COMP2212 Programming Language Concepts Coursework - Submission Two

Julian Rathke

Semester 2 2019/20

Second Five Problems

Here are the second set of problems that I would like you to solve in your new Stream Programming Language. The input and output conventions are exactly as before but I repeat them here for convenience.

Input and output conventions

We will model streams using sequences of integers and, in general, inputs will take the following form:

```
a11 a21 a31 a41 ... an1
a12 a22 a32 a42 ... an2
. . .
a1k a2k a3k a4k ... ank
. . .
```

where every line is terminated with the new line character `EOL` and the file ends with `EOF`. The `as` are arbitrary (positive or negative) 32 bit integers (you can safely use a 64 bit implementation, we will not test for architecture specific overflow behaviour). The number n is the number of input sequences, which are the columns in the input. You can assume that all of the input streams will have the same length, or in other words, that all of the columns will be of the same size. This length will be finite in all tests but may be arbitrarily large. Note that empty inputs *are* allowed: if you are expecting n sequences and are given an empty input, you should assume that your input is n empty sequences. The values in each row are separated by a single space character. For example, the following is an input that describes 2 sequences of length 3:

```
2 1
3 −1
4 1
```

For each problem you are expected to produce a single output on `stdout`. Your output is expected be of the form:

```
o1
o2
o3
. . .
oN
```

where the length of the output—that is, the number of lines—will depend on the length of the input. Where the output depends on an input stream that has exhausted its input then no further outputs are required. Every line must be terminated with the Unix new line character `EOL` (i.e. **not** Windows `CR+LF`).

Problem 6 - Two-Three Shuffle

Take two sequences $a_1\ a_2\ a_3\ a_4\ a_5\ \ldots$ and $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ \ldots$ as an input and output the sequence $a_1 \quad a_2 \quad b_1 \quad b_2 \quad b_3 \quad a_3 \quad a_4 \quad b_4 \quad b_5 \quad b_6 \ldots$, that is, the output sequence alternates between input sequences, but reads twice from input one and then three times from input two.

Example input:	Expected output:
	1
	2
1 5	5
2 6	6
3 7	7
4 8	3
	4
	8

Note that input value 8 in the second input stream is consumed as this is the next logical output and is available in the (second) input stream.

Problem 7 - Skip Two then Three

Take a sequence $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ \ldots$ and produce the sequence

$$a_3 \quad a_7 \quad a_{10} \quad a_{14} \quad a_{17} \quad a_{21} \quad \ldots$$

This is the values from the input stream where we skip over two elements, output the next, then skip over three elements, output the next. This is then repeated for the next entries in the input streams.

Example input:	Expected output:
0	
0	
1	
0	
0	
0	
2	1
0	2
0	3
3	4
0	5
0	
0	
4	
0	
0	
5	

Problem 8 - Checksum Differences

Take two sequences $a_1\ a_2\ a_3\ a_4\ a_5\ \ldots$ and $b_1\ b_2\ b_3\ b_4\ b_5\ \ldots$, and produce the sequence

$$a_1 \quad b_1 \quad a_2 \quad b_2 \quad a_3 \quad b_3 \quad a_4 \quad b_4 \quad a_5 \quad b_5 \quad (a_1 - b_1 + a_2 - b_2 + a_3 - b_3 + a_4 - b_4 + a_5 - b_5) \ldots$$

That is a shuffle of the first input stream and the second input stream but after taking 5 entries from each of the input stream, a checksum value consisting of the the sum of the last five entries of the first input stream less the sum of the last five entries of the second input stream.

Example input:	Expected output:
	1
	5
	2
1 5	4
2 4	3
3 3	3
4 2	4
5 1	2
6 0	5
	1
	0
	6
	0

Problem 9 - Counter Padding

Take a sequence $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ \ldots$ and produce the sequence

$$a_1 \quad 1 \quad a_2 \quad 2 \quad a_3 \quad 3 \quad a_4 \quad 4 \quad \ldots a_n \quad n \ldots$$

That is the single input stream is copied to output but in between each entry is a incrementing counter value.

Example input:	Expected output:
	23
	1
	12
23	2
12	53
53	3
90	90
27	4
4	27
	5
	4
	6

Problem 10 - Fibonacci Sequences

Take a sequence $a_1\ a_2\ a_3\ a_4\ a_5 \ldots$ and output the sequence

$$a_1 \quad a_1 + a_2 \quad 2a_1 + a_2 + a_3 \quad 3a_1 + 2a_2 + a_3 + a_4 \quad 5a_1 + 3a_2 + 2a_3 + a_4 + a_5 \ldots$$

where the coefficients of each input term in the sums follows the Fibonacci series 1 1 2 3 5 8 ... from when it first appears. Recall that the Fibonacci series starts with two 1s and then the subsequent terms are always the sum of the previous two.

Example input:	Expected output:
1	1
0	1
0	2
0	3
0	5

Example input:	Expected output:
1	1
2	3
3	7
4	14
5	26

Second submission - due Thursday May 14th 4pm

What to submit. You will need to submit a single zip file containing:

First, provide the source code for five programs named (`pr6.spl`, `pr7.spl`, `pr8.spl`, `pr9.spl`, `pr10.spl`) written in your language that solve the additional problems. We will run our tests on your solutions and award marks for solving the additional problems correctly.

Second, a 3 page report on your language **in pdf format** named 'report.pdf' that explains the main language features, its syntax, including any scoping and lexical rules as well as additional features such as syntax sugar for programmer convenience, type checking, informative error messages, etc. In addition, the report should explain the execution model for the interpreter, e.g. what the states of the runtime are comprised of and how they are transformed during execution. This report, together with the five programs will be evaluated qualitatively and your marks will be awarded for the elegance and flexibility of your solution and the clarity of the report.

Third, a text file called 'group.txt' that contains, the usernames of all students in your pair or group, as well as an agreed declaration of how marks are to be distributed amongst the members of your group. For example, 50-50, 60-40 etc. If such an agreed declaration cannot be made then please state this in your declaration instead and arrange for each member of the group to submit separately. Otherwise, please make sure that there is only one submission per group.

How to submit. All submissions must be done through handin.

Marks. This coursework counts for 40% of the total assessment for COMP2212. There are a total of 40 marks available. These are distributed between the two submissions as follows:

Submission one has 10 marks available. There are 2 marks available for functional correctness of each of the first five problems. You will receive the results of Submission 1 prior to the second deadline.

Submission two has 30 marks available. We will award up to 10 marks for the qualitative aspects of your solution, as described in your programming language report. We will award up to 20 marks for your solutions to the second five problems. For each problem there will be 4 marks available for functional correctness only. You have the option of resubmitting the interpreter, for a 50% penalty on this component. If you decide to resubmit your interpreter in the second submission the maximum possible total coursework mark is therefore capped at 30 marks.

Any late submission to either component will be treated as a late submission overall and will be subject to the standard university penalty of 10% per working day late.