Cache Performance

Question 1:

Suppose our processor has separate L1 instruction cache and data cache. Our CPI_{base} is 2 clock cycles, whereas memory accesses take 100 cycles. Our I$ miss rate is 3% while our D$ miss rate is 10%. 40% of our instructions are loads or stores.

a. **What is our processor's CPI_{stall}?**

To improve the performance our processor, we add a unified L2 cache between the L1 caches and memory. Our L2 cache has a hit time of 10 cycles and a global miss rate of 2%.

b. **What is our new CPI_{stall}?**

Question 2:

You want your AMAT to be <=2 cycles. You have two levels of cache.
L1 Hit Time is 1 cycle; L1 miss rate is 10%;
L2 Hit Time is 3 cycles; L2 Miss Penalty is 100 cycles
What must you optimize your L2 miss rate to be?

Cache Question (From Su 98 Final, Question 4)

1. A 32kB cache has a linesize of 16 bytes and is 4-way set-associative. How many bits of an address will be in the Tag, Index, and Offset?

2. In a 2-way set-associative cache, three addresses, A, B, and C, all have the same index but distinct tags. What is a minimum sequence of accesses which, if repeated, will maximize the miss rate in the cache if it uses the LRU replacement policy?

3. If the above sequence is repeated for a long period of time, what will the miss rate be if the cache uses an LRU replacement policy?

4. If the hit time is 1 cycle, and the miss penalty is 3 cycles, what will be the average memory access time (in clock cycles) for the LRU replacement policy using the above sequence?

5. If the sequence is repeated for a long period of time, will the miss rate be improved if random is used as the replacement policy?

MIPS Mystery Question (From Fa 10 Midterm)

What does the assembly function mystery return? Write your answer as a binary number.

Address		Instruction
0x08001000	mystery:	addiu $sp, $sp, -4
0x08001004		sw $ra, 0($sp)
0x08001008		addiu $v0, $zero, 0
0x0800100c		jal inner
0x08001010		lw $ra, 0($sp)
0x08001014		addiu $sp, $sp, 4
0x08001018		jr $ra
0x0800101c	inner:	lw $v0, 4($ra)
0x08001020		jr $ra

Cache Question (From Fa 11 Midterm)

Take a look at the following C function **sum_iter** run on a 32-bit MIPS machine. On this system, these **structs** are aligned to two - word boundaries since **sizeof(struct Node) = 8**. Assume the total space taken up by the linked list is greater than (and a multiple of) the cache size.

```
struct Node {
        int n;
        struct Node *next;
};
int sum_iter (struct Node *head) {
        int sum = 0;
        while (head != NULL) {
                sum += head - >n; // load from head+0
                head = head - >next; // load from head+4
                }
        return sum;
}
```

Given a direct - mapped data cache with this configuration: **INDEX: 13 bits, OFFSET: 7 bits**

a. How many words are in a block?

b. How many bytes of data does this cache hold? (in IEC format)

Define A and B as your answers to (a) and (b) above, respectively. For questions (c) and (d) below, use these variables in your answer if necessary. Also, when we mention hit rate below, we're talking about accessing data (not instructions).

c. What is the lowest possible cache hit rate for the **while** loop in **sum_iter**?

d. What is the highest possible cache hit rate for the **while** loop in **sum_iter**?

e. To achieve this maximum hit rate, we obviously could have every Node next to every other node, like an array. Ho wever, that's too strict a constraint -- we can still achieve this hit rate if that's not the case. What's the loosest constraint for how the Nodes are distributed in memory to get the best hit rate?