Travaux Pratiques de langage C n°4

Cours d'Informatique de l'École Centrale d'Électronique

—Ingénieur Première Année—

Un mécanisme est disponible en langage *C* pour modifier la valeur d'un paramètre d'une fonction : pour cela, on devra manipuler des *adresses de variables*, c'est-à-dire des emplacements en mémoire servant à stocker des valeurs. Dans ce cadre, une variable qui contient elle-même une adresse sera appelée un *pointeur*. Cette séance est donc consacrée à :

- l'accès et la modification d'une case mémoire ;
- l'introduction de la notion de pointeur ;
- l'introduction de la notion de tableau ;
- la réalisation d'un programme de tri d'un tableau d'entiers par une méthode simple : le tri par recherche du minimum.

▸ **Exercice 1.** *Soit le programme C suivant :*

```
#include <stdio.h>
void main(void)
{ int a, b, *c;
  a=2; b=a; c=&a;
  printf("%d %d %d\n",a,b,*c);
  a=3;
  printf("%d %d %d\n",a,b,*c);
}
```

Commenter l'affichage du programme et rajouter une instruction qui affiche l'adresse de `a` *et la valeur de* `c`*.*

▸ **Exercice 2. (réalisation d'un swap de deux variables)** *Vérifions nos connaissances en matière de passage de paramètres à une fonction, en commentant le résultat de l'exécution du programme suivant :*

```
#include <stdio.h>

void swap(int a, int b)
{ int tmp=a;
  a=b; b=tmp;
  printf("Dans la fonction : a = %d et b = %d\n", a, b);
}
```

```
void main(void)
{ int a=5,b=3;
  printf("Avant de rentrer dans la fonction  : a = %d et b = %d\n", a, b);
  swap(a, b);
  printf("En sortant de la fonction : a = %d et b = %d\n", a, b);
}
```

Comment modifier ce programme afin qu'il réalise vraiment l'échange des valeurs des deux variables ?

▸ **Exercice 3.** *Définir une constante* `MAXTAB` *au moyen d'un* `#define`*, puis déclarer dans un programme un tableau de* `MAXTAB` *entiers que l'utilisateur du programme devra remplir au clavier (fonction* `LireTab`*). Le programme affichera enfin le tableau au moyen d'une fonction* `AffTab`*. Modifier ensuite le programme pour qu'il traite des tableaux contenant un nombre variable d'entiers.*

▸ **Exercice 4. (palindrome)** *Un* palindrome *est un mot (que l'on peut voir comme un tableau de caractères) dont la lecture droite-gauche se confond avec la lecture gauche-droite*[1]*. Si on écrit les phrases en omettant les espaces qui séparent les mots (et les accents), on peut par exemple considérer la phrase "élu par cette crapule" comme le palindrome "eluparcettecrapule". Par ailleurs, le mot vide est un palindrome.*

Écrire la fonction `Palindrome` *qui prend en premier argument un tableau de caractères et en deuxième argument le nombre de cases de ce tableau qui sont effectivement remplies (c'est-à-dire la longueur du mot contenu dans le tableau) et qui teste si le mot du tableau est un palindrome. On utilisera deux algorithmes différents : l'un sur deux tableaux, l'autre sur un seul.*

▸ **Exercice 5. (tri par recherche du minimum)** *On se propose dans cet exercice de réaliser une fonction de tri en ordre croissant d'un tableau d'entiers. On divise le problème en plusieurs étapes :*

1. *écrire une fonction de nom* `MinTab` *prenant en paramètres un tableau d'entiers ainsi que sa taille, et retournant la position du minimum des valeurs du tableau ;*

2. *maintenant que l'on sait trouver la position du minimum des n valeurs du tableau, on peut donner une technique de tri par recherche du minimum. En effet, le minimum occupera la première case du tableau trié. Ayant amené la dite valeur à sa place, il restera à trier la fin du tableau, c'est-à-dire $n-1$ valeurs. Notre algorithme s'arrêtera dès lors qu'il n'y aura plus qu'une valeur à trier. Donner une fonction de nom* `TriMin` *implémentant cette méthode.*

[1] Enfin un langage universel !