Task 1

Python code in the accompanied py file.

Output

2 cores

```
15249633789), (True, 0.0050029754638671875
08296966552734), (True, 0.0030028820037841
e, 0.01562643051147461), (True, 0.0), (Tru
011570453643798828), (True, 0.004655361175
7001399993896484), (True, 0.00600171089172
Overall Time: 8.729722 seconds
```

1 core

```
7,00), (True, 0.005001949510302734),

3), (True, 0.005000591278076172), (Ti
4), (True, 0.006001949310302734), (Tri
4), (True, 0.00400090217590332), (Tri
), (True, 0.004001140594482422), (Tri
), (True, 0.00900125503540039), (True,
, (True, 0.0040090217590332), (True,
, (True, 0.005002021789550781), (Trui
), (True, 0.008019447326660156), (Tri
, (True, 0.006001710891723633), (Trui
), (True, 0.006001710891723633), (Trui
), (True, 0.006001710891723633), (Trui
), (True, 0.006001710891723633), (Trui
), (True, 0.006001710891723633), (Trui
)
```

Lessons learned

- 1. When the pool size is 2 the program can run 2 tasks parallel with one another which improves performance by utilising more of the CPU.
- 2. The prime checking function is highly efficient since it takes close to 0 seconds to check each prime number even to 6 decimal places.
- 3. Larger calculations benefit more from greater parallelism. The time difference between using pool size 1 and 2 will be greater for larger calculations.
- 4. When the calculations are small the pool size of one can actually take less time due to the associated overhead costs of parallelism.

Task 2

Python code in the accompanied py file.

1 core

Lessons learned

- 1. Pool size of 2 is faster than pool size of one due to parallelism allowing for more CPU usage
- 2. At lower calculation sizes the time difference between pool size one and two is not as dramatic. I.e parallelism has greater merit for larger calculations.
- 3. At lower calculation sizes multicore processing may actually take longer because of the associated overhead with higher parallelism.
- 4. At very large sizes the speedup may be limited by the serial parts of the processing which cannot be made parallel.