

# Pratique du C :

## Pointeurs : structures autoréférentes

Licence Informatique — Université Lille  
Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 4 — 2016-2017

### Exercice 1 — Copie d'une chaîne de caractères.

Donnez la définition de la fonction de prototype :

```
char * strdup(const char *) ;
```

qui prend en paramètre une chaîne de caractères et retourne un pointeur sur une copie de cette chaîne.

Vous devrez utiliser la fonction :

```
#include <stdlib.h>
void * malloc(int) ;
```

1  
2

Vous devez prendre en compte la gestion d'erreur.

### Exercice 2 — Gestion de pile.

On se propose de donner les définitions des fonctions de prototypes :

```
int is_empty(pile_t) ;
void init(pile_t *) ;
int push(pile_t *, int) ;
int pop(pile_t *, int *val) ;
void destroy(pile_t *) ;
```

Si besoin est, ces fonctions retournent NULL en cas d'erreur.

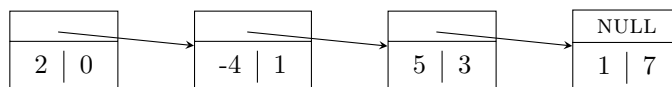
La taille de la pile n'étant pas bornée a priori, on utilisera l'allocation dynamique de mémoire.

### Questions.

1. Donnez la définition d'un type `pile_t`.
2. Comment est représentée la pile vide ?
3. Expliquez les paramètres des prototypes ci-dessus.
4. Donnez la définition de ces fonctions.

### Exercice 3 — Représentation et manipulation de polynômes.

Un polynôme peut être vu comme une liste de monômes. Un monôme est constitué par un coefficient (que nous considérerons être entier) et un degré (que nous considérerons comme positif). Par exemple, le polynôme  $x^7 + 5x^3 - 4x + 2$  sera représenté comme suit :



---

**Questions :**

1. Donnez des définitions des types de données `monome_t` et `polynome_t`.

2. Donnez la définition de la fonction de prototype :

```
void inc_pol(polynome_t);
```

qui incrémente le degré de tous les monômes d'un polynôme de 1.

3. Donnez la définition de la fonction de prototype :

```
int eval_pol(polynome_t,int) ;
```

qui évalue un polynôme en un entier.

4. Donnez la définition de la fonction de prototype :

```
void print_pol(polynome_t) ;
```

qui retourne sur la sortie standard une chaîne de caractères représentant un polynôme.

5. Donnez une fonction de prototype :

```
void free_pol(polynome_t *) ;
```

qui détruit un polynôme.

6. Donnez la définition d'une fonction de prototype :

```
polynome_t read_pol(void) ;
```

qui lit sur l'entrée standard un polynôme donné sous la forme d'une suite de coefficients des monômes de degrés croissants. Le premier entier entré représente le coefficient du monôme de degré le plus élevé ; le nombre de coefficients entrés est égal au degré du polynôme plus 1. Par exemple, le polynôme de l'exemple ci-dessus est entré par la chaîne de caractères :

```
1 0 0 0 5 0 -4 2
```

7. Donnez la définition d'une fonction de prototype :

```
void add_mon_pol(monome_t *, polynome_t *);
```

qui additionne un monôme à un polynôme. Attention, on ne doit pas stocker les monômes de coefficient nul.