**Chapter 14**

**14.2 Refer to the structure of an activation record for these questions.**

***a.* What is the purpose of the dynamic link?**

***b.* What is the purpose of the return address?**

***c.* What is the purpose of the return value?**

a. Dynamic Link: when control returns to the caller, it will be able once again to access its local variables. If the dynamic link is destroyed, then we will have trouble restarting the caller correctly when the callee finishes.

b. Return Address: when control returns to the caller, it will be able once again to access its address, load it to PC and continue the program has not done in the caller.

c. Return Value: when control returns to the caller, it will be able to transfer the value accessed by the callee and return the result to caller which is the main purpose of caller.

**14.4 For each of the following items, identify whether the caller function or the callee function performs the action.**

***a.* Writing the parameters into the activation record.** the caller function

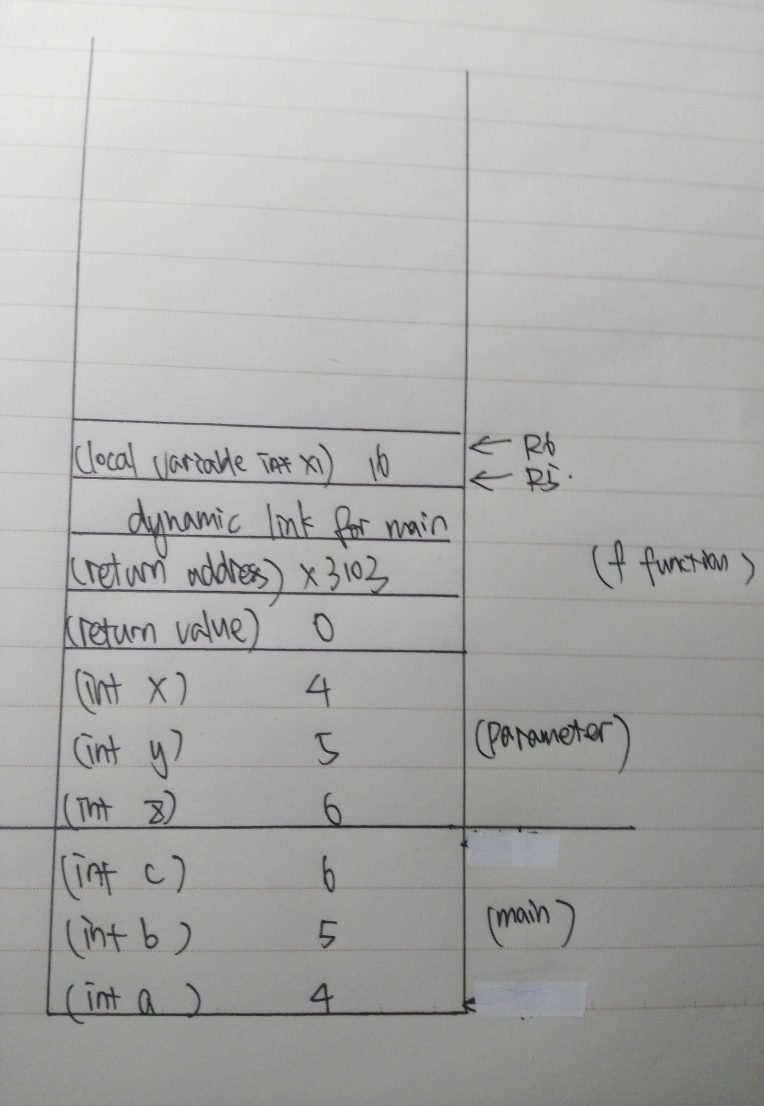
**b. Writing the return value.** the callee function

