

Exercice 1 :

Écrire un programme en 'C' utilisant une variable **a** de type **int** et une variable **b** pointeur sur **int** et qui fait les affichages suivants (les ... étant remplacés par les valeurs correspondantes). L'affichage d'une adresse peut se faire dans un **printf** avec **%p**.

```
Adresse de a : ...
Valeur de a : 33

La variable b pointe maintenant sur a.
Adresse de b : ...
Valeur pointee par b : ...

La valeur affectee a la variable a est maintenant 10.
Adresse de b : ...
Valeur pointee par b : ...

La valeur pointee par la variable b vaut maintenant 20.
Adresse de a : ...
Valeur de a : ...
```

Exercice 2 :

Parmi les trois fonctions suivantes, quelles sont celles provoquant des problèmes en lien avec la gestion des pointeurs ? Préciser pourquoi et corriger les.

<pre>int *p1() { int a; a = 1; return (&a); }</pre>	<pre>int *p2() { int *p; *p = 1; return p; }</pre>	<pre>int *p3() { int *p; p = (int *) malloc(sizeof(int)); *p = 1; return p; }</pre>
---	--	---

Exercice 3 :

On souhaite faire recueillir et stocker les dates de différents événements intervenants dans un processus. Pour cela, on associe à une date un jour de la semaine (entier de 1 à 7), une heure représentée par un triplet (heure, minutes, secondes) au format 16:33:49.26. On note que les secondes sont stockées sous la forme d'un réel, contrairement aux heures et minutes.

1. Définir un type **Date**.
2. Écrire une fonction de saisie (**par adresse**) d'une variable de type **Date**.
3. Écrire une fonction d'allocation mémoire et d'initialisation d'un tableau de **n Date**, avec **n** passé en paramètre.
4. Écrire une fonction d'affichage d'un tableau de **n Date** donné.
5. Écrire un programme de test et de libération mémoire.

On considère maintenant une structure **DesDates** contenant un tableau de **Date** et le nombre de dates stockées dans le tableau.

6. Définir un type **DesDates**.
7. Écrire une fonction de saisie d'une variable de type **DesDates** (par adresse) qui contiendra **n** dates.
8. Écrire une fonction qui trie les **n** dates contenues dans une variable **DesDates** donnée dans l'ordre chronologique (prise en compte du jour, puis de l'heure).
9. Écrire une fonction qui initialise un tableau de **m DesDates**, chaque variable de type **DesDates** contenant un nombre à lire de **Date**. Comment gérer ce nombre de dates potentiellement distinct pour chaque variable ?
10. Écrire une fonction qui recherche et affiche toutes les dates associées à un jour donné dans une variable **DesDates** donnée contenant **n Dates**.
11. Modifier le programme pour tester ces sous-programme, sans oublier de libérer la mémoire allouée.