The heart of the game SimAquarium is a tight loop that calculates the average position of 256 algae. You are evaluating its cache performance on a machine with a 1024-byte direct-mapped data cache with 16-byte blocks (B = 16). You are given the following definitions:

```
struct algae_position {
float x;
float y; };

struct algae_position grid[16][16];

float total_x = 0, total_y = 0;
int i,j;
```

You should also assume the following:
•sizeof(float) == 4 (Bytes).
•grid begins at memory address 0.
•The cache is initially empty.
•The only memory accesses are to the entries of the grid array grid.
The variables i, j, total_x, total_y are stored in registers.

We consider the following code:

```
for (i = 0; i < 16; i++) {
     for (j = 0; j < 16; j++) {
          total_x += grid[j][i].x;
          total_y += grid[j][i].y;
     }
}
```

Analyze the cache performance of this code (provide some short explanations so we see how you got the result; it helps to draw the cache):
(a) What is the total number of reads?
(b) What is the total number of reads that miss in the cache?
(c) What is the miss rate?
(d) Would the miss rate be if the cache were twice as big?
(e) How will the answer (a)-(d) change if the looping order is swapped (first over i, then over j)