

Pointeurs

Un pointeur est une variable pouvant contenir l'adresse d'une autre variable. Le but de ce TD est d'apprendre à manipuler les pointeurs sur des variables scalaires ou structurées et sur des tableaux. Au passage, on en profitera pour faire quelques rappels sur la manipulation des structures, des tableaux et des fonctions.

► Exercice 1. Pointeur et variables scalaires

Que fait le programme suivant :

```
1 void titi(int * i, int * j){
2   if(*i < *j)
3     *i = *j;
4   else
5     *j = *i;
6 }
7
8 int main(){
9   int x=10;
10  int y=20;
11  titi(&x, &y);
12  printf("x=%d et y=%d\n", x, y);
13 }
```

On dessinera l'évolution de la pile d'appel.

► Exercice 2. Pointeurs et structures

Créer une structure `Date_t` contenant trois champs : le jour, le mois, l'année. Écrire les fonctions suivantes :

- `creer_date` qui prend en entrée trois entiers et renvoie une structure `date_t`.
- `afficher_date` qui prend en entrée une date et affiche son contenu.
- `cmp_date` qui prend en entrée deux dates `d1` et `d2` et renvoie

$$\begin{cases} 0, & \text{si } d1 = d2 \\ -1, & \text{si } d1 > d2 \\ 1, & \text{si } d1 < d2 \end{cases}$$

- `incrementer_date` qui prend en entrée un pointeur sur une `date_t` et incrémente le jour. On fera attention à changer le mois et l'année si besoin est.

Créer une structure `etudiant_t` contenant quatre champs : un nom, un prénom, un numéro d'étudiant et une date de naissance. Écrire les fonctions suivantes :

- `creer_etudiant` qui prend en entrée deux chaînes de caractères, un entier et une date et renvoie une structure `etudiant_t`.
- `afficher_etudiant` qui prend en entrée un étudiant et affiche son contenu.

- *renommer_etudiant* qui prend en entrée un pointeur sur un étudiant, un prénom, et modifie le prénom de l'étudiant.

Rappel : on pourra utiliser les fonctions de manipulations de chaînes de caractères.

- *int strlen(const char *s)* ; qui renvoie la longueur d'une chaîne de caractères.
- *int strcmp(const char *s1, const char *s2)* ; qui compare les chaînes de caractères *s1* et *s2* et renvoie 0 si les chaînes sont égales.
- *char *strcpy(char *dest, const char *src)* ; qui copie le contenu de la chaîne *src* dans *dest*.

► Exercice 3. Pointeurs et tableaux

Écrivez les fonctions suivantes :

- la fonction *affiche_tableau* qui prend en entrée un tableau d'entier *t* et sa taille et affiche le contenu du tableau.
- la fonction *init_tableau* qui prend en entrée un tableau d'entier *t*, sa taille, un élément *e*, et initialise chaque case du tableau à *e*.
- la fonction *init_tableau2D*, identique à la précédente, pour un tableau à deux dimensions.
- la fonction *recherche_element* qui prend en entrée un tableau *t*, sa taille, et un entier *x*, renvoie la position de la première occurrence de *x* et *-1* si ce dernier n'apparaît pas.
- la fonction *recherche_element_trie* dans laquelle on suppose que le tableau est trié.

► Exercice 4. Tableaux de structures

- Écrire la fonction *recherche_etudiants* qui prend en paramètre un tableau d'étudiants, sa taille, le nom d'un étudiant, et affiche la liste des étudiants ayant le même nom.
- Écrivez un programme dans lequel on déclare et on initialise un tableau d'étudiants. Utilisez la fonction *recherche_etudiants* pour afficher les étudiants ayant le nom *Grappe*.