



1- Boucles définies - Suite de Fibonacci

En mathématiques, la **suite de Fibonacci** est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent.

Notée (F_n) , elle est donc définie par $F_0 = 0$, $F_1 = 1$ et $F_n = F_{n-1} + F_{n-2}$ pour $n \geq 2$

Exemple :

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	...	F_n
0	1	1	2	3	5	8	13		$F_{n-1} + F_{n-2}$

Dans un premier temps, on souhaite saisir une valeur n et afficher le nombre correspondant dans la suite de Fibonacci. On donne pour cela l'algorithme suivant :

Environnement :	En entrée : Le clavier En Sortie : L'écran	
Schéma algorithmique		Lexique des variables
<u>Début</u> lire : nbr Si nbr = 0 ou nbr = 1 Alors resultat = nbr Sinon ancienAvant = 0 ancien = 1 Pour indice allant de 2 à nbr resultat \leftarrow ancien + ancienAvant ancienAvant \leftarrow ancien ancien \leftarrow resultat FinPour FinSi Ecrire : resultat <u>Fin</u>		nbr _{entier} resultat _{entier} ancien _{entier} ancienAvant _{entier} indice _{entier}

1. En langage C sous NetBeans, réalisez un projet nommé **Fibonacci**. Dans le programme principal **main.c**, codez l'algorithme de Fibonacci. On remplacera juste **lire : nbr** par une simple affectation.
2. Après avoir vérifié avec les valeurs proposées en exemple et lorsque votre programme est opérationnel enregistrez le fichier **main.c** sous le nom **version2.c**.
3. Modifiez votre programme pour afficher la suite complète pour un nombre donné.

2- Boucles indéfinies

2.1- Plus grand commun diviseur

Soit l'algorithme suivant permettant de calculer le plus grand commun diviseur de deux nombres entiers

Environnement :	En entrée : Le clavier En Sortie : L'écran
Schéma algorithmique	Lexique des variables
<u>Début</u> 1) lire : dividende, diviseur <u>TantQue diviseur \neq 0</u> 2) memo \leftarrow diviseur 3) diviseur \leftarrow dividende modulo diviseur 4) dividende \leftarrow memo <u>FinTantQue</u> Ecrire « le PGCD est : » dividende <u>Fin</u>	dividende _{entier} diviseur _{entier} memo _{entier}

1. Vérifier par une simulation manuelle que le PGCD des nombres 21 et 15 est bien 3

Etape	dividende	diviseur	memo
1	21	15	?

2. Dans un nouveau projet nommé PGCD, codez, en langage C, l'algorithme du PGCD proposé.
3. Après avoir vérifié avec plusieurs valeurs, et lorsque votre programme est opérationnel enregistrez le fichier **main.c** sous le nom **version2.c**.
4. Modifiez le programme pour qu'il réalise des soustractions successives au lieu d'utiliser l'opérateur **modulo**. Le programme doit pouvoir fonctionner lorsque le dividende est supérieur au diviseur et inversement.

2.2- Calcul de PI à 10^{-4} près

1. Réalisez un nouveau projet nommé CalculPI.
2. Codez l'algorithme permettant de calculer la valeur de PI à 10^{-4} près sachant que la somme : $1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 \dots$ tend vers $\pi/4$, algorithme étudié dans le TD4 d'algorithmique. Pour le calcul de la valeur absolue, vous utiliserez la fonction **fabs()** de la librairie math.h. <https://www.gladir.com/CODER/C/fabs.htm>
3. Modifiez votre programme afin de déterminer le nombre de tours de boucle nécessaire pour obtenir cette valeur.
4. Calculez à nouveau la valeur de PI à 10^{-5} , combien de tours sont-ils nécessaires maintenant ?

2.3- Un drôle de fermier

Un fermier fait l'élevage de montons et de dindons et, au moment de payer ses impôts, il déclare curieusement : « j'ai dans mon levage 36 têtes et 100 pattes ! »

1. Dans un nouveau projet nommé *DroleDeFermier*, réalisez un programme en langage C permettant de déduire, si cela est possible, le nombre de moutons et de dindons que possède le fermier. En vous inspirant de la réflexion suivante :

Si moutons=36 alors dindons=0 donc pattes= 36x4 + 0x2 = 144 impossible
Si moutons=35 alors dindons=1 donc pattes= 35x4 + 1x2 = 142 impossible
Si moutons=34 alors dindons=2 donc pattes= 34x4 + 2x2 = 140 impossible
...
jusqu'à ce quepattes = 100.

Le nombre de pattes et le nombre de têtes seront initialisés au début du programme lors de la déclaration des variables.

Résultats attendus : Désolé, je ne trouve pas ! si aucune solution n'est possible
 Le fermier possède x moutons et y dindons si une solution est trouvée

3- Boucles imbriquées

En utilisant le formatage des nombres de la fonction **printf**, coder l'algorithme du programme permettant d'afficher la figure représentant un sablier proposée lors de l'exercice 4 du TD algorithme 4.

La taille de la matrice sera initialisée lors de la déclaration de la variable. Cette taille devant forcément être impaire, par une condition vous vérifierez que c'est effectivement le cas avant de fabriquer la figure, sinon un message informera l'utilisateur que la taille choisie ne convient pas.