1) In your own words, explain what locality of reference is. In temporal locality, we expect code and data to be repeatedly used, so bringing them into cache is beneficial because, once in cache, each later use saves time. But consider the following for-loop with an array access.

```
for(i=0;i<n;i++) a[i] = a[i] + c;
```

Each loop iteration is to a different array element, a[0], a[1], a[2], etc. Once we’ve used a[0], we don’t use it again. So how does array access support locality of reference? That is, when we bring in a[0] to the cache, why is this beneficial? What kind locality of reference does array access support?

2) Suppose a computer has 16 Gigabytes of memory and a cache of 1024 blocks where each block can store 32 words. The word size is 64 bits and memory is word addressable.
   a. What is the format of memory addresses if the cache is direct-mapped?
   b. What is the format of memory addresses if the cache is 2-way set associative?
   c. What is the format of memory addresses if the cache is 8-way set associative?
   d. What is the format of memory addresses if the cache is fully associative?

3) Read problem #11 on page 404. Use the same memory and cache layout and determine for the 20 memory references below which ones are hits and which ones are misses, and compute the hit rate. Assume the cache is already loaded where block 0 has tag 5, block 1 has tag 3, block 2 has tag 5 and block 3 has tag A. The 20 memory references are: 53, 54, 55, 5A, 5B, 5C, AF, B0, B1, C6, C7, C8, B9, 5B, 5C, 5D, 00, 01, 4F, 50. Since the references are in order, if there is a miss, you have to update the cache to keep track of what is currently stored. Your answer need only list whether each of the 20 accesses is a hit or miss (you don’t have to draw the cache as it updates). Also include the cache’s overall hit rate.

4) A processor’s core has an L1 cache and all of the cores share an L2 cache on the chip. These are backed up by DRAM. Assume the L1 cache has a hit time of .4 ns, the L2 cache has a hit time of 1 ns and DRAM has an access time of 10 ns. The L1 cache has a hit rate of 90.6%, L2 has a hit rate of 95.9%, and DRAM has a hit rate of 100%.
   a. What is the effective access time for this processor?
   b. We add an L3 cache to back up L2, and L3 is then backed up by DRAM. The L3 cache has a hit time of 2.5 ns and a hit rate of 98.2%. What is the effective access time now?
   c. Starting from part a (without the L3 cache of part b), we add virtual memory with a TLB with a hit time of .4 ns hit time and hit rate of 97.8%, a page table in memory with the same hit time as DRAM (10 ns) and a 100% hit rate, and a DRAM’s hit rate is now 99.99922% with the swapping process taking 1,000,000 ns. What is the effective access time now?

5) Why will a 2-way set associative cache have a better hit rate than a direct-mapped cache? Why will it be slower?

6) Assume a computer is word addressable with 64M of addresses and is running a process whose page table is given below. Answer the following questions.
   a. How big is a page/frame in words?
   b. What is the physical address for the logical address 10001111111111?
   c. What is the physical address for the logical address 01101010101010?
d. What is the physical address for the logical address 0011111000011?

e. How large is this program (approximately) in words?

f. How much of the program is in memory?

<table>
<thead>
<tr>
<th>Page Number</th>
<th>Frame Number</th>
<th>Valid Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>----</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>----</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>----</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>----</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

7) We are considering making one of two adjustments to a computer. First, we might add an L2 cache to backup the L1 cache. The L1 cache has a miss rate of 15% and its estimated that the L2 cache will have a miss rate of 25%. Note: the L2 cache is used when the L1 cache has a miss, but the speedup occurs when the L2 cache is a hit. If the L2 cache misses, there is no speedup. The L2 cache is 20 times faster than DRAM. Second, we might change the processor from 1 core to 2 cores. When both cores are in use, this makes the processor twice as fast. Both cores will be able to be used 33% of the time. Compute the speedup of both enhancements. Does the common case win out?

8) Answer these questions about the various forms of I/O.

   a. Why is interrupt-driven I/O better than programmed I/O?
   b. What is the byte count, as used in DMA I/O and what happens when this count reaches 0?
   c. What can channel I/O do that DMA I/O cannot?

9) Look at the flowchart in figure 7.4 on page 420. What is the ISR? Where is the ISR located? What does the CPU have to do before it can run the ISR and what does it do after the ISR runs?

10) Imagine that we have 9 disks in a RAID 5 configuration where disk 9 stores a parity byte for the data bytes on the other 8 disks of the same location (track/sector/surface). If the 8 data bytes are as follows, determine the parity byte (assuming even parity): 00110011, 11110000, 10101111, 01001011, 01110111, 00000000, 11001110, 01010100.

11) Answer the following regarding the RAID levels:

   a. Rate the RAID levels which are used in terms of cheapest to most expensive.
   b. Why is RAID 4 not used? How does RAID 5 improve over it?
   c. Since RAID 1 offers full redundancy, if you used RAID 1 would you not need to back up your files in another location? Offer a brief explanation.