Solution:

1. In direct addressing, the effective address is simply the sum of the base address and the offset. In this case:

   Effective Address = Base Address + Offset
   Effective Address = 0x2000 (Base Address) + 0x003A (Offset)
   Effective Address = 0x203A

   So, the effective address for the variable located at offset 0x003A using direct addressing mode is 0x203A.

   ANS - 203A

2. In Little Endian byte order, the least significant byte (LSB) is stored at the lowest memory address, and the most significant byte (MSB) is stored at the highest memory address. When calculating the effective memory address using PC relative addressing, we need to consider both the program counter and the offset.

   Here's how you can calculate the effective memory address:

   1. Program Counter (PC) contains the address of the instruction itself, which is `0x1012`.
   2. The instruction specifies an offset of `0x00A8`.

   Now, let's calculate the effective memory address:

   Effective Address = PC + Offset

   Effective Address = `0x1012` (PC) + `0x00A8` (Offset)

   Effective Address = `0x1012` + `0x00A8`
   Effective Address = `0x10BA`

   So, when the instruction at address `0x1012` with a PC relative offset of `0x00A8` is executed, it will access the effective memory address `0x10BA` in Little Endian byte order.

   ANS - 10BA

3. Each instruction consists of the following fields:

   - Opcode: 6 bits
   - Source Register Identifier A: 5 bits
   - Source Register Identifier B: 5 bits

- Immediate Value: 16 bits

So, the size of each instruction in bits is: 6 + 5 + 5 + 16 = 32 bits.

Since there are 8 bits in 1 byte, we need to convert the size from bits to bytes: 32 bits / 8 bits per byte = 4 bytes per instruction.

Now, we have 200 instructions in the program. To find the total memory consumption, multiply the size of one instruction by the number of instructions: Total Memory Consumption = 4 bytes/instruction * 200 instructions = 800 bytes.

So, the program text consumes 800 bytes of memory.

ANS - 800