1. [5 points]
Here is a link that can be of help. http://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units

The table below captures some of the information from the above link, subject to the following constraints:

- We restrict our choices, as the question stated, to GeForce GTX
- We restrict our choices to single die (i.e. single GPU in the chip). So entries that have “2x..” are not taken into account.
- If there are several entries for the same item (like memory size), we pick the largest number.

<table>
<thead>
<tr>
<th>Model</th>
<th>#cores</th>
<th>mem size (GB)</th>
<th>mem bw (GB/s)</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeForce GTX 950</td>
<td>768</td>
<td>4</td>
<td>106</td>
<td>2015</td>
</tr>
<tr>
<td>GeForce GTX 960</td>
<td>1280</td>
<td>4</td>
<td>120</td>
<td>2015</td>
</tr>
<tr>
<td>GeForce GTX 1060</td>
<td>1280</td>
<td>6</td>
<td>192</td>
<td>2016</td>
</tr>
<tr>
<td>GeForce GTX 1070</td>
<td>1920</td>
<td>8</td>
<td>256</td>
<td>2016</td>
</tr>
<tr>
<td>GeForce GTX 1080</td>
<td>2560</td>
<td>8</td>
<td>320</td>
<td>2016</td>
</tr>
<tr>
<td>GeForce GTX Titan X</td>
<td>3564</td>
<td>12</td>
<td>480</td>
<td>2016</td>
</tr>
</tbody>
</table>

2. [2 points]
- Number of cores is steadily increasing. The fact that you may find a GPU in a later year with number of cores less than a previous year just means that the company is targeting a low-end market segment with their GPU, not for technological issues.
- The amount of memory does not increase with the same rate, same with the memory bandwidth. This means that the main bottleneck will be the memory access and memory size.

3. [2 points]
- n-body problems
- matrix operations
- neural-networks
- genetic algorithms
- the Black-Scholes (mathematical model for financial market)

[3 points] The main characteristics of the above examples are:
- They involve consuming large amount of data
- Each application can be divided to similar procedures working on different data.
- Not very severe dependencies among the procedures
4. **[2 points]** Yes, because there is a big slowdown caused by moving the data from system memory to GPU memory. If we do not have enough data to ensure a lot of parallelism in the GPU, to overcome this performance loss, we may see *worse* performance for GPU over CPU-only version.

5. **[6 points]**
   a) Yes, each core can be assigned a subset of the numbers and check the existence of the number in them. This is done in parallel.
   b) Cannot be done on GPU because the operations are dependent on each other.
   c) Yes, it is a highly parallel operation.