Chapter 6 ACTIVATION RECORD

No. 6.3

```
int f(int a, int b)
{
    int c[3], d, e;
    d = a + 1;
    e = g(c, &b);
    return e + c[1] +b;
}
```

For each a, b, c, d, e should be kept in the memory or register?

variable	**in memory**	**in register**	**reason**
a		√	P132: pass function parameters in registers
b	√		P133: passed by reference, accessed by a procedure
c	√		P133: an array, accessed by a procedure
d		√	P132: intermediate results of expressions, P130: no used after the function g called
e		√	P132: the function result

问题的思考：

需要准确把握变量在内存和寄存器中存储的情形，而且要特别注意是针对 c 语言的，书中这个方面的内容在 REGISTERS 和 FRAME-RESIDENT VARIABLES 两节中分别有介绍。从多个角度来说明原因将更加贴切。

No. 6.7

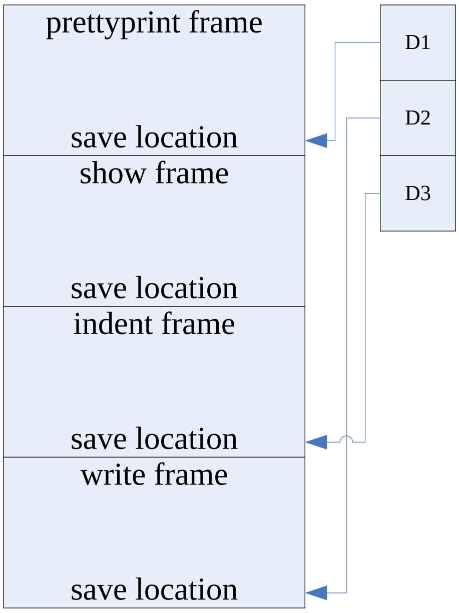
a.

indent uses the variable output from ***prettyprint***'s frame. To do so it starts with its own static link:

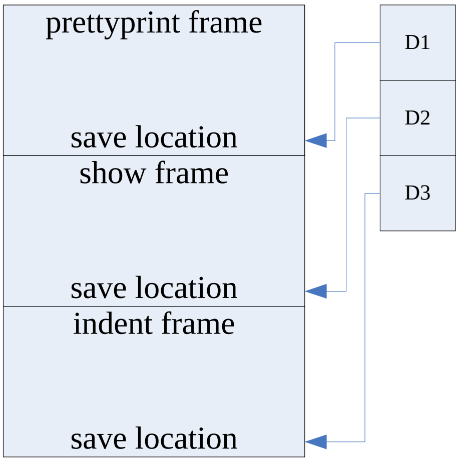
1. get the frame pointer to the ***show***;
2. then fetches ***show***'s static link;
3. get the frame pointer to the ***prettyprint***;
4. then fetches ***output***.

b.

indent is at depth 3, when we fetch the variable ***output*** in Line14, the display and stack view as below:



calling ***write*** in ***indent***



after calling ***write*** in ***indent***, before use the ***output***

When we use the variable ***output***:

1. get the D2 (the frame pointer of ***show***), there is no local variable output;
2. then get D1 (the frame pointer of prettyprint), so fetch the variable output.