2014 FRACTRAN

FRACTRAN is an esoteric programming language invented by John Conway. It is so called because a FRACTRAN program consists of a list of fractions, such as:

$$\left\{ \frac{17}{91}, \frac{78}{85}, \frac{19}{51}, \frac{23}{38}, \frac{29}{33}, \frac{77}{29}, \frac{95}{23}, \frac{77}{19}, \frac{1}{17}, \frac{11}{13}, \frac{13}{11}, \frac{15}{14}, \frac{15}{2}, \frac{55}{1} \right\}$$

This incredible sequence will output only *prime* numbers — indeed, if left for long enough, it will generate *all* the prime numbers in increasing order!

To run a FRACTRAN program consisting of fractions $\left\{ \frac{a_1}{b_1}, \ldots, \frac{a_k}{b_k} \right\}$ the following procedure is applied:

(1) The input is a positive integer A. Set X := A;
(2) Find the lowest *i* for which $X \times \frac{a_i}{b_i}$ is an integer. Set Y := $X \times \frac{a_i}{b_i}$;
(3) If Y = 2^n (where $n \geq 0$) then halt and output the integer *n;*
(4) Set X := Y and return to step (2).

If we want our program to keep producing output — such as for our prime number program — we remove the "halt" instruction from step (3). However, in everything that follows we will not consider this.

Question 1

Step (2) might fail on some lists of fractions. What condition can we impose on b_k to make sure that it can always be completed?

$$\left\{ \frac{2}{3}, \frac{3}{1} \right\}$$
program 1

For example, consider *program 1* being run on the input *4*. On the first iteration, 4 × 2/3 = 8/3 is not an integer, but 4 × 3/1 = 12 is, so Y := 12. As this is not a power of 2 we set X := 12 and repeat. This time 12 × 2/3 = 8 = 2^3 is an integer and a power of 2, so the program halts and outputs 3.

Question 2

$$\left\{ \frac{14}{13}, \frac{8}{7}, \frac{91}{1} \right\}$$
program 2

Run *program 2* on the input *8*, *briefly* outlining the steps of the computation. You may use any convenient format (e.g. a table).

It is helpful when running a FRACTRAN program to consider the *prime factorization* of the fractions, rather than their raw numerical values. For example $\frac{2^{11} \times 7^3 \times 11}{5^5 \times 101}$ is preferable to $\frac{7727104}{315625}$. The same goes for values such as A, X and Y. Fortunately, all integers have unique prime factorizations.

Question 3

Suppose you wrote a FRACTRAN interpreter; i.e. code to execute FRACTRAN programs. Explain how using prime factorizations will lead to a better implementation than keeping raw numerical values.

Question 4

Rewrite *program 2* in factorized form.

For convenience for the remaining questions, if a program *receives a* we mean *a* is a non-negative integer and the input A (in step (1)) will be 2^a.
If it *receives a and b* (non-negative integers), the input will be 2^a3^b.

It is not too hard to write FRACTRAN programs that simulate basic arithmetic operations. For instance, if *program 1* receives *a*, the program runs as:

$$2^a \rightarrow 3\times2^a \rightarrow 2^{a+1}$$

and the output is *a+1*. So *program 1* is a program that adds one to a number — though not the simplest such program!

<u>Question 5</u>
What is the simplest (i.e. shortest) program that adds one to a number?

$$\left\{\frac{2}{3},\frac{1}{1}\right\}$$

program 3

<u>Question 6</u>
Consider *program 3*. If this program receives *a* and *b*, what is the output?

$$\left\{\frac{1}{2\times3},\frac{1}{3},\frac{1}{1}\right\}$$

program 4

<u>Question 7</u>
Now consider *program 4*. If this program receives *a* and *b*, what is the output? Ensure you cover all the cases.

