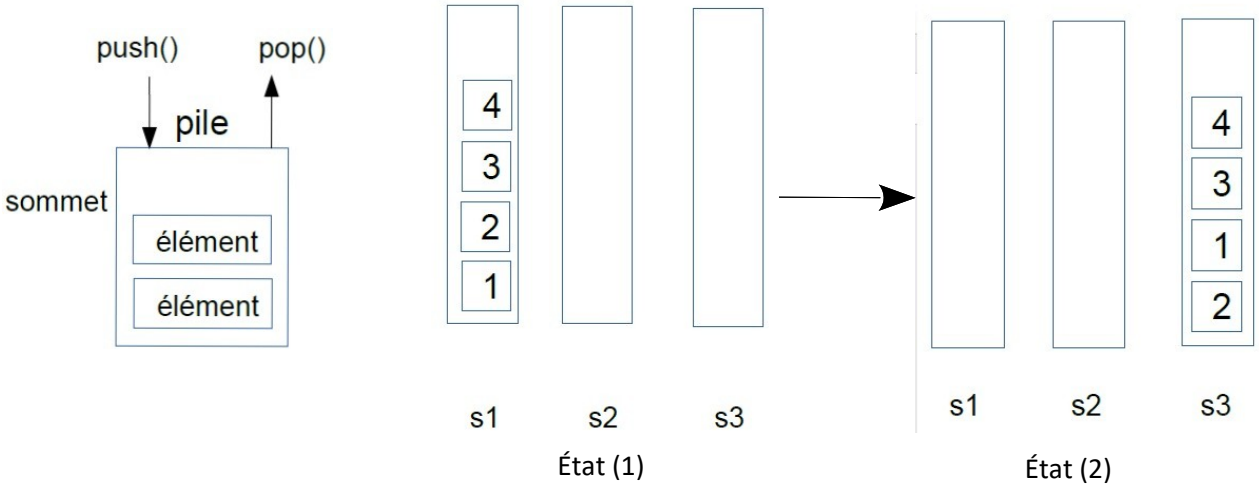
TP Piles -Files

Partie 1 – Piles

a) En considérant que les piles s1, s2, s3 sont initialement dans l'état (1) utilisez les prototypes de fonctions vus en cours pour que les 3 piles soient dans l'état final (2). Faites l'exercice en papier. Bien evidemment faire pop(s1), pop(s1),pop(s1), pop(s1),push(s3,2),push(s3,1),push(s3,3),push(s3,4) n'est pas accepté ! Il faut utiliser s2 comme pile intermediaire.



b) Codez les fonctions des piles vues en cours (et faites les tests) avec les structures internes suivantes :
Ne pas oublier de créer fichiers séparés, c.-à-d. , main.c, stack.c, stack.h.

Structure 1 (avec une liste chaînée)

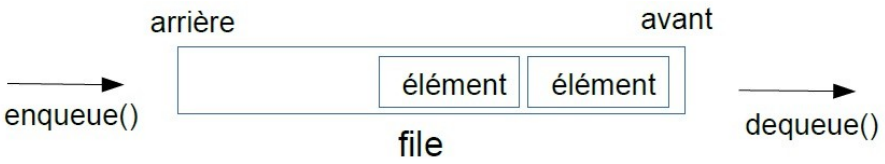
```
struct Stack {
        struct List* liste;
};
```

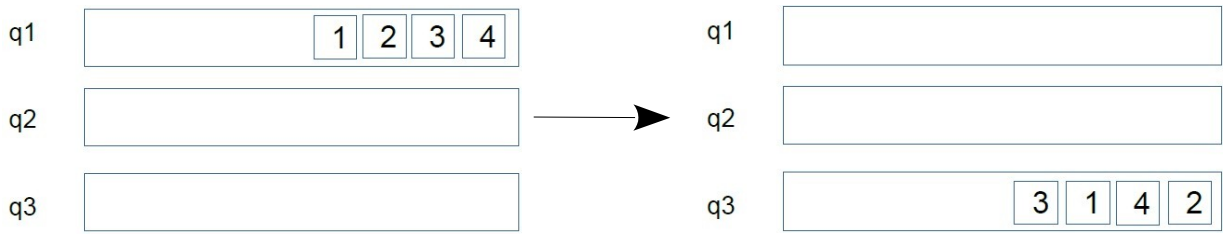
Structure 2 (avec un tableau statique)

```
#define MAX_SIZE 30;
struct Stack {
        int size;
        int top;
        int elems[MAX_SIZE];
};
```

Partie 2 – Files

a) En considérant que les files q1, q2, q3 sont initialement dans l'état (1) utilisez les prototypes de fonctions vus en cours pour que les 3 files soient dans l'état final (2). Faites l'exercice en papier.
Bien evidemment faire, dequeue(q1),dequeue(q1),dequeue(q1),dequeue(q1), enqueue(q3,2),enqueue(q3,4),enqueue(q3,1),enqueue(q3,3) n'est pas accepté !
Il faut utiliser q2 comme file intermediaire.





État (1) État (2)

b) Codez les fonctions des files vues en cours (et faites les tests) avec les structures internes suivantes :
Ne pas oublier de créer fichiers séparés, c.-à-d., main.c, queue.c, queue.h.

Structure 1 (avec une liste chaînée)

```
struct Queue {
        struct List* liste;
} ;
```

Structure 2 (avec un tableau statique)

```
#define MAX_SIZE 30;
struct Queue {
        int size;
        int front;
        int rear;
        int elems[MAX_SIZE];
} ;
```