

Programmation Impérative 2 - L2 Informatique

TP2 - Début sur les pointeurs

Exercice I - Ecrire une fonction `change` avec pour type de retour `void` et une variable entière en argument et dont l'objectif est de mettre le contenu de la variable passée en argument à 5. On testera deux cas: un où la fonction prend directement la variable en argument et le second où elle prend un pointeur sur la variable. Vérifiez que seule la seconde version fonctionne.

Exercice II - Ecrire le module `allocation.c` vu en cours.

Exercice III - Ecrire un programme qui alloue un tableau de caractères dont la taille est donnée par l'utilisateur, puis remplit le tableau par des caractères tirés aléatoirement (les valeurs pourront correspondre à la table ASCII). Le programme doit ensuite afficher le tableau de gauche à droite, puis de droite à gauche en parcourant le tableau à l'aide d'un pointeur (et non pas à l'aide d'une variable de type compteur). Le programme libèrera ensuite l'espace mémoire alloué avant de terminer.

Testez ce qui se passe si on essaie de pointer des cases en dehors du tableau alloué (zones de mémoire situées avant ou après, et plus ou moins éloignées).

Exercice IV - Ecrire un programme qui calcule et affiche le triangle de Pascal, le nombre de lignes étant donné sur la ligne de commande :

```
>pascal 5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

La formule pour le calcul de chaque case est donnée par

$$C[i, j] = C[i - 1, j - 1] + C[i - 1, j]$$

de sorte que chaque ligne commence par un 1.

Le triangle devra être stocké dans une matrice allouée dynamiquement de sorte que chaque ligne contienne uniquement le nombre de cases nécessaires. Une fois le triangle affiché, les espaces alloués devront être libérés.

L'utilisateur devra pouvoir rentrer plusieurs entiers sur la ligne de commande, dans ce cas le programme affichera autant de triangle de Pascal qu'il y a d'argument sur la ligne.

Exercice V - Vous devrez faire le TP d'Anne Canteaut sur le débogueur `gdb`, présent dans l'annexe B du poly donné en cours (le tp seul est aussi disponible sur Claroline sous le nom `tp_gdb.pdf`). Les fichiers `exemple.c` et `entree.txt` sont disponibles sur Claroline. Effectuez le TP au moins jusqu'à la question 9. Lisez au moins le reste du TP pour connaître les différentes possibilités si vous n'avez pas le temps de toutes les tester.

A noter qu'une interface graphique existe: `ddd`.

Vous pouvez aussi expérimenter une autre situation de buffer overflow avec des chaînes et `scanf` par exemple.