Exercise 6.12 (a)

- If (g > h) g = g + h;
  else g = g – h;

#assuming ‘g’ in $s0, and ‘h’ in $s1

```
        slt   $t0, $s1, $s0   # check if h < g. If yes, $t0 = 1, else, $t0 = 0
        bne   $t0, $0, Add1   #if $t0 = 1, branch to ‘Add1’
        sub   $s0, $s0, $s1   #else, continue to subtraction
        J     done
Add1:   add   $s0, $s0, $s1
Done:   #some other codes
```

Try 6.12(b)(c)

Exercise 6.14 (a)

- Identify what registers should be used in this 'strcpy' proc.

-input: 'x' is the base address of 1st array, as input, it is contained in $a0-$a3

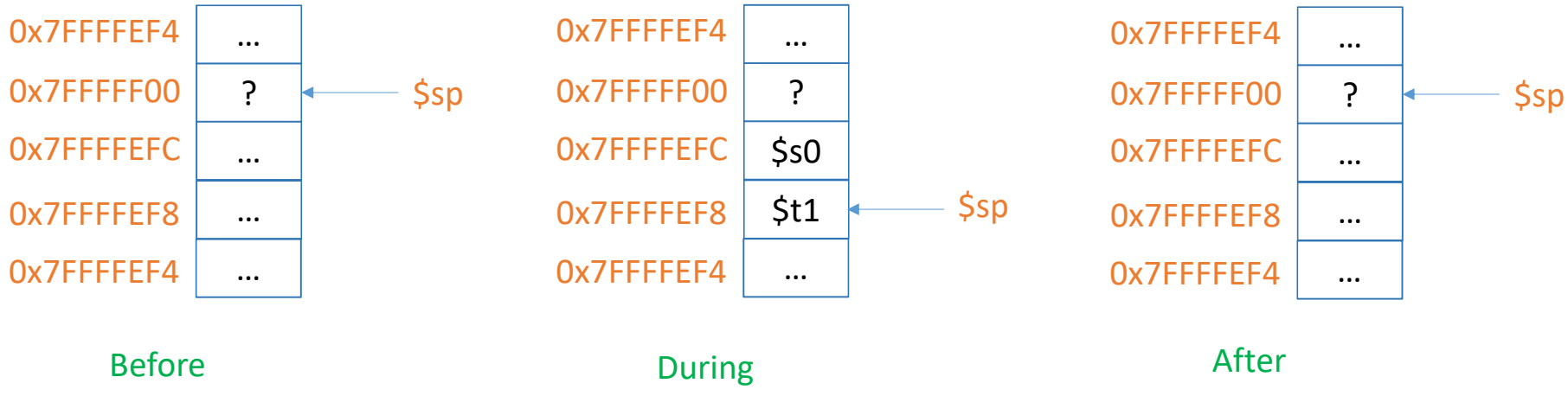
-input: 'y' is the base address of 2nd array, as input, it is contained in $a0-$a3

#So, assume x in $a0 , y in $a1

```
Strcpy: addi $s0, $0          # i = 0. i in $s0
Loop:   sll  $t0, $s0, 2      # $t0 = i * 4 (get the byte offset)
        add  $a0, $a0, $t0    # $a0 contains address of x[i]
        lw   $t1, 0($a0)      # $t1 = x[i]
        beq  $t1, $0, Exit    # If x[i] == 0, while loop breaks
        add  $a1, $a1, $t0    # $a1 contains address of y[i]
        sw   $t1, 0($a1)      # y[i] = $t1
        addi $s0, $s0, 1      # i = i + 1
        j    Loop
Exit:   #some other codes
```

Exercise 6.14 (b)

- We assume that only $s0 and $t1 are stored in the stack



- Write the caller MIPS code that calls the ‘strcpy’ procedure!