1. Context Switching (15 points)

The answer to question 4, hw1 includes the following code:

```c
int sp1, sp2;
void do_yield12(void) {
    do_switch(&sp1, &sp2);
}
```

The `do_switch()` function saves the current stack pointer into the memory location pointed to by the first argument, and loads a new value of the stack pointer from the memory location pointed to by the second argument.

In effect, the `do_switch` function has one input (`sp2`) and one output (`sp1`); rather than being passed or returned by value (i.e. in the normal fashion), the input and output are passed by address.

a) Can the new stack pointer (the input, `sp2`) be passed by value, as shown in the following code? Why or why not? (7.5 points)

```c
do_yield12() {
    do_switch(&sp1, sp2);
}
```

b) Can the old stack pointer be returned by value? Why or why not? (7.5 points)

```c
do_yield12() {
    sp1 = do_switch(&sp2);
}
```

2. Synchronization primitives (30 points)

A rendezvous is a synchronization primitive defined as follows:

Given a rendezvous `R` and an integer value `I`, we define a single operation on `R`:

```c
int result = rendezvous(R, I)
```

with the following definition:

- If a thread `T1` calls `rendezvous(R, I1)` and there is no thread waiting, it waits.
- If a second thread `T2` calls `rendezvous(R, I2)` and there is a thread `T1` waiting, then:
  `rendezvous` returns in `T2` with value `I1`, and in `T1` with value `I2`.

In other words, a rendezvous allows two threads to meet and exchange values. If a thread is first, then it will wait until the second one arrives; if it is second, then it will return immediately.

Provide pseudo-code to implement a rendezvous using either a monitor or semaphores.
3. Address translation (30 points)

To the right are figures representing a logical view and an in-memory view of a 2-level page table mapping 2 virtual pages.

Each gray box in the lower figure represents a single page table entry, of the following format:

[... page number (20 bits) ...] [11 0s] [present bit]

We wish to map the following two virtual pages:

70301000 ... 70301FFF
08002000 ... 08002FFF

For each of the four gray fields in the lower figure, please give:

• the offset (0-1023) within the page table or page directory
• the page number contained in the field.