

```

5
6 struct algae_position grid[16][16];
7 int total_x = 0, total_y = 0;
8 int i, j;

```

还有如下假设：

- `sizeof(int)==4`。
- `grid` 从内存地址 0 开始。
- 这个高速缓存开始时是空的。
- 唯一的内存访问是对数组 `grid` 的元素的访问。变量 `i`、`j`、`total_x` 和 `total_y` 存放在寄存器中。


确定下面代码的高速缓存性能：

```

1     for (i = 0; i < 16; i++) {
2         for (j = 0; j < 16; j++) {
3             total_x += grid[i][j].x;
4         }
5     }
6
7     for (i = 0; i < 16; i++) {
8         for (j = 0; j < 16; j++) {
9             total_y += grid[i][j].y;
10        }
11    }

```

- 读总数是多少？
- 缓存不命中的读总数是多少？
- 不命中率是多少？


 **练习题 6.19** 给定练习题 6.18 的假设，确定下列代码的高速缓存性能：

```

1     for (i = 0; i < 16; i++){
2         for (j = 0; j < 16; j++) {
3             total_x += grid[j][i].x;
4             total_y += grid[j][i].y;
5         }
6     }

```

- 读总数是多少？
- 高速缓存不命中的读总数是多少？
- 不命中率是多少？
- 如果高速缓存有两倍大，那么不命中率会是多少呢？

 **练习题 6.20** 给定练习题 6.18 的假设，确定下列代码的高速缓存性能：

```

1     for (i = 0; i < 16; i++){
2         for (j = 0; j < 16; j++) {
3             total_x += grid[i][j].x;
4             total_y += grid[i][j].y;
5         }
6     }

```

- 读总数是多少？
- 高速缓存不命中的读总数是多少？