TP 5

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

TP5-1

Ecrire un algorithme qui demande un nombre entier n, compris entre 1 et 13 (on supposera que la saisie est bonne et qu'il n'est pas nécessaire de la vérifier). L'algorithme demandera ensuite à l'utilisateur de saisir n nombres entiers, les stockera tous dans un tableau, puis réaffichera seulement les nombres saisis qui sont des multiples de 3.

1) Quelle taille doit faire le tableau pour pouvoir stocker toutes les nombres saisis quelle que soit n?

2) Ecrire cet algorithme sur papier (inspirez vous de l'exemple du cours)

3) Adapter l'algorithme en C

TP5-2

Ecrire un algorithme qui demande un nombre entier n compris entre 1 et 17 puis invite l'utilisateur à saisir n nombres réels qui seront stockés dans un tableau. L'algorithme demandera ensuite un nombre réel m et affichera combien de nombres parmis les n saisis sont supérieur à m. L'algorithme s'assurera que le nombre n saisi au départ respecte les conditions requises et redemandera un saisie tant que nécessaire.

1) Ecrire sur papier la partie de l'algorithme qui permet d'obtenir un nombre n valide

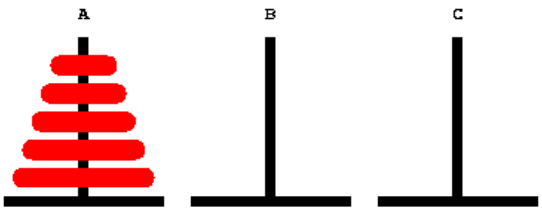
2) Ecrire l'algotrihme entier sur papier

3) Adapter l'algorithme en C

TP5-3

On dispose de 3 tours sur lesquels sont empilés n disques de largeurs différentes. Au début, tous les disques sont empilés du plus grand au plus petit sur la 1ere tour. Le but est de déplacer tous les disques sur la 3eme tour en respectant les règles suivantes :

- On ne peut déplacer qu'un disque à la fois
- On ne peut placer un disque que sur un disque plus grand que lui ou sur un emplacement vide .



1) Résoudre le problème sur papier pour n=3 et 4

2) Ecrire un programme C permettant de jouer à ce jeu. On pourra représenter chaque disque par le nombre symbolisant sa taille

3) Trouver l'algorithme permettant de résoudre ce problème et essayer de l'implémenter en C. (Indice : interessez-vous particulièrement aux déplacements du plus petit disque)

TP 5

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

TP5-1

Ecrire un algorithme qui demande un nombre entier n, compris entre 1 et 13 (on supposera que la saisie est bonne et qu'il n'est pas nécessaire de la vérifier). L'algorithme demandera ensuite à l'utilisateur de saisir n nombres entiers, les stockera tous dans un tableau, puis réaffichera seulement les nombres saisis qui sont des multiples de 3.

1) Quelle taille doit faire le tableau pour pouvoir stocker toutes les nombres saisis quelle que soit n?

2) Ecrire cet algorithme sur papier (inspirez vous de l'exemple du cours)

3) Adapter l'algorithme en C

TP5-2

Ecrire un algorithme qui demande un nombre entier n compris entre 1 et 17 puis invite l'utilisateur à saisir n nombres réels qui seront stockés dans un tableau. L'algorithme demandera ensuite un nombre réel m et affichera combien de nombres parmis les n saisis sont supérieur à m. L'algorithme s'assurera que le nombre n saisi au départ respecte les conditions requises et redemandera un saisie tant que nécessaire.

1) Ecrire sur papier la partie de l'algorithme qui permet d'obtenir un nombre n valide

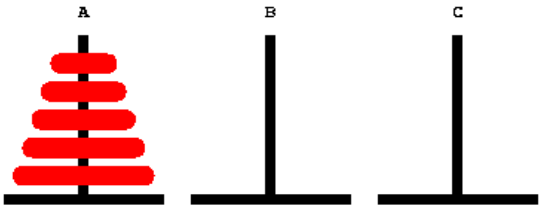
2) Ecrire l'algotrihme entier sur papier

3) Adapter l'algorithme en C

TP5-3

On dispose de 3 tours sur lesquels sont empilés n disques de largeurs différentes. Au début, tous les disques sont empilés du plus grand au plus petit sur la 1ere tour. Le but est de déplacer tous les disques sur la 3eme tour en respectant les règles suivantes :

- On ne peut déplacer qu'un disque à la fois
- On ne peut placer un disque que sur un disque plus grand que lui ou sur un emplacement vide .



1) Résoudre le problème sur papier pour n=3 et 4

2) Ecrire un programme C permettant de jouer à ce jeu. On pourra représenter chaque disque par le nombre symbolisant sa taille

3) Trouver l'algorithme permettant de résoudre ce problème et essayer de l'implémenter en C. (Indice : interessez-vous particulièrement aux déplacements du plus petit disque)