National Technical University of Athens
School of Electrical & Computer Engineering
Department of Computer Science
`https://courses.softlab.ntua.gr/pl2/`

Programming Languages II

Unless otherwise stated, exercises should be submitted in electronic form, via the collaborative learning environment `moodle.softlab.ntua.gr`. Deadlines will be strict. You are allowed at most one late exercise.

Exercise 6 Parallel and Concurrent Haskell

Deadline: 2/2/2020

In this exercise you are asked to experiment with *parallelism* and *concurrency* in Haskell. The goal is to write a parallel/concurrent program that achieves some performance improvement, compared to the sequential version, when executed in a multicore architecture.

Task description. In how many ways can we choose K elements out of a set of N elements, if their order is not important? Obviously, the answer is

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

This number may be very large and, for this reason, we want to calculate the remainder (modulo) of its division by a large prime number P.

Input and output. Your program must read from the standard input (stdin) and print the results to the standard output (stdout).

The first line of the input will contain an integer number T, the number of queries that will follow. Each one of the next T lines will contain three space-separated integer numbers, N, K and P. These numbers represent a query that you have to answer. Assume that $0 \leq K \leq N < P \leq 10^9$ and that P is a prime number.

The output must consist of exactly T lines, each of which will contain exactly one integer number: the answer to the corresponding query.

What you must hand in.

1. A sequential program in Haskell, which will be used as the basis for performance comparison.
2. One (or more) parallel or concurrent programs in Haskell, based (alternatively) on one of the following or in combinations thereof:

   a. Parallelism: (a1) Evaluation strategies, (a2) `Par` monad.

   b. Concurrency: (b1) `IO` monad and `MVar`, (b2) Software Transactional Memory.

   The input and output of the parallel or concurrent programs must adhere to the same format as those of the sequential program.

3. One short report that describes your experiences and conclusions. Your report should explain the "parallelization" approaches that you have tried, contain performance diagrams and analyze the performance improvement (scalability) that you achieved, in comparison to the sequential program.