Summing 100,000 Numbers on 100 Processors

Start by distributing 1000 elements of vector A to each of the local memories (in Al) and summing each subset in parallel

```c
sum = 0;
for (i = 0; i < 1000; i = i + 1)
    sum = sum + Al[i]; // sum local array subset */
```

The processors then coordinate in adding together the partial sums (Pn is the number of the processor, send(x,y) sends value y to processor x, and receive() receives a value)

```c
half = 100;
limit = 100;
repeat
    half = (half + 1) / 2; /* dividing line */
    if (Pn >= half && Pn < limit)
        send(Pn-half, sum);
    if (Pn < (limit/2))
        sum = sum + receive();
    limit = half;
until (half == 1); /* final sum in P0’s sum */
```
2. Trace the second segment of code that adds together the partial sums assuming 10 processors.

```
sum  sum  sum  sum  sum  sum  sum  sum  sum  sum  sum
P0  P1  P2  P3  P4  P5  P6  P7  P8  P9
```
3. For a 64 processor system, compare the interconnection network for each of the following topologies.

<table>
<thead>
<tr>
<th></th>
<th>Bus</th>
<th>Ring</th>
<th>Torus</th>
<th>6-cube</th>
<th>Fully connected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network bandwidth</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bisection bandwidth</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total # of Switches</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Links per switch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total # of links</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>