

CSE 307 - Principles of Programming Languages
Fall, 2019
Homework Assignment 02
Having fun with SML

Assigned: 09/18/2019
Due: 09/27/2019, at 11:55 PM
Total Points: 35

Solve each of the following problems using SML.

Problem 1 (5 pts.):

Write a function in SML that computes Fibonacci numbers. The input to the function will be an integer. The output will be the corresponding Fibonacci number for the input.

Include several calls to your Fibonacci function with different values. What is the smallest value for which you get an "uncaught exception Overflow" raised? (Put answer in comment.)

Don't worry about inputs < 1

Problem 2 (10 pts. total):

Part A (4 pts.):

Write an SML function that takes as input a list, and then cycles the list once. That is, given a list [a₁, a₂, a₃, ..., a_n] your function returns the list [a₂, a₃, ..., a_n, a₁].

Part B (6 pts.):

Write a recursive SML function that takes two inputs, an integer, 'i', and a list, 'l'. The function will produce a list that has been cycled i times.

For example, if the inputs are: 3, [1, 2, 3, 4, 5, 6]
The list produced by the function should be: [4, 5, 6, 1, 2, 3]

You must demonstrate the functions defined in parts A and B on at least 2 non-empty lists.

Problem 3 (5 pts.):

Write a recursive SML function that takes a list as input and builds a new list that contains a duplicate of each element of the input list. That is, given the list: [a₁, a₂, ..., a_n] the function will produce the list: [a₁, a₁, a₂, a₂, ..., a_n, a_n].

You must demonstrate the use of your function on at least 2 non-empty lists in your program.

Problem 4 (5 pts):

Write a set of mutually recursive functions that can be used to split a list into three (3) separate lists.

You can define more than two mutually recursive functions at a time linked by the "and" keyword. Each function in the set only has to call one of the other functions in the set for the entire set to be mutually recursive.

You must demonstrate the use of your function to split a list into three separate lists in your program.

Problem 5 (10 pts):

Write a higher-order function, *F*, whose inputs are a list of functions and a list of values . *F* will apply each function from the first list to the corresponding value in the second list. The result of each function application will be preserved in a tuple, where the first element is the input to the function, and the second element is the result. The final output from the higher-order function *F* will be a list whose elements are the ordered pairs.

For example:

```
F([g1, g2, g3][4, 5, 6]);  
will produce: [(4, g1(4)), (5, g2(5)), (6, g3(6))]
```

You must demonstrate your higher-order function on at least one set of inputs in your program.

Submission Instructions:

1. Please write each program in a separate file.
2. The file name should correspond to the problem for which your program is written.
Example: Your program for problem 1 should be called "p1.sml".
3. Please collect and submit your program files as a compressed zip file.
4. The title of the compressed file should be:
Example: cse307_hw02-LastNameFirstName
5. This is an individual assignment. Any collaboration on writing your programs will be treated as a violation of academic integrity.