TA: Jade Cheng ICS 312 Quiz Solution #4 September 17, 2009

Passing Argument Via the Stack

Question: Calling the subroutine using the stack.

```
Answer:
                       title calc (calc.asm - a separated file)
                       public calc
                       .model small
                       .code
                       proc near
                                                             ; evaluates X - 2Y, the result in ax.
               calc
                                      bp
                                                             ; save bp.
                       push
                                      bp, sp
                       mov
                                                             ; bp points to the stack top.
                       mov
                                      ax, [bp + 4]
                                                             ; ax has X.
                       sub
                                     ax, [bp + 6]
                                                             ; ax = X - Y.
                                                             ; ax = X - 2Y.
                       sub
                                     ax, [bp + 6]
                       pop
                                     bp
                                                             ; restore bp.
                       ret
                                      4
                                                             ; pop IP and add 4 to sp.
               calc
                       endp
                       end
```

Question: Calling the subroutine by reference using the stack.

```
title calc (calc.asm - a separated file)
Answer:
                       public calc
                       .model small
                       .code
               calc
                                                              ; evaluates X - 2Y, the result in ax.
                       proc near
                       push
                                                              ; save bp.
                                       bp
                                       bp, sp
                       mov
                                                              ; bp points to the stack top.
                                       bx, [bp + 4]
                                                              ; bx has the address of X.
                       mov
                       mov
                                      ax, [bx]
                                                              ; as has the value of X.
                                                             ; bx has the address of Y.
                                      bx, [bp + 6]
                       mov
                       sub
                                      ax, [bx]
                                                              ; ax = X - Y.
                       sub
                                       ax, [bx]
                                                              ; ax = X - 2Y.
                       pop
                                       bp
                                                              ; restore bp.
                                       4
                                                              ; pop IP and add 4 to sp.
                       ret
               calc
                       endp
                       end
```