1. [10 pts] Tests on an Intel Pentium III processor reveal that the performance of the code at right depends heavily on the value of \( \text{skip} \).

<table>
<thead>
<tr>
<th>( \text{skip} )</th>
<th>( \text{time} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16,352</td>
<td>40 ms</td>
</tr>
<tr>
<td>16,384</td>
<td>480 ms</td>
</tr>
</tbody>
</table>

Surprisingly, with a larger value of \( \text{skip} \), for which the code will add fewer numbers, the code takes twelve times as long. Explain how the cache is making a difference here. (Recall that a Pentium III’s L1 cache is 4-way set associative, with 512 lines, each of 32 bytes. Note that the cache holds \( 512 \times 32 = 16,384 \) bytes.)

```c
int sum(char *arr, int skip) {
    int i, k, total;
    total = 0;
    for(k = 0; k < 4000; k++) {
        for(i = k; i < 4000000; i += skip) {
            total += arr[i];
        }
    }
    return total;
}
```

2. [10 pts] Define the relocation problem that arises in the context of segments.

3. [10 pts] Describe the purpose of the dirty bit found in each page table entry in most virtual memory systems. How does the use of the dirty bit help the system’s performance?