I Explain three vectorization clauses of your choice that can be used with #pragma omp simd.

Aligned: wie im Beispiel 🡪 Sagt dem Compiler, dass die überreichten Pointer n byte aligned sind.

Collapse: schreibt verschachtelte for-loops zu einer um

Reduction: reduce Funktion, also reduziert alle Elemente eines Arrays in eines durch z.B. addition.

I Give reasons that speak for and against vectorization with intrinsics compared to guided vectorization with OpenMP.

Intrinsics sind präziser und compilerunabhängiger ABER viel schwerer zu schreiben/verstehen, Code wird länger, und das Testen dauert länger, da die Korrektheit erst gegeben ist, wenn wir alle Edge-Cases per Hand schreiben (🡪 also auch höhere Fehlerrate im Durchschnitt).

„the gain of performance portability comes with a loss of code portability“

I What are the advantages of vector intrinsics over assembly code?

* Man muss keine spezifischen Register angeben
* Höhere portability zwischen Comilern
* Intrinsics sind wrapper, also eine andere, einfachere, Ausdrucksweise von Assembly
* Um einiges einfacher zu lernen, schreiben, verstehen usw.

I What are the corresponding vectors of the three intrinsic data types: \_\_m256, \_\_m256d and \_\_m256i.

\_\_m256 ein Vektor mit 8x32-bit floats

\_\_m256d ein Vektor mit 4 doubles

\_\_m256i ein Vektor mit signed oder unsigned integers