CS1010E 16/17S1 Farey Sequence (Question)

Tags & Categories

Tags:

Categories:

Related Tutorials

Task Content

Farey Sequence

Topic Coverage

- Nested control statements
- Assignment and expressions

You may use other programming constructs, but **do not** use the math library functions.

Problem Description

A Farey sequence of order n, denoted F_n, is the sequence of reduced fractions between 0 and 1 which have denominators less than or equal to n, typically arranged in order of increasing size.

The example below illustrates the generation of fractions of the sequence F_4.

Denominator	Numerator	Fraction	Reduced?
1	0	0/1	Yes
1	1	1/1	Yes
2	0	0/2	No (can be reduced to 0/1)
2	1	1/2	Yes
2	2	2/2	No (can be reduced to 1/1)
3	0	0/3	No (can be reduced to 0/1)
3	1	1/3	Yes
3	2	2/3	Yes
3	3	3/3	No (can be reduced to 1/1)
4	0	0/4	No (can be reduced to 0/1)
4	1	1/4	Yes
4	2	2/4	No (can be reduced to 1/2)
4	3	3/4	Yes
4	4	4/4	No (can be reduced to 1/1)

Hence the sequence of reduced fractions of F_4 is

```
0/1,1/1,1/2,1/3,2/3,1/4,3/4
```

In this task, you are to generate the Farey sequence of a given input order n (`> 0`).

This task is divided into several levels. Read through all the levels (from first to last, then from last to first) to see how the different levels are related. **You may start from any level.**

- **Deadline: Submit your work to CodeCrunch by Thursday, 29 September, 23:59:59.**

Level 1

Name your program `farey1.c`

Write a program that reads in an integer order `n (> 0)`, and outputs the denominator values from `1` to `n`.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1
1
```

```
$ ./a.out
4
1
2
3
4
```

```
$ ./a.out
8
1
2
3
4
5
6
7
8
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < farey.in | diff - farey1.out
```

To proceed to the next level (say level 2), copy your program by typing the Unix command

```
cp farey1.c farey2.c
```

Level 2

Name your program `farey2.c`

Write a program that reads in an integer order `n (> 0)`, and outputs the denominator values from `1` to `n`. For each denominator value, output all possible numerator values such that the fraction spans from `0` to `1`.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1
0/1
1/1
```

```
$ ./a.out
4
0/1
1/1
0/2
1/2
2/2
0/3
1/3
2/3
3/3
0/4
1/4
2/4
3/4
4/4
```

```
$ ./a.out
8
0/1
1/1
0/2
1/2
2/2
0/3
1/3
2/3
3/3
0/4
1/4
2/4
3/4
4/4
0/5
1/5
2/5
3/5
4/5
5/5
0/6
1/6
2/6
3/6
4/6
5/6
6/6
0/7
1/7
2/7
3/7
4/7
5/7
6/7
7/7
0/8
1/8
2/8
3/8
4/8
5/8
6/8
7/8
8/8
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < farey.in | diff - farey2.out
```

To proceed to the next level (say level 3), copy your program by typing the Unix command

```
cp farey2.c farey3.c
```

Level 3

Name your program `farey3.c`

A fraction p/q is irreducible if and only if p (≥ 0) and q (> 0) have no common divisor other than 1.

Write a program that reads in an integer order `n (> 0)`, and outputs the possible fractions spanning from `0` to `1`. Alongside each fraction, output `YES` if the fraction is **irreducible**, or `NO` otherwise.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1
0/1,YES
1/1,YES
```

```
$ ./a.out
4
0/1,YES
1/1,YES
0/2,NO
1/2,YES
2/2,NO
0/3,NO
1/3,YES
2/3,YES
3/3,NO
0/4,NO
1/4,YES
2/4,NO
3/4,YES
4/4,NO
```

```
$ ./a.out
8
0/1,YES
1/1,YES
0/2,NO
1/2,YES
2/2,NO
0/3,NO
1/3,YES
2/3,YES
3/3,NO
0/4,NO
1/4,YES
2/4,NO
3/4,YES
4/4,NO
0/5,NO
1/5,YES
2/5,YES
3/5,YES
4/5,YES
5/5,NO
0/6,NO
1/6,YES
2/6,NO
3/6,NO
4/6,NO
5/6,YES
6/6,NO
0/7,NO
1/7,YES
2/7,YES
3/7,YES
4/7,YES
5/7,YES
6/7,YES
7/7,NO
0/8,NO
1/8,YES
2/8,NO
3/8,YES
4/8,NO
5/8,YES
6/8,NO
7/8,YES
8/8,NO
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < farey.in | diff - farey3.out
```

To proceed to the next level (say level 4), copy your program by typing the Unix command

```
cp farey3.c farey4.c
```

Level 4

Name your program `farey4.c`

Write a program that reads in an integer order `n (> 0)`, and outputs the corresponding Farey sequence. Each fraction of the Farey sequence is output on a separate line.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1
0/1
1/1
```

```
$ ./a.out
4
0/1
1/1
1/2
1/3
2/3
1/4
3/4
```

```
$ ./a.out
8
0/1
```

```
1/1
1/2
1/3
2/3
1/4
3/4
1/5
2/5
3/5
4/5
1/6
5/6
1/7
2/7
3/7
4/7
5/7
6/7
1/8
3/8
5/8
7/8
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < farey.in | diff - farey4.out
```

To proceed to the next level (say level 5), copy your program by typing the Unix command

```
cp farey4.c farey5.c
```

Level 5

Name your program `farey5.c`

Write a program that reads in an integer order `n (> 0)`, and outputs the corresponding Farey sequence with fractions of the same denominator comma-separated on each line. There should be no blank spaces in the output.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1
0/1,1/1
```

```
$ ./a.out
4
0/1,1/1
1/2
1/3,2/3
1/4,3/4
```

```
$ ./a.out
8
0/1,1/1
1/2
1/3,2/3
1/4,3/4
1/5,2/5,3/5,4/5
1/6,5/6
1/7,2/7,3/7,4/7,5/7,6/7
1/8,3/8,5/8,7/8
```

Check the correctness of the output by typing the following Unix command

```
./a.out < farey.in | diff - farey5.out
```

Click here to submit to CodeCrunch.

Do not use the Submit button below until the deadline is over.

Submission (Course)

Select course: CS1010E (2016/2017 Sem 1) - Programming Methodology ▼

Your Files:

SUBMIT (only .java, .c, .cpp and .h extensions allowed)

To submit multiple files, click on the Browse button, then select one or more files. The selected file(s) will be added to the upload queue. You can repeat this step to add more files. Check that you have all the files needed for your submission. Then click on the Submit button to upload your submission.