Functional Programming
Exercises Week 1
(for October 16, 2009)

1. Read Chapter 1 and Appendix A of the lecture notes.

2. Consider the following piece of C code:

```
int calls = 0;

void call(){ calls++; }

int numberofcalls(){ return calls; }

int readint(){
  int n;
  printf("n␣=␣");
  scanf("%d", &n);
  return n;
}
```

Determine whether the functions `call`, `numberofcalls`, and `readint` have side-effects and/or are pure.

3. For each of the basic types as well as `coord`, `direction`, and `number` (as defined in the lecture notes) give at least one value of that type.

4. Consider the algebraic datatype

```
type shape = Rect of float * float | Square of float | Circle of float
```

where `Rect(`a`,`b`)` represents a rectangle of length a and width b, `Square` a is a square of side length a, and `Circle` r denotes a circle with radius r.

Use pattern matching to define the functions `circum : shape -> float` computing the circumference and `area : shape -> float` computing the area of the given geometric shape.

5. Use recursion to define a function `gcd : int -> int -> int` computing the greatest common divisor of two integers.

6. Consider the following recursive definition for computing binomial coefficients:

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n; \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{otherwise;} \end{cases}$$

for $n \geq k \in \mathbb{N}$. Define a recursive function that computes binomial coefficients. Make sure your function terminates for all possible arguments.