Licence de Chimie / Physique-Chimie

Algorithmique et Programmation : suppléments

La réalisation de ces exercices peut impliquer la connaissance de tout ou partie du programme du semestre. Ils ne sont donc pas associés à un chapitre déterminé. Le but de ces exercices est de vous faire travailler sur des problèmes nouveaux dans le cadre des révisions de fin de semestre.

Certains exercices sont plus longs que d'autres, mais aucun n'est *très* difficile : il suffit de réfléchir au problème, de créer votre programme pas à pas et – éventuellement – de poser des questions. Les parties avancées des exercices nécessitent en général une réflexion plus poussée, mais pas de nouvelles connaissances en langage C.

1) Nombre aléatoire

La fonction `rand()` renvoie un nombre entier aléatoire, parfois très grand. Écrivez une fonction `random(int n)` qui renvoie un nombre entier aléatoire entre 0 et n. Écrivez une fonction `frandom()` qui renvoie un nombre réel aléatoire dans l'intervalle $[0; 1[$ (indice : la constante `RAND_MAX` est la plus grande valeur possible pour la fonction `rand`). Vérifiez le programme en faisant afficher plusieurs nombres à la suite.

2) Cryptographie

Écrivez une fonction `encode` qui prend un caractère comme argument et renvoie le caractère suivant. Écrivez une fonction `decode` qui prend un caractère comme argument et renvoie le caractère précédent. Le programme principal demande à l'utilisateur une phrase (avec espace : utiliser `gets()`). Ensuite, il affiche la phrase encodée puis la phrase décodée.
Avancé : réfléchissez à une façon plus complexe de coder une phrase mettant en jeu le générateur de nombre aléatoire de l'exercice 1. Que devront utiliser les fonctions encode() et decode() ?

3) Jolies couleurs

Il est possible d'écrire du texte en couleur sur le terminal. Par exemple :

```
print "\033[32mBonjour!\033[0m\n";
```

Cette ligne écrira «Bonjour !» en vert (32) et la couleur revient à la normale (0) à la fin de la ligne. Écrivez un programme qui demande une phrase à l'utilisateur et la recopie avec les voyelles et le consonnes de deux couleurs différentes. Les couleurs possibles sont : noir (30), rouge (31), vert (32), jaune (33), bleu (34), magenta (35), cyan (36), blanc (37), et défaut (0).
Avancé : affichez chaque *mot* de la phrase avec une couleur différente prise au hasard. On utilisera la fonction random() définie dans l'exercice 1.

4) Jolies couleurs (2)

Cet exercice peut se faire juste après avoir fait le triangle de Pascal dans les exercices de TP classiques. Modifiez ce programme pour qu'il n'écrive pas les éléments du triangle de Pascal, mais un caractère `O` rouge si l'élément est pair et un caractère `.` bleu si l'élément est impair. Affichez ensuite ce nouveau triangle de Pascal avec au moins 20 lignes. Que constatez-vous ?

Avancé : vous pouvez faire la même chose en distinguant les multiples de 3 ou plus.

5) Calcul de π

L'aire d'un disque est πr^2 où r est le rayon du disque. Si on considère le quart nord-est du plan ($x > 0$ et $y > 0$), le rapport entre les aires du quart de cercle de rayon 1 et le carré de côté 1 est égal à $\pi/4$. En tirant aléatoirement un point dans ce carré, il aura donc une probabilité $p = \pi/4$ de se trouver à l'intérieur du quart de cercle. On peut donc utiliser un simple générateur de nombre aléatoire pour calculer π...
Écrivez un programme qui demande à l'utilisateur combien d'essais il veut faire. Ensuite, pour chaque essai, on génère un point de coordonnées (x, y) avec $x \in [0; 1[$ et $y \in [0; 1[$. Si ce point se trouve à l'intérieur du cercle (soit $r < 1$), on compte cet essai comme accepté. À la fin du programme, le rapport du nombre d'essais acceptés sur le nombre total d'essais doit converger vers $\pi/4$: écrivez tous les 100000 essais une ligne indiquant le numéro de l'essai, la valeur du rapport et la valeur exacte de $\pi/4$ pour comparer. Que constatez-vous ?
Avancé : cette technique constitue la base de la méthode de Monte Carlo. Faites une recherche rapide sur ses applications usuelles en chimie.

6) Barre de progression

Lorsqu'un programme met du temps à accomplir une tâche, il est pratique pour l'utilisateur de savoir où on en est, d'où l'utilité d'afficher la progression sur une seule ligne constamment remise à jour. Écrivez un programme qui écrit où en est la boucle principale en affichant un texte de ce genre :

```
Progression: 45 / 478
```

Ici, on en est à l'itération 45 sur un total de 478. On peut aussi rajouter une indication en pourcentage, plus pratique. À l'intérieur de la boucle, une instruction `system("sleep 0.01s")` permet d'avoir le temps de voir le texte. Vous pouvez superposer les lignes de deux façons :
- Le caractère `\r` permet de ramener le curseur en début de ligne : à placer au début de la ligne de texte. Quelles sont ses limitations ?
- La séquence `\033[1A\033[2K` fait remonter le curseur d'une ligne et efface la ligne.

Avancé : vous pouvez améliorer l'effet visuel en insérant une barre de progression. Essayez de reproduire le rendu suivant en utilisant les couleurs de votre choix :

```
Progression: [##########__________________________________]  24%
```

Faites-en ensuite une fonction facilement portable et utilisez-la dans le calcul de π de l'exercice précédent en remplacement de l'affichage demandé.

7) Cristal

On dispose d'un fichier `diamond.cell` contenant la description d'une maille élémentaire de diamant. Écrivez un programme permettant d'exécuter les étapes suivantes :
- lecture du fichier, stockage des informations dans des variables adéquates
- demande à l'utilisateur du nombre de mailles suivant x, y et z
- duplication de la maille élémentaire et écriture du cristal au format PDB (*Protein Data Bank*) dans `output.pdb`

Le format PDB peut être lu par Rasmol, un logiciel de visualisation en 3D. Chaque ligne décrit un atome grâce à son indice, son nom et ses coordonnées (dans cet ordre) :

```
"ATOM %6d %2s           1    %8.3f%8.3f%8.3f\n"
```

Avancé : Créez vos propres fichiers de maille élémentaire avec le même format que pour le diamant. Vous pouvez ensuite demander à l'utilisateur quel composé il souhaite créer.