Lab session (5th Jan)

General:

- You are allowed only stdio.h as header.
- Use the input handler provided for each problem.

Problem 1

Statement:

- Input: $n \in \mathbb{N}$.
- Goal: Compute $f(n) = ((\sum_{i=0}^{n} i)^2 - \sum_{i=0}^{n} i^2))$
- Output: Print $f(n)$ `\n`

Implementation Rules:

- You are not allowed to use `long`. You can only use the plain old `int` that we taught you in class.

Remark: *Be careful of* `int` *overflow! If $f(n)$ fits inside an* `int`*, then your program should not have an overflow.*

Problem 2

Statement:

- Input: $a, b \in \mathbb{N}$ given as two integers separated by a space.
- Goal: Find the greatest common divisor (gcd) of a and b.
- Output: Print gcd(a, b) followed by `\n`

Implementation Rules:

- Use Euclid's GCD algorithm, described as follows:
  - Without loss of generality, we assume $a < b$.
  - If a divides b, then a itself is the gcd.
  - Else, divide b by a to get remainder r.
  - Find gcd of r and a.

Remark 1: The "*without loss of generality*" is not guaranteed by the inputs of the program.
Remark 2: The above can be written iteratively, or recursively. Usually iteration is faster.

Problem 3:

Statement:

- Input: $n \in \mathbb{N}$ followed by a set A of n natural numbers $a_1, a_2, \ldots, a_n$ and a number $k \in \mathbb{N}$ such that the numbers in A are given in descending order. (For convenience, we assume $n \leq 50$)
- Goal: Check if $k \in A$.
- Output:
  - If $k \in A$, then print 1 `\n`
  - Else print 0 `\n`

Implementation Rules:

- You have to use an idea similar to the bisection method taught in class.

Problem 4:

Homework from lecture 6: compute square root upto a precision parameter using bisection method. (No submission to server)