

CSE 305 Programming Languages Spring, 2010

Homework 9

Maximum Points: 15

Due 10:30 AM, Friday, April 9, 2010

Professor Shapiro

April 2, 2010

Write the answers in a file named `hw9.txt`. Put your name and user name at the top of the file, and submit the file, using the UNIX command, `submit_cse305 hw9.txt`. You will be directed to submit 3 more files below.

1. (5) Early in school, I had to memorize the multiplication tables from 1 to 12. I hope you did too. Use the Haskell interactive shell, `ghci` and a single list comprehension expression to print a piece of the multiplication tables, namely from 6 to 8, in the format shown below. You can start `ghci` by entering `ghci` as a Unix command, and exit by entering `:quit` as a `ghci` command. Here's an example of what you should be able to do:

```
<timberlake:~:1:27> ghci
GHCi, version 6.12.1: http://www.haskell.org/ghc/  :? for help
Loading package ghc-prim ... linking ... done.
Loading package integer-gmp ... linking ... done.
Loading package base ... linking ... done.
Loading package ffi-1.0 ... linking ... done.
Prelude> _____<your list comprehension expression here>_____
[ ((6,6),36),((6,7),42),((6,8),48),((7,7),49),((7,8),56),((8,8),64) ]
Prelude> :quit
Leaving GHCi.
<timberlake:~:1:28>
```

- (a) (3) Enter your list comprehension expression here in your submission, and submit it as a one-line file named `hw9q1.hs`.
 - (b) (2) Include a transcript of your `ghci` run here.
2. (5) One of the simplest iterative loops to write is one that just prints the numbers from 1 to 10 (inclusive), each on its own line. Write a program in C, C#, C++, Common Lisp, Erlang, Fortran, Haskell, Java, Prolog, Python, or Ruby that prints the numbers from 1 to 10 (inclusive and in increasing order), each on its own line, using recursion. That is, your main program should call a subprogram that does the printing and recursively calls itself.
 - (a) (3) Place your program here, and submit it in a file named `hw9count` with an appropriate extension.
 - (b) (2) Place a transcript of a run of your program here. Include the compile and run steps as appropriate.

Continue on next page.

3. (5) In the Lecture Notes on Statement Level Control Structures, I have a recursive Common Lisp function named `polishPrefixEval` that evaluates an expression presented in Polish Prefix notation. Write a program in C, C#, C++, Common Lisp, Erlang, Fortran, Haskell, Java, Prolog, Python, or Ruby that uses iteration to evaluate an expression presented in Polish Prefix notation. Your main program should call a function named `polishPrefixEval`, which should use any iterative loop to evaluate the expression and return its value. The main program should clearly print the expression evaluated and the value returned by `polishPrefixEval`. Use the 3 examples used in the Lecture Notes, plus two more:
- (a) `+ * 2 3 4` which should evaluate to 10;
 - (b) `* + 2 3 4` which should evaluate to 20;
 - (c) `* 2 + 3 4` which should evaluate to 14;
 - (d) `/ 20 - 8 3` which should evaluate to 4;
 - (e) `* 2 ** 3 4` which should cause some kind of error.

I recommend representing the expression as a list whose successive elements are the elements of the expression, starting with the first. However, you may use any convenient sequential collection for this. You may use any convenient data type to represent the operators, `+`, `-`, `*`, and `/`.

- (a) (3) Place your program here, and submit it in a file named `hw9prefix` with an appropriate extension.
- (b) (2) Place a transcript of a run of your program here. Include the compile and run steps as appropriate.