Functional Programming

1. Reduction, Specification and Variant

Consider the following function declaration:
fun product n =
If n = 1 then
1
else
n ∗ product (n−1)

1. Give a detailed step-by-step evaluation of product 3, Every step should be a valid SML expression.
2. What does the function compute?
3. Write a specification for the function.
4. Give a variant for the function.

2. Currying

Consider this function declaration:
fun p l u s x y = x + y
• Write the function declaration as a value declaration val plus = ... Your declaration should be equivalent to the declaration above.
• What happens when the declaration val foo = plus 4 5 is entered?
• What happens when the declaration val bar = plus 4 is entered?
• Give a step-by-step evaluation of plus 3 4.

3. Types

Give functions with the following types:
1. int −> int
2. int −> int −> int
3. int −> int ∗ int
4. int ∗ int −> int
5. int −> real −> string −> string
6. int ∗ (string ∗ string ∗ int) −> int ∗ string
In each case, try to find a function that is defined for all possible input values, and where the result depends on all parameters. Name the functions funN, where N is replaced by the type number in the list above.

4. Divisibility

2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder.
Give an SML definition for a function lcm n that returns the smallest positive number that is evenly divisible (i.e., divisible without remainder) by all of the numbers from 1 to n. Use auxiliary functions as appropriate.
You must decide how to handle the case when n < 1.

5. Testing

Use the file lab1 test.sml to test your solution.
1. Place lab1 test.sml in the same directory as your solution file lab1.sml.
2. Start Poly/ML in that directory: e.g., cd lab1dir; poly.
3. Enter use "lab1 test.sml"; at the Poly/ML prompt to run the tests.
4. Check the output for failing tests.
5. Quit Poly/ML (e.g., by typing Ctrl+D).
6. Fix all failing tests by modifing your lab1.sml file.
7. Repeat steps 2.-6. until all tests succeed.