5. *Fork* (5 points) Consider this code using Linux's `fork`:

```c
int x = 7;
if(fork()) {
    x++;
    printf(" %d ", x);
    fork();
    x++;
    printf(" %d ", x);
} else {
    printf(" %d ", x);
}
```

What are *all* the different possible outputs (order of things printed) for this code? (Hint: There are four of them.)
Question 4: Caches (11 pts)

We have a 64 KiB address space and two different caches. Both are 1 KiB, direct-mapped caches with random replacement and write-back policies. Cache X uses 64 B blocks and Cache Y uses 256 B blocks.

a) Calculate the TIO address breakdown for Cache X:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Index</th>
<th>Offset</th>
</tr>
</thead>
</table>

b) During some part of a running program, Cache Y’s management bits are as shown below. Four options for the next two memory accesses are given (R = read, W = write). Circle the option that results in data from the cache being written to memory.

<table>
<thead>
<tr>
<th>Line</th>
<th>Slot</th>
<th>Valid</th>
<th>Dirty</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1000 01</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0101 01</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1110 00</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000 11</td>
</tr>
</tbody>
</table>

(1) R 0x4C00, W 0x5C00  (2) W 0x5500, W 0x7A00  (3) W 0x2300, R 0x0F00  (4) R 0x3000, R 0x3000

c) The code snippet below loops through a character array. Give the value of LEAP that results in a Hit Rate of 15/16 for Cache Y.

```
#define ARRAY_SIZE 8192
char string[ARRAY_SIZE];   // &string = 0x8000
for(i = 0; I < ARRAY_SIZE; i += LEAP) {
    string[i] |= 0x20;   // to lower
}
```


d) For the loop shown in part (c), let LEAP = 64. Circle ONE of the following changes that increases the hit rate of Cache X:

- Increase Block Size
- Increase Cache Size
- Add a L2$
- Increase LEAP

e) For the following cache access parameters, calculate the AMAT. All miss and hit rates are local to that cache level. Please simplify and include units.

<table>
<thead>
<tr>
<th>L1$ Hit Time</th>
<th>L1$ Miss Rate</th>
<th>L2$ Hit Time</th>
<th>L2$ Hit Rate</th>
<th>MEM Hit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 ns</td>
<td>40%</td>
<td>20 ns</td>
<td>95%</td>
<td>400 ns</td>
</tr>
</tbody>
</table>