Exercises — Hanoi

version #

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

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

Contents

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

1 Hanoi

Files to submit:

- hanoi/hanoi.c
- hanoi/hanoi.h

Main function: none

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

- putchar(3)

Authorized headers: You are only allowed to use the functions defined in the following headers:

- stddef.h
- assert.h
- errno.h
- err.h

1.1 Goal

The goal here is to implement a function capable of solving an instance of the tower of Hanoï problem. This function must have the following prototype:

```
void hanoi(unsigned n);
```

In the tower of Hanoï problem, you have three 'towers' on which differently sized rings are stacked on one another. A ring can only be on a larger one (or on the base if it is the first). The problem is solved when all the rings have been moved from the first to the last tower while respecting the previous rule and only moving the rings once at a time.

Your `hanoi` function must display the movements needed to solve the problem. You must display the solution with the least movements.

The `n` parameter of your `hanoi` is the number of rings on the first tower at the beginning. Its output is the sequence of movements needed to solve the problem in the form *X*->*Y* where *X* is the number of the origin tower and *Y* the number of the destination one.

For instance, `hanoi(3)` prints:

```
1->3
1->2
3->2
1->3
2->1
2->3
1->3
```

1.2 Example

To help you visualize, here are the steps of the algorithm for `hanoi(2)`:

- Initial state:

```
 -
---
 1     2     3
```

- Move 1->2:

```
---    -
 1     2     3
```

- Move 1->3:

```
      -    ---
1     2     3
```

- Move 2->3:

```
           -
          ---
1     2     3
```

It is my job to make sure you do yours.