Activity 5 - Performance Optimization (30 minutes)

This is based on “Julia for HPC course at TU Delft” by Carsten Bauer.

Let’s optimize a function!

In this activity, we will optimize the following function.

```
function work!(A)
    val = zero(eltype(v))
    for i in 1:N
        val = mod(v[i],256)
        A[i,1:N] = B[i,1:N] * (sin(val) * sin(val) - cos(val) * cos(val))
    end
    A = A/2
    return A
end
```

Initialize variables as (copy-paste it, do not modify it!):

```
using Random
Random.seed!(42)
N = 250
B = rand(N,N)
v = rand(Int, N);
const result = work!(zeros(N,N));
```

Now you show be able to compare your results to `result` as you optimize the function. To check if your result is correct:

```
@test work!(zeros(N,N)) ≈ result
```

Use `@benchmark` to measure performance of `work!(A)`.

Task 1: Try to look for type instabilities (with `@code_warntype`).

Task 2: Try to avoid unnecessary allocations (due to slicing and allocating array computations).

Task 3: Try to optimize the memory access (keyword: column-major order).