Exercises — Seq

version **#f6de30ca5ee18fb9a30e5fbfc9eb9a1de8e13354**

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.assistants.epita.fr

File Tree

```
seq/
└── seq.sh  (to submit)
```

Authorized commands : You are only allowed to use the following commands

- builtins

Reminder : Grant execution permission to your scripts before pushing them

1 Goal

You have to write a shell script allowing to generate ranges of numbers. The expected usage is:

```
42sh$ ./seq.sh FIRST INCREMENT LAST
```

If the number of arguments is not correct, the script has to return `1` and display the usage on the standard error output (see examples).

The argument `INCREMENT` must not be zero. If this is the case, the script must not display anything and return 1.

The script must behave as follow:

- If `FIRST` = `LAST`, you must display `FIRST`.
- If `FIRST` < `LAST`, you must display the numbers n in ascending order, such that:
  - `FIRST` $\leq n \leq$ `LAST`
  - $n =$ `FIRST` $+ i \times$ `INCREMENT` $(\forall i \in \mathbb{N}, i \geq 0)$

  In that case, the `INCREMENT` must be strictly positive. Otherwise, your script must not display anything and return 1.
- If `FIRST` > `LAST`, you must display the number n in descending order, such that:
  - `LAST` $\leq n \leq$ `FIRST`
  - $n =$ `FIRST` $+ i \times$ `INCREMENT` $(\forall i \in \mathbb{N}, i \geq 0)$

  In that case, the `INCREMENT` must be strictly negative. Otherwise, your script must not display anything and return 1.

Your script must return 0 on success.

2 Examples

```
42sh$ ./seq.sh 1 -1 1
1
42sh$ ./seq.sh 42 -1 42
42
42sh$ ./seq.sh 42 -2 40
42
40
42sh$ ./seq.sh 10 3 23
10
13
16
19
22
42sh$ ./seq.sh 42
Usage: ./seq.sh FIRST INCREMENT LAST
42sh$ echo "$?"
1
42sh$ ./seq.sh 3 2 1
42sh$ echo "$?"
1
```

The way is lit. The path is clear. We require only the strength to follow it.