hmwk03 listing

```
 1                  /*
 2                  This program converts a 3-digit ASCII code to its decimal equivalent
 3
 4                  file: hmwk03.s
 5
 6                  author:
 7
 8                  date:
 9
10                  Environment:  assemble with GNU assembler (GAS)
11
12                  Labels in Memory:
13
14                      hundreds  -- Represent the first digit in the hundreds position.
15                      tens      -- Represent the second digit in the 10s position
16                      ones      -- Represent the third digit in the ones position.
17                      result    -- Holds the final answer.  It will also hold intermediate results for debugging
18
19                  Register usage:
20
21                      RAX -- Accumulator.  Holds the results of multiplication and added digits
22                      RBX -- Conversion of the ASCII code to decimal value
23                      RDX -- Will not be used, but it will be cleared after every multiplication.
24                      R10 -- Holds 10.  The mul command does not allow immediate literals, so the value 10 must go in
25
26                  */
27                  .globl _start
28                  .data
29
30                     # The three-digit number is 218
31 0000 32000000      hundreds: .quad 50  # ASCII code for 2
31      00000000
32 0008 31000000      tens:     .quad 49  # ASCII code for 1
32      00000000
```

```
  33 0010 39000000     ones:     .quad 57  # ASCII code for 8
  33      00000000
  34 0018 63000000     result:   .quad 99  #  holds the output for debugging and final printing
  34      00000000
  35
  36                 .text
  37                 _start:
  38                 _initialize:
  39                   # clear result
  40 0000 4831C0       xor %rax, %rax
  41 0003 48890425     movq %rax, result    # result should now be 0
  41      00000000
  42
  43                   #load 10 into r10
  44 000b 49C7C20A     movq  $10, %r10
  44      000000
  45
  46                 _hundreds:
  47                   #process hundreds position   The rax should have 0 in it at this point
  48 0012 49F7E2       mul %r10             #multiply rax by 10
  49 0015 488B1C25     movq hundreds,%rbx   #move ascii value of hundreds digit to rbx
  49      00000000
  50 001d 4883EB30     subq $48, %rbx       #subtract 48 to convert to value of digit
GAS LISTING hmwk03.s           page 2


  51 0021 4801D8       addq %rbx, %rax      #add rbx to the accumulator
  52 0024 48890425     movq %rax, result    # DEBUG ONLY result should be 2
  52      00000000
  53
  54
  55
  56                 _exit:
  57 002c 48C7C03C     movq $60, %rax
  57      000000
  58 0033 488B3C25     movq result, %rdi
```

```
58      00000000
59 003b 0F05        syscall
60
61
```