Building the thing

Push_swap is not a very difficult project in terms of code construction. You will see that it doesn't need many steps to finish it. The hardest part is to find a solution to the problem and to to construct its algorithm.

As usual, in this page I will make checklists for you to succeed push swap at best

Main Checklist

- [x] Decide if you are going to do your project using chained lists or using tables
- [x] Handle the arguments you are given in input
- [x] (If you choose radix sort, normalize your data !)

Error Checklist

There are a lot of errors to handle in push_swap. Don't forget any of them!

- [x] **The program must work with several numerical arguments**
    - [x] ./push_swap 1 3 5 +9 20 -4 50 60 04 08
- [x] **The program also works if you receive a single character list as a parameter**
    - [x] ./push_swap "3 4 6 8 9 74 -56 +495"
- [x] **The program should NOT work if it encounters another character**
    - [x] ./push_swap 1 3 dog 35 80 -3
    - [x] ./push_swap a
    - [x] ./push_swap 1 2 3 5 67b778 947
    - [x] .push_swap " 12 4 6 8 54fhd 4354"
    - [x] ./push_swap 1 --   45 32
        - [x] these examples should return "Error\n"
- [x] **The program should NOT work if it encounters a double number**
    - [x] ./push_swap 1 3 58 9 3
    - [x] ./push_swap 3 03
    - [x] ./push_swap " 49 128   50 38   49"
        - [x] these examples should return "Error\n"
    - [x] ./push_swap "95 99 -9 10 9"
        - [x] this example should work because -9 & 9 are not equal
- [x] **The program should work with INT MAX & INT MIN**
    - [x] ./push_swap 2147483647 2 4 7
    - [x] ./push_swap 99 -2147483648 23 545
    - [x] ./push_swap "2147483647 843 56544 24394"
        - [x] these examples should work and sort your list
    - [x] ./push_swap 54867543867438 3
    - [x] ./push_swap -2147483647765 4 5
    - [x] ./push_swap "214748364748385 28 47 29"
        - [x] these examples should return "Error\n"
- [x] **Nothing has been specified when strings and int are mixed.** It's up to you what you want to do
    - [x] ./push_swap "1 2 4 3" 76 90 "348 05

Instructions Checklist

This checklist will help you to verify that you have coded all the instructions (11) to use them in your algorithms (next step). If you don't understand what these functions do you can check the main page of push_swap where you will be given a better explanation

- [x] **sa (swap a)**: If there are 2 numbers, swap the first 2 elements at the top of the stack a.
- [x] **sb (swap b)** : If there are 2 numbers, swap the first 2 elements at the top of the stack b.
- [x] **ss** : sa and sb at the same time.
- [x] **pa (push a)**: If b is not empty it takes the first element on top of b and puts it on a.
- [x] **pb (push b)**: If a is not empty, it takes the first element on top of a and puts it on b.
- [x] **ra (rotate a)**: Shifts all the elements of the stack a up by one position. The first element becomes the last.
- [x] **rb (rotate b)** : Shifts all the elements of the stack b one position upwards. The first element becomes the last one.
- [x] **rr** : ra and rb at the same time.
- [x] **rra (reverse rotate a)**: Shifts all elements of the stack down one position. the stack a. The last element becomes the first.
- [x] **rrb (reverse rotate b)**: Shifts all the elements of the stack b one position downwards. the stack b. The last element becomes the first.
- [x] **rrr** : rra and rrb at the same time

Algorithm/Sort Checklist

- [x] Check if the order of the list is correct or if it should be sorted
- [x] Make a small algorithm for small numbers (5 and less)
    - [x] Implement a condition for 2 numbers
    - [x] Create an algorithm for 3 numbers
    - [x] Create an algorithm for 4 numbers
    - [x] Create an algorithm for 5 numbers
- [x] Create another algorithm for all other numbers

And obviously at the end check if everything is correct ! **No leaks should be present**, remember to always free the allocated memory when you are not using it anymore.

And that's it... Good luck with the code! If you have any other questions don't hesitate to contact me (Laura), and I'll be happy to help you :)

Previous
Algorithms

Next
FdF

Last updated 14 days ago