Faculté des Sciences de Sfax
Département Informatique et des Communications

Année Universitaire : 2020-2021
Section : Licence en Sciences de l'Informatique
Niveau: 1ère année
Matière : Atelier de Programmation
Enseignant : Mohamed Ali Hadj Taieb

Atelier de Programmation ***** TP3

Objectifs:

- Définir des algorithmes récursifs
- Simuler l'exécution d'une solution récursive

Exercice 1 :

Comparer les solutions récursives des algorithmes du Tri Rapide et du Tri Fusion

Exercice 2 :

Donner une solution récursive en C pour la décomposition en facteur premier d'un nombre entier.

Exercice 3 :

a. Ecrire une solution récursive pour la recherche dichotomique.
b. Comparer les solutions itérative et récursive.

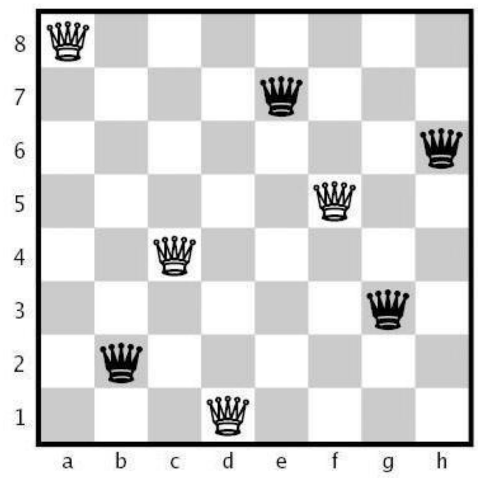
Exercice 4 :

Problème de Reines :

Le problème initialement posé par K. F. Gauss en 1842 est le suivant : est-il possible de placer huit reines sur un échiquier sans qu'aucune reine n'en menace une autre ? Pour ceux qui ne connaissent pas les règles du jeu d'échec, ou du moins la partie des règles qui concernent la reine, une reine "menace" toutes les pièces de l'échiquier qui sont placées

1. sur la même rangée;
2. sur la même colonne;
3. sur la même diagonale montante;
4. sur la même diagonale descendante;

Proposez une solution récursive.



Une solution du problème des reines

Exercice 5 :

Problème du Tour de Hanoi

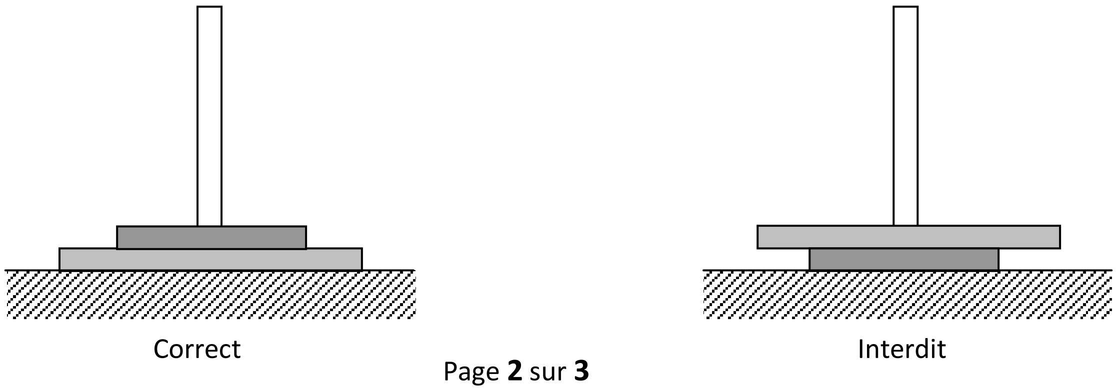
Les Tours de Hanoi sont un jeu constitué de trois tours symbolisées par des piquets, sur les quels sont enfilés des disques de tailles différentes, qui forment une pyramide cylindrique. Le nombre de disques est défini au début de la partie, et il est de trois minimums. Le but du jeu est de déplacer la pyramide située initialement sur la tour 1 vers la tour 3, comme le montre la figure 2 qui présente la position initiale et la position finale avec trois disques.

Chaque phase de jeu ne déplace qu'un seul disque à la fois, d'une tour vers une autre en respectant l'ordonnancement des disques, du plus petit (en haut) vers le plus grand (en bas), sur chaque piquet. Il est donc interdit de déplacer un disque plus grand (en bas), sur chaque piquet. Il est donc interdit de déplacer un disque plus grand vers une tour contenant en haut de la pile un disque plus petit. Voici un exemple de début du jeu : déplacer le premier disque de la tour 1 vers la tour 2, déplacer le premier disque de la tour 3 vers la tour 2. Cette suite d'actions équivaut à déplacer les deux premiers disques de la tour 1 vers la tour 2, en respectant les règles indiquées précédemment.

