

Programming Assignment #2

CS 6353 Fall 2015

In this programming assignment, you will be coding in Haskell. The requirements for your submission are listed in sections 1 and 2 below. *Follow the requirements carefully and precisely – they will be thoroughly tested.*

Section 3 provides a few example outputs. Section 4 provides some reference material that you may find useful.

As a reminder, the way I grade will be by running your program against a large test suite. You will be graded by the percentage of test cases you get correct.

The Due Date for this assignment is November 30th at 8:00am.

1. Program Specification

1. Your program must read from standard input, and output to standard output.
2. Input will consist on queries that operate on a sequence of numbers.
3. The sequence shall be defined as follows:
 - a. $F(0) = 1$
 - b. $F(1) = 1$
 - c. $F(2) = 1$
 - d. $F(3) = 1$
 - e. $F(n) = \lfloor (F(n-1)+F(n-2))*F(n-3)/F(n-4) \rfloor$

The first ten numbers of this sequence (starting at $n=0$ and ending with $n=9$) are:

1, 1, 1, 1, 2, 3, 5, 16, 31, 78

4. The user will enter a query on a line by itself, and your program will produce output based on the query, on a line by itself. The queries will be a one-word command (CASE SENSITIVE), which, depending on the command, may be followed by a single argument.
5. The commands are:
 - a. NTH – Print the n th integer in the sequence, starting with the 0th element, on a line by itself.
 - b. SUM – Print the sum of all n integers in the sequence, starting from the 0th element. Print the value on a line by itself.
 - c. BOUNDS – Given a number as an input, print, on two separate lines, the two non-inclusive bounding numbers of the sequence. For instance, if the input is 12, you print 5 and 16.
 - d. QUIT – your program should exit immediately without outputting.
6. After you give output for your command, continue to read additional command, until the QUIT command is issued, or until “end of file” is seen.

7. If an invalid command is given, or if the input is invalid (for instance, asking for the NTH item where n is less than 0) print ERR on a line by itself, and exit the program at that point (previous successful commands should output).
8. Here is the input grammar:

```
INPUT      ::= COMMAND '\n' INPUT | 'QUIT';
COMMAND    ::= NTH_CMD  | SUM_CMD | BOUNDS_CMD;
NTH_CMD     ::= 'NTH' [ \t]+ INTEGER;
SUM_CMD     ::= 'SUM' [ \t]+ INTEGER;
BOUNDS_CMD  ::= 'BOUNDS' [ \t]+ FLOAT;
INTEGER     ::= [0-9]+;
FLOAT       ::= INTEGER | [0-9]+ '.' [0-9]* | '.' [0-9]+;
```
9. White space is important—precisely follow the above grammar.

2. Other Requirements

You will receive a 0 on this if any of these requirements are not met!

10. The program must be written entirely in Haskell.
11. You must submit the source of your program in a single Haskell file, named NETID.hs, with your Net ID in place of NETID.
12. The program must be compilable with GHC—the Glasgow Haskell Compiler – I will not run any other compiler or interpreter.
13. I will compile your code using the command: `ghc NETID.hs` (where NETID is replaced with your NETID).
14. The program must compile and run in the reference environment. Even if it works on your desktop, if it doesn't work in the reference environment, you will get a 0.
15. You must submit your homework before 11:55pm on the due date.
16. You must submit the homework through the course website, unless otherwise pre-approved by the professor.
17. You must submit ONLY the Haskell source file.
18. You may not give or receive any help from other people on this assignment.
19. You may use references on the Internet to teach yourself Haskell.
20. You may NOT use code from any other program, no matter who authored it. This includes 3rd party libraries (you may use anything that comes with GHC by default).

3. Test Cases

Below are four sample test cases for you, which I will use in my testing. Typically, I use anywhere from 20-50 test cases, and will definitely use these three. I strongly recommend you create your own test harness and come up with a large number of test cases to help you get the best possible grade.

For test cases, what one would type on the command line is **BLACK**, input is in **GREEN**, and output is in **BLUE**.

Case 1

./NETID

NTH 9

78

QUIT

Case 2

./NETID

SUM 4

6

BOUNDS 12

5

16

QUIT

Case 3

./NETID

NUM

ERR

Case 4

./NETID

NTH 9

78

BOUNDS

ERR

4. Resources

List of Haskell Tutorials:

<https://www.haskell.org/haskellwiki/Tutorials>

4. Reference Environment

The reference environment is once again kimota.net. Your same login credentials will continue to work.