1. [2 points] Consider the C program below. Assume that all functions return normally.

```c
main() {
    if (fork() == 0) {
        if (fork() == 0) {
            printf("3");
        } else {
            pid_t pid; int status;
            if ((pid = wait(&status)) > 0) {
                printf("4");
            }
        }
    } else {
        if (fork() == 0) {
            printf("1");
            exit(0);
        }
        printf("2");
    }
    printf("0");
    return 0;
}
```

Out of the 5 outputs listed below, circle only the valid outputs (can be one or more or none) of this program. Assume that all processes run to normal completion.

A. 2030401  B. 1234000  C. 2300140  D. 2034012  E. 3200410
2. Consider the C program below. Assume that all functions return normally.

```c
int main () {
    if (fork() == 0) {
        if (fork() == 0) {
            printf("3");
        } else {
            printf("3");
        }
    } else {
        pid_t pid; int status;
        if ((pid = wait(&status)) > 0) {
            printf("4");
        } else {
            printf("2");
            exit(0);
        }
        printf("0");
        return 0;
    }
}
```

What are the possible output(s) of that program?
3. Consider a computer system that has a cache with 256 blocks. Each block can store 16 bytes. What will be the value stored in the TAG field of the cache block that holds the memory block containing the address 0x3CFBCF (Note: The address provided tells you that the length of an address here is 24 bits)

(i) [3] if it is a direct-mapped cache

(ii) [3] if it is a 16-way set-associative cache

(iii) [3] if it is fully associative

4. [1] What is the output of this program?

```c
int val = 10;

void handler(sig)
{
    val += 7;
    return;
}

int main()
{
    int pid;
    signal(SIGCHLD, handler);
    if ((pid = fork()) == 0) {
        val -= 3;
        exit(0);
    }
    waitpid(pid, NULL, 0);
    printf("val = %d\n", val);
    exit(0);
}
```

5. [6] Assume the following contents of the page table and the cache (all contents are hexadecimal) next page.

Also, assume the following specifications:
- Virtual addresses are 16 bits wide.
- Physical addresses are 12 bits wide.
- The page size is 256 bytes.
- The cache is 2-way set associative with a 4 bytes block size. The total cache size is 64 bytes
- A valid bit of 0 in the page table means a page fault.
The page table:

<table>
<thead>
<tr>
<th>VPN</th>
<th>PPN</th>
<th>Valid</th>
<th>VPN</th>
<th>PPN</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>6</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>5</td>
<td>0</td>
<td>11</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>02</td>
<td>3</td>
<td>1</td>
<td>12</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>03</td>
<td>4</td>
<td>1</td>
<td>13</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>0</td>
<td>14</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>05</td>
<td>7</td>
<td>1</td>
<td>15</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>06</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>07</td>
<td>3</td>
<td>0</td>
<td>17</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>08</td>
<td>5</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>09</td>
<td>4</td>
<td>0</td>
<td>19</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0A</td>
<td>3</td>
<td>0</td>
<td>1A</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>0B</td>
<td>2</td>
<td>0</td>
<td>1B</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>0C</td>
<td>5</td>
<td>0</td>
<td>1C</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>0D</td>
<td>6</td>
<td>0</td>
<td>1D</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0E</td>
<td>1</td>
<td>1</td>
<td>1E</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0F</td>
<td>0</td>
<td>0</td>
<td>1F</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The cache:

<table>
<thead>
<tr>
<th>Index</th>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Tag</th>
<th>Valid</th>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>19</td>
<td>1</td>
<td>99</td>
<td>11</td>
<td>23</td>
<td>11</td>
<td>00</td>
<td>0</td>
<td>99</td>
<td>11</td>
<td>23</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>0</td>
<td>4F</td>
<td>22</td>
<td>EC</td>
<td>11</td>
<td>2F</td>
<td>1</td>
<td>55</td>
<td>59</td>
<td>0B</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>1B</td>
<td>1</td>
<td>00</td>
<td>02</td>
<td>04</td>
<td>08</td>
<td>0B</td>
<td>1</td>
<td>01</td>
<td>03</td>
<td>05</td>
<td>07</td>
</tr>
<tr>
<td>3</td>
<td>06</td>
<td>0</td>
<td>84</td>
<td>06</td>
<td>B2</td>
<td>9C</td>
<td>12</td>
<td>0</td>
<td>84</td>
<td>06</td>
<td>B2</td>
<td>9C</td>
</tr>
<tr>
<td>4</td>
<td>07</td>
<td>0</td>
<td>43</td>
<td>6D</td>
<td>8F</td>
<td>09</td>
<td>05</td>
<td>0</td>
<td>43</td>
<td>6D</td>
<td>8F</td>
<td>09</td>
</tr>
<tr>
<td>5</td>
<td>0D</td>
<td>1</td>
<td>36</td>
<td>32</td>
<td>00</td>
<td>78</td>
<td>1E</td>
<td>1</td>
<td>A1</td>
<td>B2</td>
<td>C4</td>
<td>DE</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>0</td>
<td>A2</td>
<td>37</td>
<td>68</td>
<td>31</td>
<td>00</td>
<td>1</td>
<td>BB</td>
<td>77</td>
<td>33</td>
<td>00</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>1</td>
<td>11</td>
<td>C2</td>
<td>11</td>
<td>33</td>
<td>1E</td>
<td>1</td>
<td>00</td>
<td>C0</td>
<td>0F</td>
<td>00</td>
</tr>
</tbody>
</table>

For each of the following virtual addresses indicate:
- The corresponding physical address (the whole address, PPN and offset) in hexadecimal if there is no page fault.
- Page fault or not
- If not page fault, then cache hit or miss
- Note: The word “index” in the cache (shown above) means “set”.

The list of addresses:
0x0A7F  0x08AB  0x1019  0x101B