Il s'agit d'un jeu de réflexion dont voici la règle. Des anneaux de diamètres différents sont empilés sur un poteau. Un anneau peut être empilé sur un autre seulement s'il a un diamètre inférieur à celui de l'anneau sur lequel il repose.

Figure 1

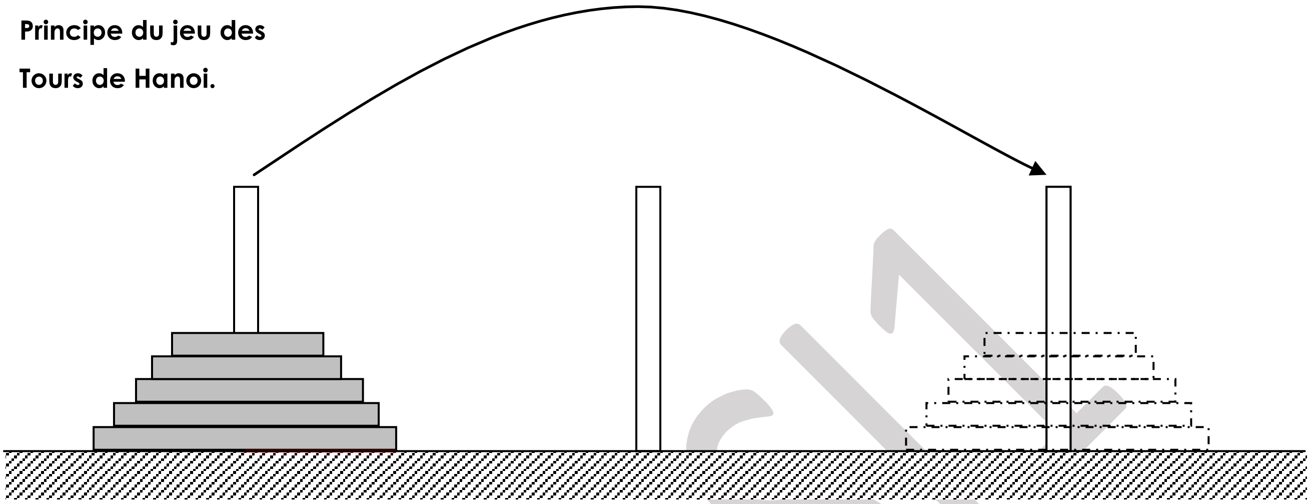
Description de la règle.



Le but de jeu est de déplacer les anneaux initialement empilés sur un seul poteau vers un autre en respectant les règles et en n'utilisant qu'un seul poteau intermédiaire.

Figure 2

Principe du jeu des Tours de Hanoi.



Les poteaux sont représentés par des piles d'entiers numérotées 0, 1 et 2. Le jeu entier sera

Exercice 6 :

Exponentiation Rapide :

L'exponentiation rapide est un algorithme utilisé pour calculer rapidement, de grandes puissances entières. En anglais, cette méthode est aussi appelée square-and-multiply ("*mettre au carré et multiplier*"). La première façon de calculer une puissance $\mathbf{n^p}$, est de multiplier **n** par lui-même **p** fois. Cependant, il existe des méthodes bien plus efficaces, où le nombre d'opérations nécessaires n'est plus de l'ordre de **p**.

Soit n un entier strictement supérieur à 1, supposons que l'on sache calculer pour tout x appartenant à l'ensemble des réels, toutes les puissances $\mathbf{x^k}$ de **x**, pour tout **k**, tel que $1 \le k < n$.

- Si n est pair alors $x^n = (x^2)^{\frac{n}{2}}$. Il suffit alors de calculer $y^{\frac{n}{2}}$ avec $y = x^2$.
- Si n est impair et n>1, alors $x^n = x \times (x^2)^{\frac{(n-1)}{2}}$. Il suffit de calculer $y^{\frac{(n-1)}{2}}$ avec $y = x^2$ et de multiplier par x le résultat.

Cette remarque nous amène à l'algorithme récursif suivant qui calcule $\mathbf{x^n}$ pour un entier strictement positif **n**:

$$\text{puissance}(x,\, n) = \begin{cases} x, & \text{si } n = 1 \\ \text{puissance}(x^2,\, n/2), & \text{si } n \text{ est pair} \\ x \times \text{puissance}(x^2,\, (n-1)/2), & \text{si } n > 2 \text{ est impair} \end{cases}$$

Proposer une solution récursive en C et étudier sa complexité.