$$\left\{\frac{2^2}{5},\frac{1}{7},\frac{5}{2},\frac{7}{1}\right\}$$

program 5

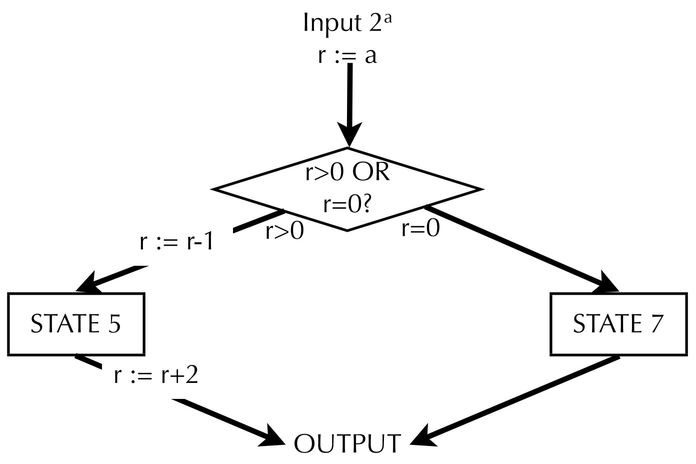
Sometimes it is necessary to use one or more prime factors to affect the control flow of the program. For example, consider *program 5* when it receives *a*. If *a=0* then the run goes:

$$1 \rightarrow 7 \rightarrow 1\ (=2^0)$$

and the output is *0*, but if *a>0* then we get:

$$2^a \rightarrow 5\times2^{a-1} \rightarrow 2^{a+1}$$

and so the program outputs *a+1*. Here *5* and *7* are just used to encode *state* rather than numerical values. The same kind of idea can be used in more complicated ways.



Note that we are using our internal value *X* to keep track simultaneously of our data and our state. The above diagram is an interpretation of this: *r* (the data) is the number of powers of 2 in *X* and we are in state *5* (or *7*) when *X* is divisible by 5 (or 7).

Question 8

Describe the output of *program 6* when it receives *a*. Briefly justify your answer using a diagram.

$$\left\{ \frac{2\times7}{5}, \frac{5}{2^2\times7}, \frac{1}{7}, \frac{7}{1} \right\}$$

program 6

It is also possible to *combine* two programs — with care! One approach is to build subroutines, i.e. groupings of fractions that perform specific tasks. We will use states to distinguish which subroutine we are currently executing.

$$\left\{ \frac{2\times7}{3\times7}, \frac{1}{7}, \frac{2\times7}{5}, \frac{5}{1} \right\}$$

pro'ram 7

Pro'ram 7 is not quite a program which receives *a* and *b*, and outputs *a+b+1*. It was designed to: enter state 5; add 1 to *a* whilst moving from state 5 to state 7; add *b* to *a* by repeatedly adding 1 to *a* whilst decreasing *b* by 1 until *b* is zero; finally exiting state 7 to output *a+b+1*.

This program does not output *a+b+1* because the first fraction is actually 2/3. This fraction will not be used only when the state is 7, neither is it guaranteed that the state will be 7 after use.

Question 9

What will *pro'ram 7* output when it receives *a* and *b*?

Question 10

Modify *pro'ram 7* to use a temporary state 11, along with state 7, to calculate *a+b+1*.

Question 11

Write a FRACTRAN program which receives *a* and outputs 2*a*. Briefly explain your program.

$$\left\{ \frac{23}{2\times13}, \frac{17}{13},\quad \frac{11\times29}{3\times23}, \frac{7\times11\times29}{5\times23}, \frac{23}{29}, \frac{31}{23},\quad \frac{3\times37}{7\times31}, \frac{5\times37}{11\times31}, \frac{31}{37}, \frac{13}{31},\quad \frac{19}{5\times17}, \frac{2\times19}{3\times17}, \frac{17}{19}, \frac{1}{17},\quad \frac{3\times5\times13}{1} \right\}$$

program 8

Program 8 receives *a*. It consists of five subroutines as indicated by the spacing. The first subroutine to be executed initializes two pieces of data (stored using powers of 3 and 5) and switches to state 13.

Question 12

When it receives 10 the program will take nearly 2500 steps before output is produced. What is that output?

It is recommended that you utilize diagrams, rather than performing the steps.