

Homework # 5

due Monday, March 2nd, 11:00 AM

1 Simple Programming

Solve the following **one line** programming problems: Exercises 10, 18, 19 of Chapter 9. Your answers should not use recursion, but should instead use the `foldr` function to do the work. Your code for `min` is permitted to crash (throw an exception) if passed an empty list. Put your solutions in file `oneline.sml`.

2 Ring

In “group theory” (a branch of mathematics), a *ring* consists of two operations “addition” and “multiplication” as well as two constants “zero” and “one.” We can declare a ring type constructor:

```
type 'a ring = ('a * 'a -> 'a) * ('a * 'a -> 'a) * 'a * 'a;
```

Now we can declare a ring for integers and a ring for reals:

```
val intRing : int ring = (op +, op *, 0, 1);
val realRing : real ring = (op +, op *, 0.0, 1.0);
```

The polynomial routines written for Homework #4 can be generalized to work for any ring. Starting with your code (or the solution to Homework #4), write curried versions of the functions `eval`, `pow` and `polyToString` that work for any ring. Their types should be as follows:

```
eval : 'a ring -> 'a list -> 'a -> 'a
pow : int -> 'a ring -> 'a list -> 'a list
polyToString : 'a ring -> ('a -> string) -> 'a list -> string
```

Thus we can write:

```
- fun polyCube r = pow 3 r;
val polyCube = fn
  : ('a * 'a -> 'a) * ('a * 'a -> 'a) * 'a * 'a -> 'a list -> 'a list
- polyToString intRing Int.toString (polyCube intRing [1,2]);
val it = "8x^3 + 12x^2 + 6x + 1" : string
- val p = polyCube realRing [1.0,2.0];
val p = [1.0,6.0,12.0,8.0] : real list
- polyToString realRing Real.toString p;
val it = "8.0x^3 + 12.0x^2 + 6.0x + 1.0" : string
- val func = eval realRing p;
val func = fn : real -> real
- func 0.1;
val it = 1.728 : real
```

Put your solution in file `newpoly.sml`.

Optional Thought: For those of you interested in mathematics, it happens that polynomials form a ring too. What would the ring of integer polynomials look like? What would a polynomial of polynomials look like?

3 Paper

Do Exercises 2 and 4 on page 163. Turn your answers in on paper. Alternately (to Ex. 4), try to compile this code as C++. What goes wrong? Explain all the problems!