

Exercice 1 CALCUL DU NOMBRE D'OR:

1. Écrire un programme permettant de calculer le nombre d'Or.
Celui-ci peut être obtenu à partir de la suite $u(n)$, dite de Fibonacci, définie par:

$$\begin{cases} u(0) = 1 \\ u(1) = 1 \\ \dots \\ u(n+1) = u(n) + u(n-1) \end{cases}$$

La suite $v(n) = \frac{u(n+1)}{u(n)}$ converge vers le nombre d'Or:

$$\frac{1 + \sqrt{5}}{2}.$$

2. Modifier ce programme de façon à demander à l'utilisateur d'entrer la précision relative ε avec laquelle la limite de la suite ($v(n)$) doit être calculée. Afficher le nombre de termes qu'il a été nécessaire de calculer dans la suite pour arriver à cette précision.

Exercice 2 LE CRIBLE D'ÉRATOSTHÈNE:

Écrire un programme permettant de déterminer les nombres premiers dans l'intervalle $[1, n]$ à l'aide du crible d'Ératosthène. Il consiste à former une table avec tous les entiers naturels compris entre 2 et n et à rayer (mise à zéro) les uns après les autres, les entiers qui ne sont pas premiers de la manière suivante: dès que l'on trouve un entier qui n'a pas encore été rayé, il est déclaré premier, et on raye tous les multiples de celui-ci. À la fin du procédé, les nombres non barrés sont des nombres premiers.

Par ailleurs, on sait que l'on peut réduire la recherche aux nombres de 2 à \sqrt{n} : en effet un nombre non premier supérieur à \sqrt{n} possède au moins un diviseur inférieur à \sqrt{n} et aura déjà été barré.

Exercice 3 Soit n un entier positif. S'il est pair, on le divise par 2. S'il est impair, on le multiplie par 3. Et on recommence. Finalement, on atteint la valeur 1 et la suite boucle sur les valeurs 1, 4, 2, 1.

1. Écrire un programme `suite` qui lit en entrée un nombre entier et affiche la suite de nombres obtenue jusqu'à la valeur 1.
2. Prévoir un compteur qui indique en combien de termes on atteint 1 et afficher sa valeur à la fin du calcul.

3. On veut à présent pouvoir tester plusieurs valeurs sans relancer à chaque fois le programme. À la fin d'un calcul, le programme doit désormais proposer à l'utilisateur de choisir une autre valeur de départ. Si c'est un entier positif, on repart pour un nouveau calcul, sinon le programme se termine.
4. Tester les valeurs 7, 106, 46, 3 et 0.
5. On veut évaluer le nombre de de termes calculés en fonction du premier terme de la suite, et stocker ces deux valeurs dans un fichier de façon à pouvoir dessiner la courbe correspondante (par exemple avec gnuplot)
 - ouvrir un fichier (formaté) au début du programme
 - à chaque calcul, écrire dans ce fichier la valeur du premier terme et le nombre de termes calculés.
 - À quel moment faut-il prévoir de fermer ce fichier pour que le programme se termine proprement? Le faire.
 - Écrire un script `lance_suite` qui crée un fichier d'entrée `in` avec les entiers de 2 à 1000 et qui exécute le programme `suite` pour ces valeurs. Dessiner (avec `gnuplot`) la courbe correspondante.

Pour dessiner la courbe avec `gnuplot`:

- taper `gnuplot`,
- taper `plot 'resu.dat' with lines`.

Voir par exemple:

- <http://www.gnuplot.info>: la home page de `gnuplot`
- <http://www.cicrp.jussieu.fr/phynum/cours/logi/logi2.html>: une introduction en français.