Liam Leech + Drew Pulliam

1- using 4-bit numbers to save space, multiply 4*3

```
  0100
x 0011
------
  0100
 0100
0000
+0000
------
  1100   = 12
```

2- using 4-bit numbers to save space, divide 7/3

```
      0010   Quotient
    ------
0011 | 0111
     - 0011
     ------
       0001   Remainder
```

1- Convert the following C code to MIPS. Assume the address of base array is associated with $s0, n is associated with $s1, position is associated with $t0, c is associated with $t1, d is associated with $t2, and swap is associated with $t3

```
for (c = 0; c < (n - 1); c++)
{
  position = c;

  for (d = c + 1; d < n; d++)
  {
    if (array[position] > array[d])
      position = d;
  }
  if (position != c)
  {
    swap = array[c];
    array[c] = array[position];
    array[position] = swap;
  }
}
```

```
        move    $t1, $zero
loop1:
        addi    $t4, $s1, -1
        bgt     $t1, $t4, EXIT
        move    $t0, $t1
        addi    $t2, $t1, 1
loop2:
        bge     $t2, $s1, IF
        lw      $t5, $t0($s0)
        lw      $t6, $t2($s0)
        blt     $t5, $t6, incr
        move    $t0, $t2
incr:   addi    $t2, $t2, 1
        j       loop2
IF:     bne     $t0, $t1, SWAP
        addi    $t1, $t1, 1
        j       loop1
SWAP:   lw      $t3, $t1($s0)
        lw      $t7, $t0($s0)
        sw      $t7, $t1($s0)
        sw      $t3, $t0($s0)
        addi    $t1, $t1, 1
        j       loop1
EXIT:
```