



Programmation élémentaire

push_swap

Responsable Astek astek_resp@epitech.eu



Table des matières

Détails administratifs	2
Description du jeu	3
Exemples	4
Le programme	5
Options	6
Fonctions autorisées	7



Détails administratifs

- Rendu : Les sources doivent être rendues sur le dépôt `SVN`, le `Makefile` devra être à la racine du dépôt.



Attention la norme sera passée sur tous les fichiers que vous rendez avec votre lib my



Description du jeu

Le jeu est constitué de 2 listes nommées `l_a` et `l_b`. Au départ `l_b` est vide et `l_a` contient un certain nombre de nombres positifs ou négatifs (sans doublons). Le but du jeu est de faire en sorte que `l_a` contienne les mêmes nombres mais dans l'ordre croissant. Pour ce faire, on ne dispose que des opérations suivantes :

- **sa** : swap les 2 premiers éléments de `l_a`
(ne fait rien s'il n'y en a qu'un ou aucun).
- **sb** : swap les 2 premiers éléments de `l_b`
(ne fait rien s'il n'y en a qu'un ou aucun).
- **ss** : sa et sb en même temps.
- **pa** : prend le premier élément de `l_b` et le met en premier dans `l_a`.
(si `l_b` est vide ne fait rien).
- **pb** : prend le premier élément de `l_a` et le met en premier dans `l_b`.
(si `l_a` est vide ne fait rien).
- **ra** : rotate `l_a`
(vers le début, le premier élément devient le dernier).
- **rb** : rotate `l_b`
(vers le debut, le premier élément devient le dernier).
- **rr** : ra et rb en meme temps.
- **rra** : rotate `l_a`
(vers la fin, le dernier élément devient le premier).
- **rrb** : rotate `l_b`
(vers la fin, le dernier élément devient le premier).
- **rrr** : rra et rrb en même temps.



Exemples

- La liste a et b sera défini ainsi :
l_a 2 1 3 6 5 8
l_b
- sa
l_a 1 2 3 6 5 8
l_b
- pb pb pb
l_a 6 5 8
l_b 3 2 1
- ra rb (on peut donc aussi dire rr)
l_a 5 8 6
l_b 2 1 3
- rra rrb (on peut donc aussi dire rrr)
l_a 6 5 8
l_b 3 2 1
- sa
l_a 5 6 8
l_b 3 2 1
- pa pa pa
l_a 1 2 3 5 6 8
l_b



Le programme

Vous devez faire un programme qui prend en paramètre la liste `l_a` sous la forme d'une liste de paramètres (Pas de doublons, tout les nombres sont bons et rentrent dans un entier). Le programme doit afficher la suite d'opérations qui permet de trier la liste. Les operations seront affichées séparées par un espace, pas d'espace au debut ni à la fin, le tout suivi d'un `'\n'`. Le but est de trier la liste avec le moins d'opérations possibles.

```
1 $ ./push_swap 2 1 3 6 5 8
2 sa pb pb pb sa pa pa pa
3 $
```



Options

On pourra faire les options :

- -v qui affiche les états de l_a et l_b à chaque étape.
- -vt la même chose en termcaps



Fonctions autorisées

- write
- malloc
- free
- exit