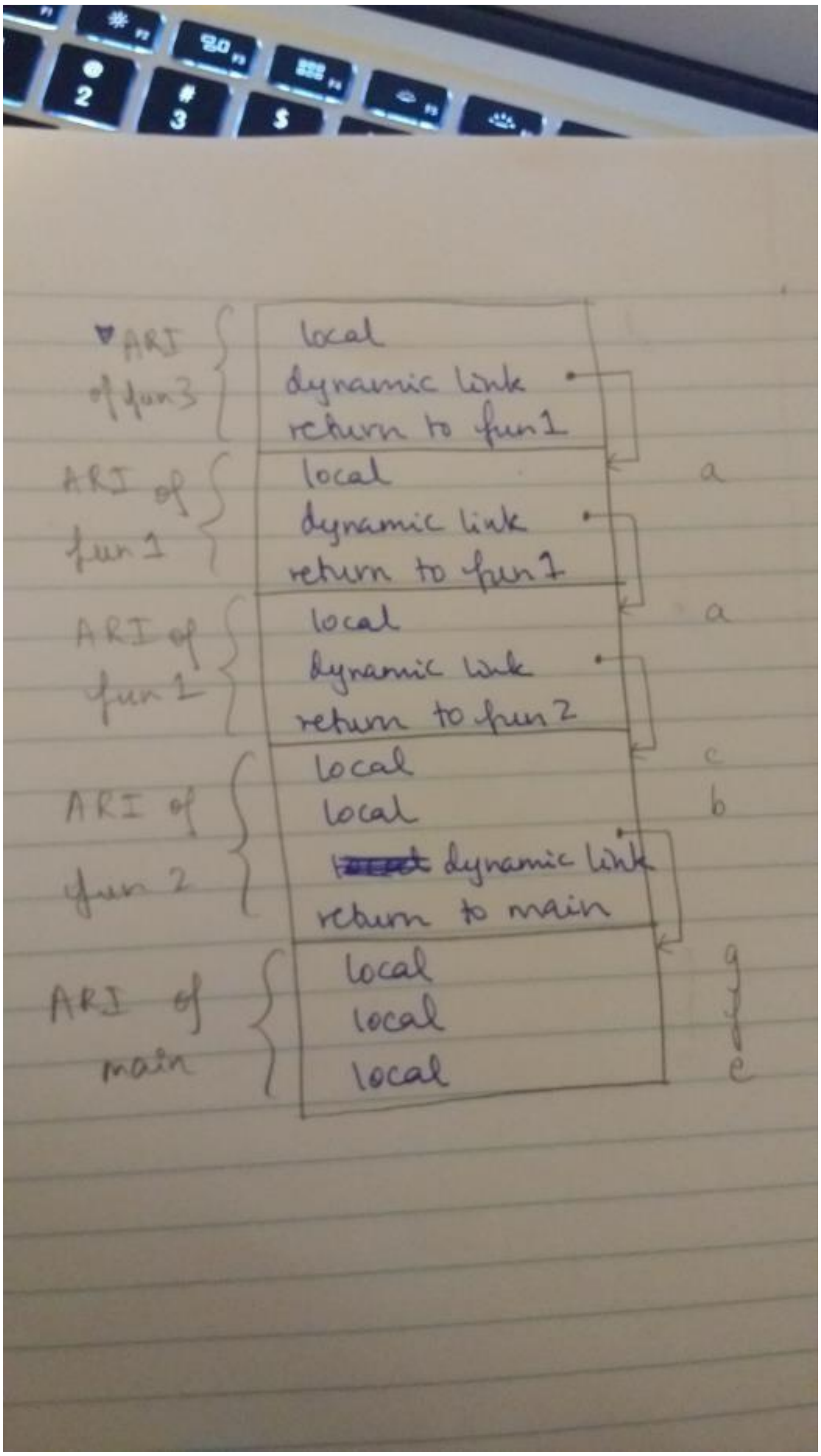
4. Show the stack with all activation record instances, including the dynamic chain, when execution reaches position 1 in the following skeletal program. This program uses the deep-access method to implement dynamic scoping.

```
void fun1() {
float a;
. . .
}
void fun2() {
int b, c;
. . .
}
470 Chapter 10 Implementing Subprograms
void fun3() {
float d;
. . .                   →1
}
void main() {
char e, f, g;
. . .
}
```

