

TP 4

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

A la fin du TP se trouvent des exercices des précédents TPs avec des indications supplémentaires. S'ils n'avaient pas été fait, ils devront être fait à la maison pour la semaine prochaine.

Rappel syntaxe de la boucle “tant que”

Problème	Algorithmie	C
Demander un nombre jusqu'à ce qu'il soit un multiple de 3	Entier nombre obtenir nombre tant que ((nombre % 3) ≠ 0) afficher nombre “pas multiple de 3” obtenir nombre fintantque afficher nombre “multiple de 3”	<pre>int nombre; scanf("%d", &nombre); while ((nombre % 3) != 0) { printf("%d pas multiple de 3\n", nombre); scanf("%d", &nombre); } printf("%d mult. de 3\n", nombre);</pre>
Demander un nombre. Chercher le plus grand nombre carré qui soit strictement plus petit que ce nombre	Entier nombre, rac, solution obtenir nombre rac ← 0 tant que ((rac x rac) < nombre) rac ← rac + 1 fintantque rac ← rac - 1 solution ← rac x rac afficher solution	<pre>int nombre, rac, solution; scanf("%d", &nombre); rac ← 0 while ((rac * rac) < nombre) { rac = rac + 1; } rac = rac - 1; solution = rac * rac; printf("solution : %d", solution);</pre>

TP4-1

Un éleveur de lapin dispose de 28 lapins adultes et de 53 jeunes lapereaux. Tous les ans, chaque couple de lapin adulte produit 3 nouveaux lapereaux. On considère que les males et femelles sont équitablement répartis et qu'un lapereau devient adulte après 1 an.

On désire savoir combien il y aura de lapins (adultes et enfants confondus) après un certain nombre d'années, choisi par l'utilisateur.

- 1) Quelle structure de controle (“si” ou “pour” ou “tant que”) est la plus appropriée pour écrire un algorithme solutionnant ce problème. Pourquoi ?
- 2) Ecrire cet algorithme sur papier (vous pouvez vous inspirer des corrections du TP1)
- 3) Adapter l'algorithme en C

TP4-2

Le problème est le même que pour l'exercice précédent, sauf que l'on désire savoir au bout de combien d'années le nombre de lapins (adultes et enfants confondus) dépassera strictement une certaine valeur, choisie par l'utilisateur.

- 1) Quelle structure de controle est la plus appropriée. Pourquoi ?
- 2) Ecrire cet algorithme sur papier
- 3) Adapter l'algorithme en C

Nouveauté : les boucles “pour” avec des pas

La boucle “pour i de 1 à 6” déjà vue les TPs précédents augmente la valeur de i de 1 à chaque itération, lui faisant prendre successivement les valeurs 1,2,3,4,5 et 6.

Petite évolution, le boucle “pour i de 1 à 9 par pas de 3” augmente la valeur de i de 3 à chaque itération, lui faisant prendre les valeurs 1,3,6 et 9. La variable sur laquelle la boucle est effectué (ici “i”) doit toujours être de type “Entier”. Les bornes de début et de fin, ainsi que la valeur du pas peuvent être des nombres fixes (ici respectivement “1”, “9” et “3”) ou des variables de types entières. Exemples :

Problème	Algorithmie	C
Demander un nombre. Afficher tous les nombres pairs inférieurs à celui-ci	Entier n, i obtenir n pour i de 2 à n par pas de 2 afficher i finpour	<pre>int n, i; scanf("%d", &n); for (i=2 ; i<= n ; i= i+2) { printf("%d\n", i); }</pre>
Demander un nombre. Afficher tous les multiples de ce nombre inférieurs ou égaux à 100	Entier i, nb obtenir nb pour i de nb à 100 par pas de nb afficher i finpour	<pre>int i, nb; scanf("%d", &nb); for (i=nb ; i<=100 ; i = i+nb) { printf("%d\n", i); }</pre>

TP4-3

On désire écrire l'algorithme suivant :

Demandez un nombre à l'utilisateur. Calculez et affichez la somme de tous les multiples de ce nombre compris entre 1 et 1000 inclus.

- 1) Ecrire cet algorithme sur papier
- 2) Adapter cet algorithme en C
- 3) Comment, en utilisant les opérateurs mathématiques autorisés, savoir si un nombre fini par 8 ?
- 4) Adapter l'algorithme et le code C pour ne pas prendre en compte les nombres finissant par 8.

TP4-4

Demandez à l'utilisateur des nombres entiers jusqu'à ce que 2 saisies successives soit identiques.

- 1) Quelle structure de controle est la plus appropriée. Pourquoi ?
- 2) Ecrire cet algorithme sur papier
- 3) Adapter l'algorithme en C

TP4-5

Pour chaque nombre pair de 2 à 18 (inclus), demandez à l'utilisateur s'il veut connaitre la valeur de la puissance de 3 par ce nombre (en lui demandant de saisir le nombre 1). Le cas échéant, calculez et affichez cette valeur.

- 1) Commencer par écrire un algorithme sur papier pour calculer et afficher la valeur de 3^{12} (en utilisant une boucle)
- 2) Ecrire l'algorithme complet sur papier
- 3) Adapter cet algorithme en C

Rappels des TP's précédents

A faire s'ils n'avaient pas encore été faits, ou à finir à la maison.

TP2-11 (sur papier puis en C)

Ecrire un algorithme qui demande un nombre de notes à saisir, invite l'utilisateur à les saisir, puis calcule et affiche leur moyenne. Faites en sorte que les notes strictement inférieures à 10 aient un coefficient 3.

Indication : Si une note est inférieure à 10, il faut ajouter 3 fois la même note à la moyenne, en faisant évoluer le nombre total de notes.

TP2-12 (un peu modifié, sur papier puis en C)

Ecrire un algorithme qui demande un nombre entier à l'utilisateur, puis vérifie s'il est le multiple d'un nombre compris entre 2 et 101. Les divisions et les modulus sont interdits, les multiplications sont les seules opérations mathématiques autorisées.

Indication : il faut trouver 2 nombres a et b tel que $nb = a \times b$. a peut aller de 2 à 101, b de 1 à nb

TP3-4

Rechercher et afficher la plus grande valeur parmi un nombre indéterminé d'entiers saisis. Un nombre négatif arrêtera la saisie.

On affichera le nombre de saisies effectuées (le dernier nombre négatif ne comptant pas). On affichera le maximum à la fin de toutes les saisies, mais seulement s'il y en a eu au moins une.

Indication : il faut stocker dans une variable la plus grande valeur saisie jusqu'à présent

TP4-bonus : Mastermind

A faire si tout le reste est fini.

Ecrire en C le jeu suivant.

Générer un nombre aléatoire entre 0 et 9999 (cf TP3-Bonus. Utiliser le modulo pour borner le nombre aléatoire généré).

L'utilisateur devra trouver ce nombre. Pour chacune de ses saisies, vous réafficherez le nombre saisi sur 4 caractères avec sous chaque chiffre le symbole 'o' si le chiffre est bon, le symbole '-' si le chiffre est présent dans le nombre recherché et rien si le chiffre n'est pas présent.

Exemples, si le nombre recherché est 1997 :

Nombre saisi	1574	9	991	1997
Nombre affiché	1574	0009	0991	1997
Information	0 -	-	00 -	0000

Tester `printf("%04d", entier)` pour afficher les nombres au format demandé.

On affichera le nombre d'essais qui auront été nécessaires.