TU Berlin, Scientific Computing, WS19/20
Homework assignment #1

Please return this assignment by by Tuesday, Nov. 12. Send it in form of a compressed (zip) file by e-mail to juergen.fuhrmann@wias-berlin.de. Please prefix file and subdirectory names with your last names, e.g. "MuellerNguyenHW01.zip". The file should contain a subdirectory with

- the commented Julia code in form of a module according to this template:

```
module MuellerNguyenHW01
function task1(;optional_parameters)
end
function task2(;optional_parameters)
end
....
end
```

- a pdf describing your answers. Hint: you can use the package Literate.jl to create this pdf from your source (e.g. via Jupyter notebook or via markdown + pandoc)

1 Summation

Write a Julia program which calculates $\sum_{n=1}^{K} \frac{1}{n^2}$ for $K = 10, 100, 1000, 10000, 100000$ and report the values for `Float16`,`Float32`, `Float64`. Compare the results to the value of $\sum_{n=1}^{\infty} \frac{1}{n^2}$ (hint: look it up under "Basel problem").

What can be done in order to improve the accuracy of the calculation ?

2 Heat conduction problem

Given:

- Domain $\Omega = (0,1)$
- Right hand side $f : \Omega \to \mathbb{R}$, $f = 1$
- Boundary values $v_L, v_R = 0$
- Transfer coefficient $\alpha = 1$

Search function $u : \Omega \to \mathbb{R}$ such that

$$\begin{aligned} -u'' &= f \quad \text{in } \Omega \\ -u'(0) + \alpha(u(0) - v_L) &= 0 \\ u'(1) + \alpha(u(1) - v_R) &= 0 \end{aligned}$$

1. Calculate the exact solution of this problem
   - What is the limit of this solution for $\alpha \to \infty$?
2. Implement the finite difference discretization as a linear tridiagonal system on an equidistributed mesh with $N = 2^k + 1$ points with $k = 6 \ldots 14$. Hint: have a look at the slides of https://www.wias-berlin.de/people/fuhrmann/blobs/SciComp-Winter-1920-l05-linsolve.html
3. Use different solution strategies to solve the resulting linear system of equations:
   a) Your implementation of TDMA (Progonka)
   b) Julia dense matrix LU factorization
   c) Julia tridiagonal matrix LU factorization
   d) Julia sparse matrix LU factorization
   e) Multiplication by the inverse
   - Check the results against the exact solution. What happens if N is increased ?
   - Provide timings using the `@benchmark` macro of the Julia BenchmarkTools. Which method is the fastest and why?
   - What is the number of nonzero entries of the inverse matrix ? Why don't we calculate the inverse instead of the LU factorization to solve the linear problem?
   - What happens for increased values of the transfer coefficient $\alpha = 1, 10, 100, 1.0 \cdot 10^5, 1.0 \cdot 10^{10}, 1.0 \cdot 10^{20}$?