```
 /*
This program converts a 3-digit ASCII code to its decimal equivalent

file: hmwk03.s

author:
date:
Environment:  assemble with GNU assembler (GAS)

Labels in Memory:
    hundreds  -- Represent the first digit in the hundreds position.
    tens      -- Represent the second digit in the 10s position
    ones      -- Represent the third digit in the ones position.
    result    -- Holds the final answer.  It will also hold intermediate results for debugging

Register usage:
    RAX -- Accumulator.  Holds the results of multiplication and added digits
    RBX -- Conversion of the ASCII code to decimal value
    RDX -- Will not be used, but it will be cleared after every multiplication.
    R10 -- Holds 10.  The mul command does not allow immediate literals, so the value 10 must go in a register.  Ri

*/
.globl _start
.data
   # The three-digit number is 218
   hundreds: .quad 50  # ASCII code for 2
   tens:     .quad 49  # ASCII code for 1
   ones:     .quad 57  # ASCII code for 8
   result:   .quad 99  #  holds the output for debugging and final printing

.text
_start:
_initialize:
  # clear result
  xor %rax, %rax
  movq %rax, result    # result should now be 0

  #load 10 into r10
  movq  $10, %r10

_hundreds:
  #process hundreds position   The rax should have 0 in it at this point
  mul %r10             #multiply rax by 10
  movq hundreds,%rbx   #move ascii value of hundreds digit to rbx
  subq $48, %rbx       #subtract 48 to convert to value of digit
  addq %rbx, %rax      #add rbx to the accumulator
  movq %rax, result    # DEBUG ONLY result should be 2

_tens:
  #process the tens position.
                       #multiply rax by 10
                       #move tens to rbx
                       #subtrace 48 to convert value of digit
                       #add rbx to the accumulator
                       #debug ONLY result should should be 21

_ones:
                      #multiply rax by 10
                      ## COMPLETE THIS SECTION

_exit:
  movq $60, %rax
  movq result, %rdi
  syscall
```