

TP 6

Adresse du cours et des corrections des TPs précédents : <https://axlbonnet.github.io/dut-gim-pres>

TP6-1

On veut écrire un programme qui va permettre de savoir si un triangle est rectangle. L'objectif est de l'écrire petit à petit à l'aide de plusieurs méthodes.

- 1) Ecrire une méthode "carre" qui prend en unique paramètre un nombre entier et retourne sa valeur au carré. Appelez la depuis la méthode principale "main" pour afficher les carrés de 5 et 142.
- 2) Ecrire une méthode "estRectangle" qui prend en paramètres 3 nombre entiers et retourne le booléen indiquant si le triangle formé par ces 3 côtés est rectangle (en partant du principe que le 1er paramètre est le plus grand). Le booléen doit être retourné avec le type "int".
Changer la méthode principale "main" pour tester si les triangles (7,3,3) et (5,4,3) sont rectangles.
- 3) Ecrire une méthode "obtenirCote" qui ne prend pas de paramètre mais retourne une dimension saisie par l'utilisateur forcément positive. Testez la depuis la méthode principale.
- 4) Ecrire le programme complet en utilisant ces 3 méthodes, en considérant que l'utilisateur saisit toujours le plus grand côté en premier.
- 5) Améliorez le programme pour prendre en compte le fait que l'utilisateur ne saisisse pas forcément le plus grand côté en premier.

TP6-2

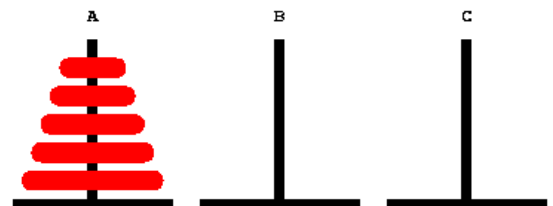
On veut trier un tableau de 6 entiers (saisis par l'utilisateur), c'est à dire réarranger ses éléments pour qu'ils soient dans un ordre croissant. On utilisera l'algorithme appelé "tri à bulle". Le but est de balayer le tableau et de comparer et inverser si nécessaire chaque paire d'éléments consécutifs. Un seul balayage ne suffisant pas nécessairement, on recommence l'opération jusqu'à obtenir un balayage sans aucune modification.

- 1) Simuler l'algorithme sur papier avec les valeurs [5,53,12,42,3]
- 2) Ecrire le programme en suivant ces conseils :
 - le tableau sera stocké dans une variable globale
 - une méthode permet de savoir si le tableau est trié
 - une méthode permet d'inverser deux éléments successifs du tableau

TP6-Bonus

Reprendre le TP sur les tours de Hanoï et (re)écrire un programme permettant d'y jouer en utilisant des méthodes. On pourra par exemple écrire une méthode qui permet d'afficher le plateau, une autre permettant de déplacer un disque entre 2 tours, une autre permettant de savoir si un coup est valide (et quelques autres si besoin).

Les tableaux contenant les états des 3 tours seront stockés dans des variables globales. Il est conseillé de stocker dans un tableau supplémentaire combien de disques contient chacune des trois tours.



Suite) Trouver l'algorithme permettant de résoudre ce problème et essayer de le simuler dans le programme. (Indices : intéressez-vous particulièrement aux déplacements du plus petit disque)

TP 6

Adresse du cours et des corrections des TPs précédents : <https://axlbonnet.github.io/dut-gim-pres>

TP6-1

On veut écrire un programme qui va permettre de savoir si un triangle est rectangle. L'objectif est de l'écrire petit à petit à l'aide de plusieurs méthodes.

- 1) Ecrire une méthode "carre" qui prend en unique paramètre un nombre entier et retourne sa valeur au carré. Appelez-la depuis la méthode principale "main" pour afficher les carrés de 5 et 142.
- 2) Ecrire une méthode "estRectangle" qui prend en paramètres 3 nombres entiers et retourne le booléen indiquant si le triangle formé par ces 3 côtés est rectangle (en partant du principe que le 1er paramètre est le plus grand). Le booléen doit être retourné avec le type "int".
Changer la méthode principale "main" pour tester si les triangles (7,3,3) et (5,4,3) sont rectangles.
- 3) Ecrire une méthode "obtenirCote" qui ne prend pas de paramètre mais retourne une dimension saisie par l'utilisateur forcément positive. Testez-la depuis la méthode principale.
- 4) Ecrire le programme complet en utilisant ces 3 méthodes, en considérant que l'utilisateur saisit toujours le plus grand côté en premier.
- 5) Améliorez le programme pour prendre en compte le fait que l'utilisateur ne saisisse pas forcément le plus grand côté en premier.

TP6-2

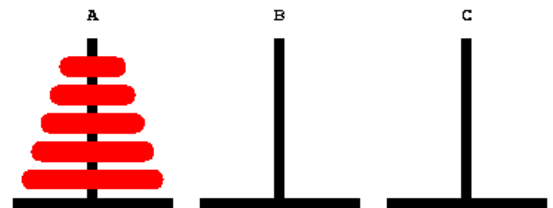
On veut trier un tableau de 6 entiers (saisis par l'utilisateur), c'est à dire réarranger ses éléments pour qu'ils soient dans un ordre croissant. On utilisera l'algorithme appelé "tri à bulle". Le but est de balayer le tableau et de comparer et inverser si nécessaire chaque paire d'éléments consécutifs. Un seul balayage ne suffisant pas nécessairement, on recommence l'opération jusqu'à obtenir un balayage sans aucune modification.

- 1) Simuler l'algorithme sur papier avec les valeurs [5,53,12,42,3]
- 2) Ecrire le programme en suivant ces conseils :
 - le tableau sera stocké dans une variable globale
 - une méthode permet de savoir si le tableau est trié
 - une méthode permet d'inverser deux éléments successifs du tableau

TP6-Bonus

Reprendre le TP sur les tours de Hanoï et (re)écrire un programme permettant d'y jouer en utilisant des méthodes. On pourra par exemple écrire une méthode qui permet d'afficher le plateau, une autre permettant de déplacer un disque entre 2 tours, une autre permettant de savoir si un coup est valide (et quelques autres si besoin).

Les tableaux contenant les états des 3 tours seront stockés dans des variables globales. Il est conseillé de stocker dans un tableau supplémentaire combien de disques contient chacune des trois tours.



Suite) Trouver l'algorithme permettant de résoudre ce problème et essayer de le simuler dans le programme. (Indice : intéressez-vous particulièrement aux déplacements du plus petit disque)