

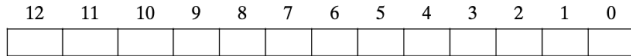
B. Memory reference:

Parameter	Value
Block offset (CO)	0x _____
Index (CI)	0x _____
Cache tag (CT)	0x _____
Cache hit? (Y/N)	_____
Cache byte returned	0x _____

6.33 ♦♦

Repeat Problem 6.32 for memory address 0x16E8.

A. Address format (one bit per box):



B. Memory reference:

Parameter	Value
Cache offset (CO)	0x _____
Cache index (CI)	0x _____
Cache tag (CT)	0x _____
Cache hit? (Y/N)	_____
Cache byte returned	0x _____

6.34 ♦♦

For the cache in Problem 6.31, list the eight memory addresses (in hex) that will hit in set 2.

6.35 ♦♦

Consider the following matrix transpose routine:

```

1  typedef int array[4][4];
2
3  void transpose2(array dst, array src)
4  {
5      int i, j;
6
7      for (i = 0; i < 4; i++) {
8          for (j = 0; j < 4; j++) {
9              dst[j][i] = src[i][j];
10         }
11     }
12 }
```

Assume this code runs on a machine with the following properties:

- sizeof(int) == 4.
- The src array starts at address 0 and the dst array starts at address 64 (decimal).
- There is a single L1 data cache that is direct-mapped, write-through, write-allocate, with a block size of 16 bytes.
- The cache has a total size of 32 data bytes and the cache is initially empty.
- Accesses to the src and dst arrays are the only sources of read and write misses, respectively.

A. For each row and col, indicate whether the access to src[row][col] and dst[row][col] is a hit (h) or a miss (m). For example, reading src[0][0] is a miss and writing dst[0][0] is also a miss.

		dst array			
		Col 0	Col 1	Col 2	Col 3
Row 0	m				
Row 1					
Row 2					
Row 3					

		src array			
		Col 0	Col 1	Col 2	Col 3
Row 0	m				
Row 1					
Row 2					
Row 3					

6.36 ♦♦

Repeat Problem 6.35 for a cache with a total size of 128 data bytes.

		dst array			
		Col 0	Col 1	Col 2	Col 3
Row 0					
Row 1					
Row 2					
Row 3					

		src array			
		Col 0	Col 1	Col 2	Col 3
Row 0					
Row 1					
Row 2					
Row 3					

6.37 ♦♦

This problem tests your ability to predict the cache behavior of C code. You are given the following code to analyze:

```

1  int x[2][128];
2  int i;
3  int sum = 0;
4
5  for (i = 0; i < 128; i++) {
6      sum += x[0][i] * x[1][i];
7  }
```