The calling sequence for this program for execution to reach fun3 is

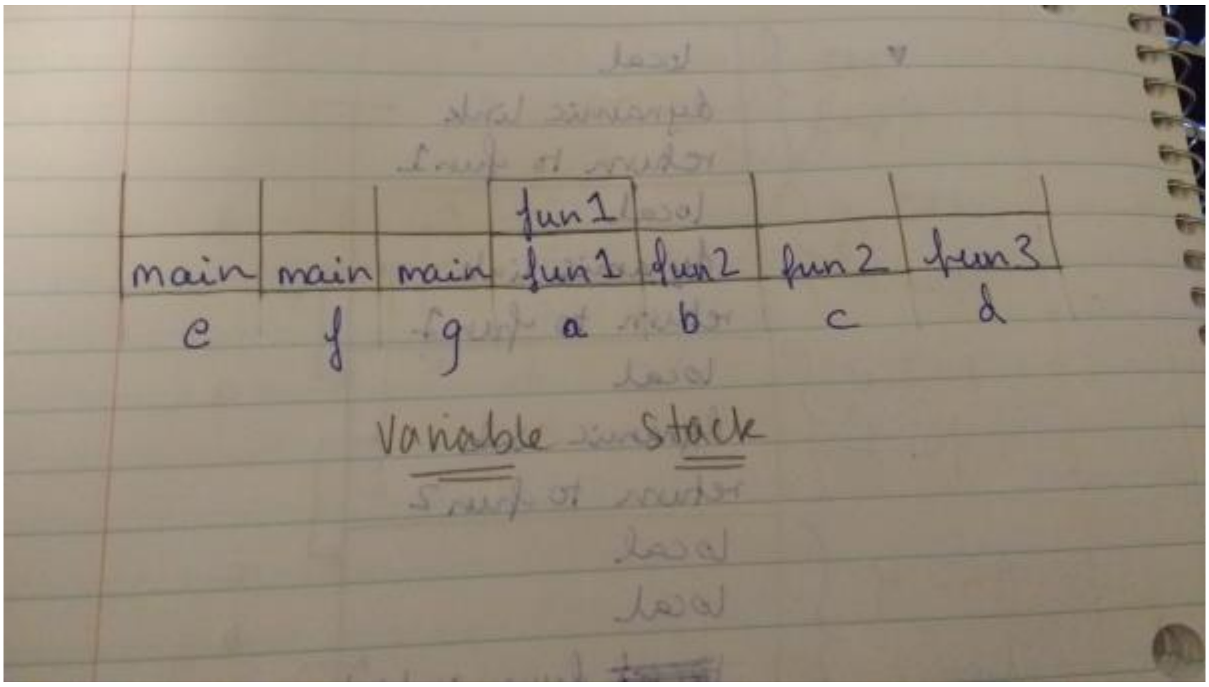
main calls fun2
fun2 calls fun1
fun1 calls fun1
fun1 calls fun3

Answer:


ARI of fun3
local
dynamic link
return to fun1
ARI of fun1
local
dynamic link
return to fun1
a
ARI of fun1
local
dynamic link
return to fun2
a
ARI of fun2
local
c
local
b
dynamic link
return to main
ARI of main
local
g
local
f
local
e

5. Assume that the program of Problem 4 is implemented using the shallow-access method using a stack for each variable name. Show found its way to that point through the sequence of calls shown in Problem 4.

Answer:



			fun1			
main	main	main	fun1	fun2	fun2	fun3
e	f	g	a	b	c	d

Variable Stack