Cache Problem Solution

The following C program is run on a machine with a cache that has four words per block and holds 256 bytes of data (i.e. total size of cache).

```c
int i, j, c, stride, myArray[256];
...
for (i = 0; i < 1000; i ++){
    for (j = 0; j < 256; j = j + stride){
        c = myArray[j] + 5;
    }
}
```

Assume the following:
- word size = sizeof(int) = 4 bytes
- myArray begins at memory address 0
- cache is initially empty
- variables i,j, and c, and stride are stored in registers

a. What is the miss rate (total misses/total references), if stride = 132 (i.e. 10000100₂) ?

(Hint: Figure out block bits and word bits).

First figure out cache information:
- 4 words per block, and each word = 4 bytes; therefore block size = 16 bytes
- Number of blocks in cache = Total Size/ Block size = 256/16 = 16
- (Note that we are ignoring the tag bits in block size)

Split of memory address:
- Word = log₂(4) = 2 bits
- Block/Index = log₂(16) = 4
- We don’t know the total number of address bits, but we will assume that any bit beyond 6 bits (2 + 4) is tag

For program:
- For each iteration of i, j is first 0 and then j = 132.
- j = 0 -> memory address in base 2 is 00_0000_00 -> cache index = 0
- j = 132 -> memory address in base 2 is 10_0001_00 -> cache index = 1
- Since cache is initially empty, we have two misses on the first reference. However after that we will have no misses.
- 2 references (i.e. locations 0 and 132) per value of i.
- Therefore, total references = 2 x 1000 = 2000
- Miss rate = Total misses/Total references = 2/2000 = 0.001
b. What is the miss rate, if stride = 131 (i.e. 100000112) ?

For program:
For each iteration of i, j is first 0 and then j = 131.
  j = 0 -> memory address 00_0000_00 -> cache index = 0
  j = 131 -> memory address 10_0000_11 -> cache index = 0
Since cache is initially empty, we have two misses on the first reference. However, the 2 misses are conflict misses as both address 0 and 131 map to same cache location. Hence we will have to 2 misses every iteration of i.
Miss rate = Total misses/Total references = 2000/2000 = 1

c. Would the miss rate from part (a) and (b) change if the cache were 2-way set associative?

With 2-way cache, the number of sets = 16/2 = 8
Therefore bits needed for block = 3 bits

For stride = 132
  j = 0 -> memory address 000_000_00 -> cache index = 0
  j = 132 -> memory address 100_001_00 -> cache index = 1
The miss rate is same as address map to different cache locations.

For stride = 131
  j = 0 -> memory address 000_000_00 -> cache index = 0
  j = 131 -> memory address 100_000_11 -> cache index = 0
They map to the same set but there two blocks per set and hence we will have two initial misses and then no misses after that. Therefore the miss rate changes to 0.001