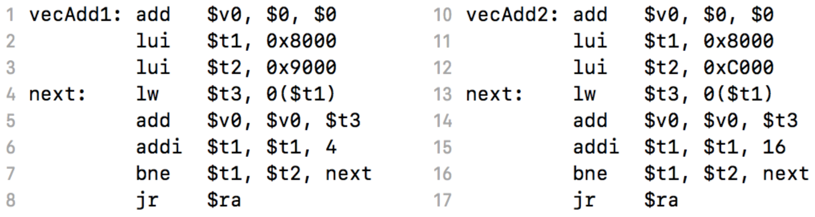
Question 3 (20 points):



```
vecAdd1: add    $v0, $0, $0
         lui    $t1, 0x8000
         lui    $t2, 0x9000
next:    lw     $t3, 0($t1)
         add    $v0, $v0, $t3
         addi   $t1, $t1, 4
         bne    $t1, $t2, next
         jr     $ra
```

(a) Code for `vecAdd1`

```
vecAdd2: add    $v0, $0, $0
         lui    $t1, 0x8000
         lui    $t2, 0xC000
next:    lw     $t3, 0($t1)
         add    $v0, $v0, $t3
         addi   $t1, $t1, 16
         bne    $t1, $t2, next
         jr     $ra
```

(b) Code for `vecAdd2`

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

Figure 1 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`?

   VecAdd1: (`0x9000 0000`-`0x8000 0000`)$/4 = 2^{28} - 2^{2} = 2^{26} = 64 \times 1024 \times 1024 = 67108864$
   VecAdd1: (`0xC000 0000`-`0x8000 0000`)$/16 = 2^{30} - 2^{4} = 2^{26} = 64 \times 1024 \times 1024 = 67108864$

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`?

   There is not temporal data reuse in either program. Thus, all the hits are because of the size of the cache block.
   For `vecAdd1` every fourth data access will be a miss. Therefore the hit ratio is 75%.
   `vecAdd2` access every fourth element of the array and thus, its hit ratio is 0%.

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?

   The hit ratio does not change because there are no conflict misses in the direct mapped cache. Higher associativity has no effect on cold misses.

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: 4 bits
   - Index: $\frac{16KB}{16\times2} = 512$ sets $\Rightarrow$ 9 bits
   - Tag: 32-9-4 = 19 bits