**INF 111, L1, 2004/2005**

**dernier TP**

Décembre 2004

# TP 12 - Coupes d'un tableau

*Laure Gonnord*
`http://www-verimag.imag.fr/~gonnord/`

## Problème – Coupes d'un Tableau

Le sujet propose divers parcours d'un tableau $T$ de taille $N = 1000$ dont tous les éléments valent 0 ou 1. Le tableau est initialisée comme suit. On définit tout d'abord la suite récurrente d'entiers suivante :

- $x_0 = 13$
- $x_i = (x_{i-1} \times x_{i-1})$ modulo 3953 pour $i$ compris entre 1 et $N - 1$.

Ensuite, pour tout $1 \leq i \leq N - 1$, on remplit le tableau par $T[i] := x_i$ modulo 2.

### Visualisation, initialisation, vérifications

#### Question 1

Déclarez la constante `N`, ainsi que les constantes entières `VRAI` et `FAUX` qui nous serviront par la suite pour donner des résultats booléens aux fonctions.

#### Question 2

Ecrire une procédure `visu` qui prend un argument un tableau `t` d'entiers de taille `N`, ainsi que deux entiers `i` et `j`, qui :

- Vérifie que les entiers $i$ et $j$ sont bien des bons indices, et vérifient $i \leq j$ sinon on imprime un message d'erreur ;
- Si ces conditions sont vérifiées, on imprime le sous-tableau $t[i..j]$, c'est-à-dire les éléments du tableau `t` d'indice compris entre (au sens LARGE) $i$ et $j$.

#### Question 3

Déclarer (dans le `main`) le tableau `T` de travail, et l'initialiser afin que pour tout $0 \leq i \leq N - 1$, `T[i]` contienne les éléments de la suite, c'est-à-dire uniquement des 0 et des 1.

#### Question 4

Écrire une fonction qui prend en argument un tableau $t$ d'entiers de taille $N$ et qui retourne le nombre de cases égales à 1 du tableau $t$.
Remplir le programme principal pour appeler cette fonction sur le tableau `T`.

### Notion de coupe, palindromes

Une **coupe** $C[i, j]$ du tableau $T$ est une suite d'éléments consécutifs $T[i]$, $T[i+1]$, $\ldots T[j]$ (un "sous-tableau"), où $0 \leq i \leq j \leq N - 1$. Noter que $C[i, j]$ est de taille $j - i + 1$.
Une coupe $C[i, j]$ est un **palindrome** si elle est égale à la coupe obtenue en prenant ses éléments à l'envers, c'est-à-dire si $T[i] = T[j]$, $T[i + 1] = T[j - 1]$, $T[i + 2] = t[j - 2]$, $\ldots$. Un palindrome peut bien entendu être de longueur impaire.

**Question 5**

Soit le tableau $T$ suivant : `[1 1 1 0 0 1 0 0 1 1]`

– La coupe $T[0, 5]$ est-elle un palindrome ? La coupe $T[2, 8]$ ? Justifier.

– Quelle est la longueur de la plus grande coupe de $T$ qui est un palindrome ?

**Question 6**

On cherche dans un premier temps à déterminer si il existe dans $T$ des coupes de longueur 7 qui sont des palindromes :

1. Écrire une fonction `palin_depuis` qui prend un argument un tableau `t` et un indice $i$ (convenant) et qui dit si le sous-tableau de taille 7 commençant à l'indice $i$, c'est-à-dire $t[i..i+6]$ est un palindrome (il retourne alors `VRAI`) ou pas (il retourne FAUX).
2. Tester l'exécution de votre algorithme "à la main" sur le tableau $T$ de la question 5, avec $i = 0$, puis $i = 2$.
3. En utilisant la fonction précédente, écrire une fonction `palin7` qui prend en argument un tableau $t$ et qui renvoie `VRAI` si le tableau $t$ contient un palindrome de taille 7, `FAUX` sinon.
4. Modifier la fonction `palin7` pour qu'elle imprime le plus petit indice $i$ tel que $t[i, i + 6]$ soit un palindrome de longueur 7. Comment faire pour vérifier visuellement que cette coupe est bien un palindrome ?
5. Compléter la fonction `main` pour rechercher un palindrome de taille 7 dans le tableau `T`.

**Question 7**

On cherche à déterminer la longueur de la plus longue coupe de $T$ qui soit un palindrome.

1. Écrire une fonction `est_palin` qui prend un argument un tableau $t$, un indice de départ $i$ et une longueur $l$ de coupe, et qui dit si la coupe $t[i, i + l - 1]$ est un palindrome (retourne `VRAI` si oui, `FAUX` sinon).
2. En utilisant la fonction précédente, écrire une fonction `palin_max` qui prend en paramètre un tableau $t$ et qui retourne la taille de la plus grande coupe de $t$ qui est un palindrome. On pourra tester différentes valeurs de $l$, en commençant par $l = N$. Pour chaque valeur de $l$, on testera les différentes valeurs de $i$ possible. Quand s'arrête-t-on ?
3. Modifier la fonction précédente pour qu'elle imprime en plus le nombre de tels palindromes et la position du premier. (écrire `palin_max2`)

## Coupes nulles, coupes équilibrées

**Question 8**

On cherche à déterminer la longueur maximale d'une coupe constituée uniquement d'éléments nuls (on appelle celà une "coupe nulle").

1. Écrire une fonction `coupe_nulle_maxi_depuis` qui prend un argument un tableau $t$, un indice de départ $i$ et qui retourne la longueur de la coupe nulle la plus grande commençant à l'indice $i$. (Remarquer que si $t[i] = 1$, alors le résultat est 0.)
2. À l'aide de la fonction précédente, écrire une fonction `plus_longue_nulle` qui prend en paramètre un tableau $t$ et qui retourne la taille de la plus grande coupe nulle de $t$.

**Question 9**

Une coupe est dite **équilibrée** si elle comprend autant de 0 que de 1. On cherche à déterminer la longueur maximale d'une coupe équilibrée, combien il y a de telles coupes, ainsi que l'indice de la première. Écrire une fonction `equil_max` qui réalise ce travail. Comment améliorer votre fonction ?