# Mid-term Exam
Due March 10

You are free to use our text, notes and other references, as well as the available computer systems to prepare your solutions to the problems below.

**1. [25 points]**
Binary search is a basic paradigm, but one where details can easily be mistaken. For this problem use Diller's language (chap. 14) augmented with the ability to access arrays in the usual manner (e.g., A[i] denotes the $i^{th}$ element of array A). Write a program fragment for the binary search algorithm that will operate on a sorted array A[1..n] and a value x, and will find an index k so that A[k] = x, and if there is none will leave k = 0; of course, your program will not change x or A. Your code can use the 'div' operation as we normally understand it. The pre-condition is $n \geq 1 \land \forall i:N \bullet 1 \leq i < n \Rightarrow A[i] \leq A[i+1]$, and the post-condition is $(1 \leq k \leq n \land A[k]=x) \lor (k=0 \land \forall i:N \bullet 1 \leq i \leq n \Rightarrow A[i] \neq x)$. Write a loop invariant for your program that is sufficient to support a formal proof of partial correctness, and discuss why your loop invariant is adequate for this. Note that you are not asked to actually provide a program proof, just a loop invariant and a discussion of its relation to the correctness of your code.

**2. [25 points]**
For this problem you are to provide a Z axiomatic description of a (global) function bestApprox : $N \times P\,N \rightarrow N$. The first argument, call it target?, is a natural number, and the second argument, call it approxs?, is a set of natural numbers. The function is to return an integer from approxs?, call it best!, that makes the value |target? – best!| as small as possible. For instance, when target? = 7 and approxs? = {2, 11, 4, 12}, the desired value of best! is 4. Your answer should include justification that your specification meets the required condition. Note: |n| denotes the usual absolute value of an integer in Z.

**3. [25 points]**
(a) For $f: X \rightarrowtail\!\!\!\!\!\!+\; Y$ (i.e., injective partial functions) and $A_1, A_2 \subseteq X$, show that using the relational image of Z, $f(\!| A_1 \cap A_2 |\!) = f(\!| A_1 |\!) \cap f(\!| A_2 |\!)$.
(b) Determine and justify whether the identity in part (a) remains true for $f: X \nrightarrow Y$ (i.e., general partial functions).

**4. [25 points]**
This problem is about modeling one particular component of a multi-server situation such as a bank or supermarket: the collection of service points at which customers may queue into lines to "check-out". At each service point, we are interested in only the sequence of customers waiting to be served.

For purposes of this problem, the server options are to be modeled as a *sequence* of checkout points. The line at each checkout is modeled as a sequence of (given type) Person. For this problem, there can be any number of checkout points and any number of customers in line for each. However, no Person can be in more than one checkout line.

First define a state schema that will serve to describe the set of all possible configurations of the checkout lines. Then define an operation schema, let's call it QueueUp, with arguments consisting of a checkout number, and a Person, that describes the state change when the indicated Person enters the line at the designated checkout point. Finally, define an operation schema, let's call it Exit, with one argument that is the index of a checkout point and that describes the state change when the first Person in line at that point completes their checkout.

Provide convincing evidence that your specification covers all the relevant possibilities and guarantees the desired effects.