**c. Writing the dynamic link.** the callee function

***d.* Modifying the value in R5 to point within the callee function's activation record.**

the callee function

**14.15**



**Chapter 17**

**17.2 Is the return address for a recursive function always the same at each**

**function call? Why or why not?**

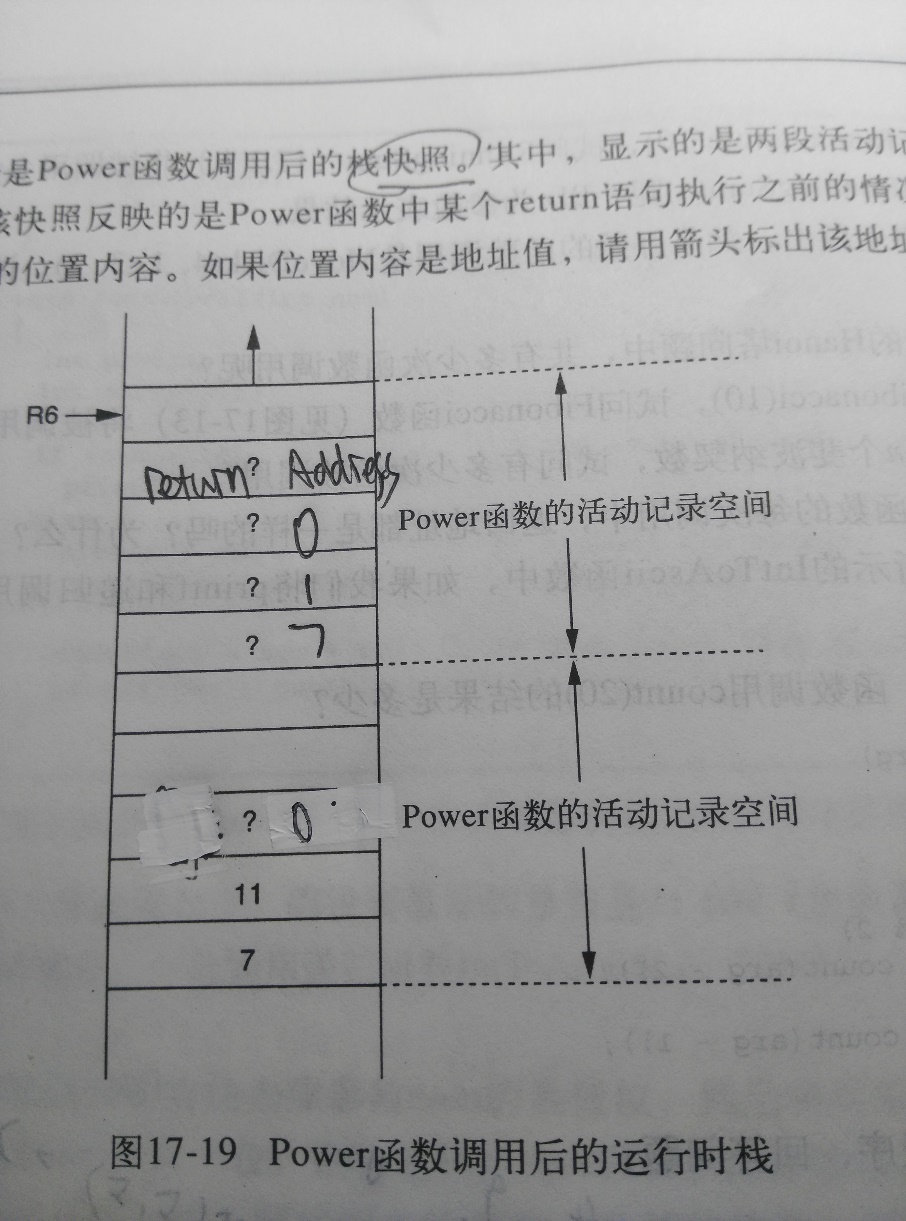
Not always the same at each function call. Because every time a function is called, and activation record is allocated for it in memory. And when the function returns to the caller, its activation record is reclaimed to be assigned later to another function. Each invocation of a function gets its own space in memory for its locals. And just like assignment 4, even though it’s a recursion, there are still different ways to go back, such as go north, east, south, east or end.

**17.5**

a. (1) 0 (2) 2 (3) 0

b. Power(a, b) = Floor(log(b)a)

c.



**17.7** It needs 4\*16 bit to perform the function once.

(1) (16\*8\*1024)/(4\*16)=2048

(2) (4\*8\*1024)/(4\*16)=512