

## 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, sum;
    sum = 0;
    for (i=0; i<D; i++) {
        for (j=0; j<D; j++) {
            sum = A[i][j] + B[D-i-1][D-j-1];
        }
    }
}
```

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 16-byte blocks and 4 total lines, indicate which accesses in the argument arrays A and B with dimensions  $D=4$  are hits and misses by marking each cell on the worksheet with either H (hit) or M (miss).

Arrays A and B begin at memory addresses `0x601080` and `0x602000`, respectively. Assume that the cache starts out empty.

**WP1 (b).** Repeat part (a), but with  $D=5$ .

**WP1.**

(a)

A	0	1	2	3
0				
1				
2				
3				

B	0	1	2	3
0				
1				
2				
3				

(b)

A	0	1	2	3	4
0					
1					
2					
3					
4					

B	0	1	2	3	4
0					
1					
2					
3					
4					