BDE 42
ANGOULEME

CONTEST

PUSH_SWAP

Common Instructions

Norm

- Your project must be written in C
- Your project must be written in accordance with the norm.
- Global variables are forbidden.

Valgrind

- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc.) except for undefined behavior.
- All heap-allocated memory must be properly freed when necessary. Memory leaks will not be tolerated.

Makefile

- Your Makefile must contain at least the rules $(NAME), all, clean, fclean and re.
- The Makefile that compiles your source files to the required output with the flags -Wall, -Wextra, and -Werror, using cc. Additionally, your Makefile must not perform unnecessary relinking.

Other

- If your project use your libft, you must copy its sources and its associated Makefile into a libft folder. Your project's Makefile must compile the library by using its Makefile, then compile the project.
- Submit your work to the assigned Git repository. Only the work in the Git repository will count.

The project

Program Name	`./{{your_login}}`
Files to Submit	`*/*.h */*.c`
Makefile	`NAME, all, clean, fclean, re`
Arguments	**Stack a :** A list of integers

External Function	`- read`
	`- write`
	`- malloc`
	`- free`
	`- exit`
	`- ft_printf (yours)`

Libft authorized	**Yes**
Description	**Sort stacks**

About the project

You have to write a program named (login) that takes as an argument the stack a formatted as a list of integers. The first argument should be at the top of the stack (be careful about the order). You have 2 stacks named a and b. At the beginning, the stack a contains a random number of unique negative and/or positive integers and the stack b is empty.

The program must display the shortest sequence of instructions needed to sort stack a with the smallest number at the top. The instructions must be separated by a ’\n’ and nothing else.
The goal is to sort the stack with the lowest possible number of operations. If no parameters are specified, the program must not display anything and should return to the prompt.

In case of error, it must display "Error" followed by an ’\n’ on the standard error. Errors include, for example: some arguments not being integers, some arguments exceeding the integer limits, and/or the presence of duplicates.

Available movement

`sa` **Swap the first 2 elements at the top of stack a.** *Do nothing if there is only one element or none.*

`sb` **Swap the first 2 elements at the top of stack b.** *Do nothing if there is only one element or none.*

`ss` **Use sa and sb at the same time.**

`pa` **Take the first element at the top of b and put it at the top of a.** *Do nothing if b is empty.*

`pb` **Take the first element at the top of b and put it at the top of b.** *Do nothing if b is empty.*

`ra` **Shift up all elements of stack a by 1.** *The first element becomes the last one.*

`rb` **Shift up all elements of stack b by 1.** *The first element becomes the last one.*

`rr` **Use `ra` and `rb` at the same time.**

`rra` **Shift down all elements of stack a by 1.** *The last element becomes the first one.*

`rrb` **Shift down all elements of stack a by 1.** *The last element becomes the first one.*

`rrr` **Use `rra` and `rrb` at the same time.**

Levels

In this competition, there are three different levels. Each one wins an intra title.

100

To participate at this level, **you must be part of the 2025 promo**. Your goal is to make as few moves as possible. We consider a timeout after 60 seconds.

500

To participate at this level, **you must be part of the 2025 promo**. Your goal is to make as few moves as possible. We consider a timeout after 60 seconds.

1000

This level is available for all promotions. Your goal is to make as few moves as possible. We consider a timeout after 3 minutes.

Submission

Submit your assignment in a branch from this Git repository. The branch must be named by your login. When it’s done, you can make a pull request, it will be merged by an admin. Don’t hesitate to double check the names of your files to ensure they are correct.

Feel free to organize your files as you wish as long as you turn in the mandatory files and comply with the requirements.

Don’t forget norm and memory leaks !

If something is not clear, you can take a reference the original push_swap subject or contact amblanch or rgodet on Discord.

You have until February 15th at 9:42 PM to submit the project.

Good luck :)