

TRAVAUX DIRIGES – Semaine n°2

Thèmes

- Algorithmes
- Structures de contrôle itératives et conditionnelles

Préparation

Les exercices sont à préparer (en pseudo-code ou en langage C).

Exercice 1. La factorielle d'un nombre entier naturel

Ecrivez en pseudo-code un algorithme qui permet de calculer la factorielle d'un entier naturel n . Vous utiliserez dans une première version une boucle *tant que*. Vérifiez que votre boucle s'arrête.

On rappelle que la factorielle de n (notée $n!$) est le produit des nombres entiers strictement positifs inférieurs ou égaux à n .

$$n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$$

Codez un programme qui permet de calculer la factorielle d'un nombre positif que vous initialiserez dans le code à l'aide d'une boucle *for*. Vérifiez l'arrêt de la boucle.

Donnez un plan de tests unitaires.

Exercice 2. Les diviseurs d'un nombre

Affichez tous les diviseurs d'un nombre n en les séparant par un espace.

On rappelle que le diviseur d d'un nombre n est tel que le reste de la division entière de n par d est nul.

Facultatif : Quelles sont les modifications à apporter au code pour afficher les diviseurs 3 par ligne. Indication : utilisez l'opérateur modulo.

Donnez un plan de tests unitaires.

Exercice 3. PGCD de deux nombres

Le PGCD (plus grand commun diviseur) de deux entiers naturels est le plus grand entier naturel qui les divise tous les deux.

Calculez le PGCD de deux nombres sachant que :

$$\text{PGCD}(a, 0) = a ;$$

$$\text{Si } b \neq 0 \text{ alors } \text{PGCD}(a, b) = \text{PGCD}(b, a \% b)$$

Donnez un plan de tests unitaires.

Exercice 4. Nombre premier

Un nombre n est premier s'il a deux diviseurs : 1 et n lui-même (1 n'est pas premier).

Déterminez si un nombre est premier en testant ses diviseurs potentiels et en remarquant que si un nombre n'est pas premier il a au moins un diviseur (autre que 1) plus petit ou égal à sa racine carrée.

Donnez un plan de tests unitaires.

Exercice 5. Eratosthène et les nombres premiers

Affichez les n premiers nombres premiers en utilisant l'algorithme dit du crible d'Eratosthène.

Ce dernier initialise un tableau *crible* de booléens à vrai (de l'indice 2 à l'indice n) puis teste chaque nombre i du tableau ainsi : si *crible*(i) est vrai alors i est premier et on barre (c.à.d. on marque à faux) tous les multiples de i . Il est inutile de barrer les multiples des nombres non premiers.

Remarquez par ailleurs que certains multiples de i ont déjà été testés s'ils sont aussi multiples d'un nombre plus petit que i . Il suffit donc de barrer les multiples de i à partir de $i*i$.

Donnez un plan de tests unitaires.