IN 261 - Pré-formation C 7

ENSTA ParisTech

1 Piles

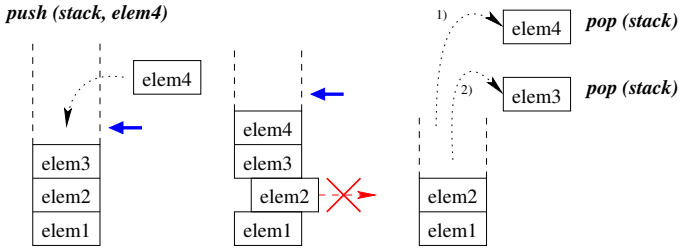
Une pile est une structure de données permettant de stocker des éléments d'un même type.

Une pile d'entiers permettra de stocker des entiers, une pile de « **struct** blah_t » permettra de stocker des « **struct** blah_t ».

Une pile est une structure « dernier entré-premier sorti ».

`push` : on empile (rajoute) une information sur le sommet de la pile.

`pop` : on dépile (retire) l'information se trouvant au sommet de la pile.



En C on représente l'espace de stockage d'une pile par un tableau.

Pour savoir où empiler et d'où dépiler, on mémorise un « pointeur de pile » indique la prochaine place libre dans le tableau.

Le « pointeur de pile » n'est pas un pointeur C : c'est un entier positif représentant un indice dans le tableau de la pile.

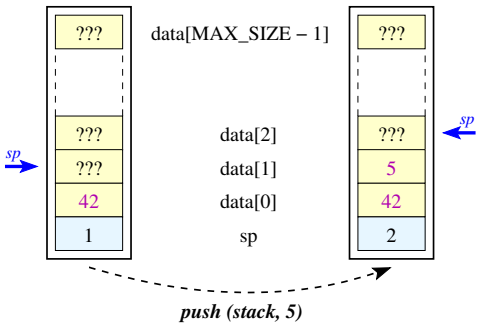
En C, une pile de taille fixe (64) contenant des éléments de type « `un_type` » est représentée par une structure :

```
#define MAX_SIZE 64
struct stack {
  unsigned int sp ;       /* Pointeur de pile. */
  un_type data[MAX_SIZE] ;   /* Zone de stockage de la pile. */
};
```

Initialement, une pile est vide (ne contient pas d'éléments) : le « pointeur de pile » est à 0.

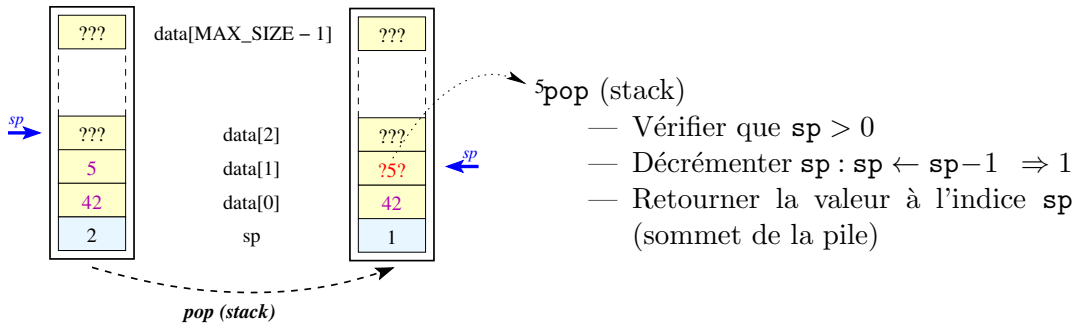
Essayer de dépiler dans une pile vide (« pointeur de pile » == 0) provoque une erreur.

Essayer d'empiler dans une pile pleine (« pointeur de pile » == `MAX_SIZE`) provoque une erreur.



`push` (stack, 5) :

- Vérifier que `sp` $< MAX\_SIZE$
- 5 mis à l'indice `sp` $\Rightarrow$ 1
- Incrémenter `sp` : `sp` $\leftarrow$ `sp`$+1$ $\Rightarrow$ 2



On peut implanter une pile de taille non fixe en allouant dynamiquement (avec `malloc`) le tableau et en le ré-allouant (nouvelle allocation, recopie, libération de l'ancien tableau) lorsque la pile est pleine.