For the following two sequence of memory read accesses (byte addressed memory) and cache parameters given, determine the best remaining parameter to find the lowest miss rate. Assume that this memory system only has a L1 cache. Block size must be a power of 2. Access addresses are in hexadecimal.

1. Access: 0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38
   Total Cache Size = 32 bytes
   Associativity = 1 (direct mapped)
   Block Size = \(32\) \(B\)
   Miss Rate based on the above parameter = \(\frac{2}{8 \text{ accesses}} = 25\%\)

2. Access: 0x00, 0x40, 0x20, 0x10, 0x02, 0x44, 0x23, 0x15
   Total cache size = 32 bytes
   Associativity = \(4\)-way
   Block Size = 8 bytes
   Miss Rate based on the above parameter = __________
Everything else is the same, higher associativity means fewer misses.

DM  2-way  FA

Total of 256 blocks

0, BS, 2BS, 3BS ... 256 BS

100% Miss FA

$\frac{1}{257}$ Miss DM

$\frac{1}{257}$ Miss N-way
Describe what is the forwarding unit doing for the following instructions in each cycle:

ADDI X1, X1, #20
STUR X1, [X2, #4]
LDUR X2, [X1, #8]
SUBS X3, X1, X2
B.LT END
ADD X4, X1, X3

<table>
<thead>
<tr>
<th></th>
<th>Ifetch</th>
<th>Reg</th>
<th>Exec</th>
<th>Mem</th>
<th>Wr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STUR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDUR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.LT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- X1 ADD -> STUR
- X2 ADD -> LDUR
- X3 SUBS -> ADD
- X1 ADDI -> SUBS
- X1 ADDI illegal
- Flags SUBS -> B.LT
- Cycle 3: 6
- Cycle 4: 7
- Cycle 5: 8
- Cycle 6: 9
- Cycle 7: 10

**Forwarding Unit from Class:**

![Forwarding Unit Diagram]
Draw the constraint graph for the following code.

8: STOR x
7: ADD x, xo, 1
6: SUBI x, [x, xo]
5: LDUR x
4: STOR x
3: ADD x, xo, 3
2: ADD x, 1, [x, 1]
1: LDUR x
0: ADD x, xo, 1

Code.
2-way ULIM

<table>
<thead>
<tr>
<th>ALU/LD</th>
<th>ALU/BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: LDUR</td>
<td>3: SUBI</td>
</tr>
<tr>
<td>4: STUR</td>
<td>2: ADD</td>
</tr>
<tr>
<td>5: LDUR</td>
<td>8: STUR</td>
</tr>
<tr>
<td>7: ADD</td>
<td>8: STUR</td>
</tr>
<tr>
<td>6: ADD</td>
<td>7: ADD</td>
</tr>
</tbody>
</table>

Loop
For the code below, would a 1-bit predictor or 2-bit predictor be more accurate at predicting the "if (i%3==0)" branch? Determine the branch prediction accuracy of both.

```c
int count = 0;
while (count < 100) {
    if (count % 2 == 0) {
        for (int i = 0; i < 10; i++) {
            if (i % 3 == 0) {
                count++;
            }
        }
    }
    count++;
}
```

1-bit
\[
\frac{4}{10} = 40\%
\]

2-bit
\[
\frac{5}{10} = 50\%
\]