On my honor, I have not given, nor received, nor witnessed any unauthorized assistance on this work.

Print name and sign: _____

Question:	1	2	3	4	Total
Points:	7	7	6	10	30
Score:					

1. (7 points) Dr. Summet has been asked to give an expert mini-lecture on base and bounds addressing. Here's what she says:

   "Base/bounds-based virtual memory is really easy. Imagine you have a base register and a bounds register in each CPU. The base points to the physical memory location where an address space is relocated; the bounds tells us how big such an address space can be. Let's do an example to understand this better.

   Assume we have the following base/bounds pair:

   ```
   Base : 0x1000
   Bounds : 0x10
   ```

   Now assume we have the following virtual memory references by a process (in this order):

   ```
   0x0, 0x4, 0x8, 0xc
   ```

   The corresponding physical addresses that will be referenced are (in this order):

   ```
   0x1000, 0x1004, 0x1008, FAULT (because this one is out of bounds)
   ```

   Make sense?"

   Not to put too fine a point on it, but Prof. Summet isn't correct and has several errors in her explanation. Point out her errors and correct them.

2. Assume a system is using base and bounds with the following system characteristics:

   - a 1KB (1024 bytes) virtual address space
   - a base register set to 10000
   - a bounds register set to 100

   For each of the following *physical memory* locations, give the corresponding virtual address translation or state that the physical memory location could not be legally accessed by the running program.

   (a) (1 point) 0 _____

   (b) (1 point) 1000 _____

   (c) (1 point) 10000 _____

   (d) (2 points) 10050 _____

   (e) (2 points) 10100 _____

3. (6 points) Segmentation is a generalization of base-and-bounds. Give one advantage of segmentation over base and bounds. Then give one disadvantage.

4. (10 points) Assume the following in a simple segmentation system that supports two segments: one (positive growing) for code and a heap (Segment 0), and one (negative growing) for a stack (Segment 1):

   - Virtual address space size 128 bytes
   - Physical memory size 512

   Segment register information:

   - Segment 0 base (grows positive) : 0
   - Segment 0 limit : 20 (decimal)
   - Segment 1 base (grows negative) : 0x200 (decimal 512)
   - Segment 1 limit : 20 (decimal)

   Circle all of the following which are valid *virtual memory* accesses

   A. `0x1d` (decimal: 29)

   B. `0x7b` (decimal: 123)

   C. `0x10` (decimal: 16)

   D. `0x5a` (decimal: 90)

   E. `0x0a` (decimal: 10)