

Cache Hit/Miss Rates (12 points):

Consider the following function, which takes the dimension D of two square arrays to process, and pointers to the arrays:

```
void procArr(int D, int A[D][D], int B[D][D]) {
    int i, j, res;
    res = 0;
    for (i=0; i<D; i++) {
        for (j=0; j<D; j++) {
            res += A[j][i] + B[i][j];
        }
    }
}
```

For the following questions, only consider memory accesses to the arrays in `procArr` — i.e., assume that all other variables are mapped to registers — and that `ints` are 4 bytes wide.

WP1 (a). Given a direct-mapped data cache with 8-byte blocks and 4 total lines, indicate which accesses in the argument arrays `A` and `B` with dimensions $D=2$ are hits, misses, and miss-evictions by marking each cell on the worksheet with either `H` (hit), `M` (miss), or `E` (miss-eviction).

Arrays `A` and `B` begin at memory addresses `0x601000` and `0x602008`, respectively. Assume that the cache starts out empty.

WP1 (b). Repeat part (a), but with arrays of dimension $D=3$.

This time, assume arrays `A` and `B` begin at memory addresses `0x601000` and `0x602020`, respectively.

WP1.

(a)

A	0	1
0		
1		

B	0	1
0		
1		

(b)

A	0	1	2
0			
1			
2			

B	0	1	2
0			
1			
2			