

**Due: Monday, March 5, 2018, by 11:59 pm (upload electronically)**

```
main()
{
    long a[3][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12}};
    long y = val_ele(3,4,a,1,2);

    return y;
}
```

```
val_ele:
    pushl    %ebp
    movl     %esp, %ebp
    movl     20(%ebp), %eax
    sall     $2, %eax
    imull     12(%ebp), %eax
    movl     24(%ebp), %ecx
    leal     0(, %ecx, 4), %edx
    addl     16(%ebp), %edx
    movl     (%edx, %eax), %eax
    popl     %ebp ←STOP HERE
    ret
```

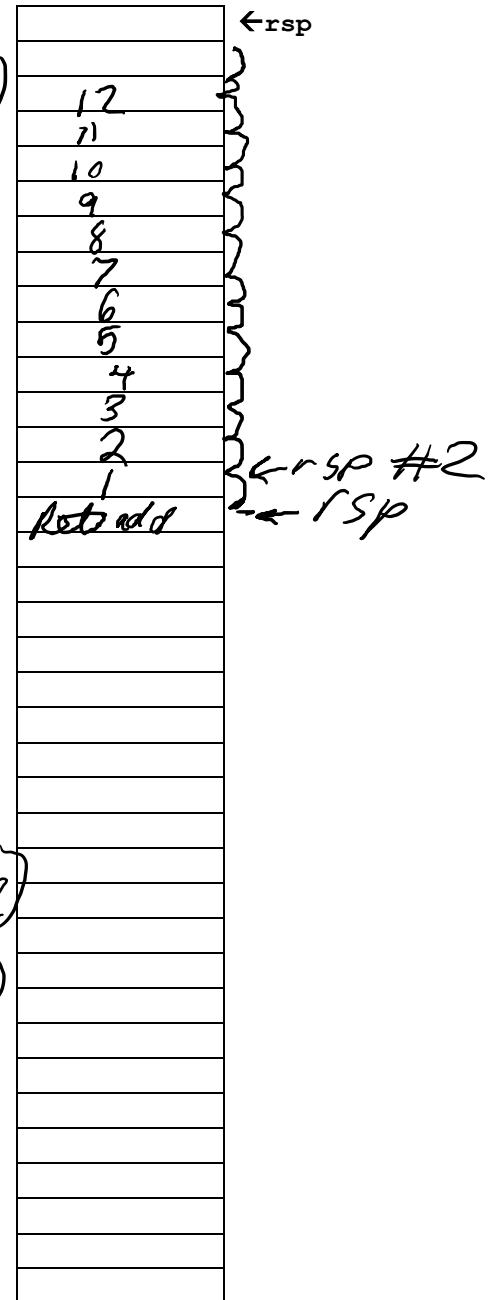
```
main:
    leal    4(%esp), %ecx // first three lines, align esp on 16-byte
    andl    $-16, %esp    // boundary and have it point at the same
    pushl   -4(%ecx)      // return address, as shown on the stack
    pushl   %ebp
    movl    %esp, %ebp    // completed the stack to here .. continue
    pushl   %ecx
    subl    $64, %esp
    movl    $1, -56(%ebp)
    movl    $2, -52(%ebp)
    movl    $3, -48(%ebp)
    movl    $4, -44(%ebp)
    movl    $5, -40(%ebp)
    movl    $6, -36(%ebp)
    movl    $7, -32(%ebp)
    movl    $8, -28(%ebp)
    movl    $9, -24(%ebp)
    movl    $10, -20(%ebp)
    movl    $11, -16(%ebp)
    movl    $12, -12(%ebp)
    pushl   $2
    pushl   $1
    leal    -56(%ebp), %eax
    pushl   %eax
    pushl   $4
    pushl   $3
    call    val_ele
    movl    $0, %eax
    movl    -4(%ebp), %ecx
    leave
    leal    -4(%ecx), %esp
    ret
```

[illegible]

Complete the stack trace showing how the array is stored on the stack in main and how the array is referenced in the call to val\_ele. Finally, what value is returned from the call to val\_ele? \_\_\_\_\_

A reference to 7 on the stack

```
val_ele:
    movslq    %ecx, %rcx
    salq      $3, %rcx
    movslq    %esi, %rsi
    imulq     %rcx, %rsi
    movslq    %r8d, %r8
    leaq      (%rdx,%r8,8), %rax
    movq      (%rax,%rsi), %rax
    ret       ←STOP HERE
```

$$2 * 8 = (16 + 2 \text{ sp } \#?)$$


3 4 RSP#2 1 2  
(4) (11) (12) (16 + RSP#2)  
(32) (11 < 3) (48 + RSP#2)  
(8)