Question 3 (20 points):

```
1 vecAdd1: add   $v0, $0, $0
2          lui   $t1, 0x8000
3          lui   $t2, 0x9000
4 next:    lw    $t3, 0($t1)
5          add   $v0, $v0, $t3
6          addi  $t1, $t1, 4
7          bne   $t1, $t2, next
8          jr    $ra
```

(a) Code for `vecAdd1`

```
10 vecAdd2: add   $v0, $0, $0
11          lui   $t1, 0x8000
12          lui   $t2, 0xC000
13 next:    lw    $t3, 0($t1)
14          add   $v0, $v0, $t3
15          addi  $t1, $t1, 16
16          bne   $t1, $t2, next
17          jr    $ra
```

(b) Code for `vecAdd2`

Figure 1: Two versions of a function that sum elements of a vector.

Figure **??** shows two versions of a code that returns the sum of elements of a vector. Both versions of this code are executed in a processor with a 16KB L1 Data Cache with 16-byte cache blocks.

1. (**5 points**) Assume that this is a 32-bit address machine. How many elements of the vector are accessed by `vecAdd1` and how many elements of the vector are accessed by `vecAdd2`?

2. (**5 points**) If the L1 Data Cache is directly mapped, what is the hit ratio for the L1 Data Cache for `vecAdd1` and for `vecAdd2`?

3. (**5 points**) What is the effect in the hit ratios if the L1 Data Cache retains the same total data storage of 16KB but is made two-way set associative?

4. (**5 points**) Assume that this machine has a 32-bit address bus. If the L1 Data Cache is two-way set associative, how many bits are used for each of the following components of a data cache access?

   - Offset:
   - Index:
   - Tag: