```
1)  Laptop: 4 cores (8 threads)   Intel i7-7700HQ
    Desktop: 4 cores (4 threads)  Intel i5-3570k
    Phone: 4 Kryo 385 Gold + 4 Kryo 385 Silver cores (8 threads)    Snapdragon 845

2) Dennard Scaling:
    Transisters shrink in all dimensions (Area, Delay and Power) and electric field stays constant.

    Dennard Scaling brokedown because voltage thresholds got too close together (leakage problem),
    making values difficult to distinguish. This forced voltage and frequency to stay relatively constant.


3)
Summit: 14.668 TFlops/kW
Sierra: 12.724 TFlops/kW
Sunway TaihuLight: 6.051 TFlops/kW
Tianhe-2A: 3.324 TFlops/kW
Piz Daint: 8.905 TFlops/kW
Trinity: 2.663 TFlops/kW
AI Bridging Cloud Infrastructure: 12.056 TFlops/kW
SuperMUC-NG: Unavailable
Titan: 2.1423 TFlops/kW
Sequoia: 2.177 TFlops/kW


Top 3 Most Efficient:
Summit, Sierra, ABCI

4)
Hosting a web server that dispatches a new thread for every new client that connects.
Can have a rough idea, but won't know exactly how many concurrent (nor total) clients
will connect to the webserver and how many ensuing functions will be called on behalf of
each client.

5)
i = 1;
prod = d[0];

for(i = 1; i <= SIZE; i++){
    prod = prod * v[i];
}

d[SIZE] = prod; //don't need this line if we don't care to store in the array


Because we don't care about storing the intermediary values, we can use an product variable
that builds the final result. This variable is stored in a register and therefore does not
need to be loaded. Because the load of v[i] can occur during the previous multiplication
of prod * v[i-1], the II becomes 1.
```