**TP3 : Tableaux**

# 1 Quelques problèmes simples

Pour chacun des exercices suivants, écrire un programme en C en utilisant des tableaux pour résoudre le problème. Chaque programme commence par demander un nombre $n$ à l'utilisateur (si besoin), puis résoud le problème.

## 1.1 Interrupteurs

On dispose de $n$ interrupteurs numérotés de 1 à $n$ reliés à $n$ lampes. Initialement, toutes les lampes sont éteintes. On effectue ensuite les manipulations suivantes :

- appuyer sur tous les interrupteurs dont le numéro est un multiple de 1 (donc tous),
- refaire de même avec tous les interrupteurs dont le numéro est un multiple de 2,
- etc, jusqu'à appuyer sur les multiples de $n$ (donc juste le dernier interrupteur).

Quelles sont les lampes allumées à la fin de la procédure ?

## 1.2 Cartes

On prend un jeu de $n$ cartes numérotées de 1 à $n$, puis on effectue la manipulation suivante : on prend la première carte et on la met derrière le paquet, on jette la suivante. On continue de la sorte en gardant une carte à chaque fois et en jetant la suivante jusqu'à ce qu'il ne reste plus qu'une carte.

Quelle carte reste à la fin ?

## 1.3 Nombres aléatoires

*[Utiliser la fonction random de la librairie stdlib].*

Remplir un tableau de taille $n$ avec des nombres aléatoires tirés entre 1 et $k$ puis répondre (en affichant le résultat) aux questions suivantes :

- quels sont les plus petit et plus grand éléments du tableau (un seul parcours est suffisant),
- quelle est la fréquence d'apparition de chaque nombre,
- quelle est la distribution du nombre d'apparitions des nombres, c'est-à-dire combien de nombres apparaissent une seule fois, combien apparaissent deux fois...
- trier le tableau par la méthode de votre choix.

## 1.4 Permutations

Une permutation de $\{1..n\}$ est un ordonnancement de ces chiffres. Par exemple, une permutation de $\{1..5\}$ serait $(3, 2, 5, 1, 4)$.

On peut tirer une permutation "aléatoire" en procédant de la manière suivante :

1. partir de la permutation identité : $(1, 2, 3, 4, \cdots, n)$,
2. pour $k$ allant de 0 à $n-2$, tirer un nombre $l$ entre 1 et $n-k$ et échanger les nombres aux positions $l$ et $n-k$.

Écrire un programme pour répondre aux questions suivantes :
- calculer la probabilité d'avoir la permutation identité en générant des permutations aléatoires (on prendra des permutations de taille 7 et on fera 10 000 ou 100 000 tirages),
- calculer la probabilité d'avoir une autre permutation fixée (par exemple $(7, 6, 5, 4, 3, 2, 1)$),
- *question subsidiaire : quelle est la valeur exacte de cette probabilité ?*

# 2 Calculs sur de grands entiers

On va représenter des nombres de grande taille par un tableau d'entiers dont chaque case représente un chiffre. On peut stocker de tels nombres de deux façons, en mettant le chiffre des unités dans la première ou bien dans la dernière case du tableau...

On veut effectuer les opérations de base sur ces entiers en reprogrammant celles-ci (attention, il faut trouver un moyen de gérer le signe de ces entiers longs).

## 2.1 Entiers de taille bornée

Dans un premier temps, on se limite à des entiers dont la taille maximale est fixe (100 chiffres par exemple). Écrire plusieurs fonctions pour répondre aux questions suivantes :
- saisie d'un entier au clavier (et création d'un entier long),
- affichage d'un entier long,
- somme et différence de deux entiers,
- produit de deux entiers,
- factorielle d'un entier.

Quelle est la plus grande factorielle calculable avec des entiers de taille bornée par 100 ?

## 2.2 Entiers de taille variable

Sachant que le nombre de chiffres de $k!$ est majoré par $2k\sqrt{k}$, modifier le programme précédent pour calculer $k!$ pour un $k$ donné.