

TD1 : Algorithmique

V1.3.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Document en ligne : mickael.martin.nevot.free.fr

1 Introduction

« Au lieu de ce grand nombre de préceptes dont la logique est composée, je crus que j'aurois assez des quatre suivants, pourvu que je prisse une ferme et constante résolution de ne manquer pas une seule fois à les observer.

Le premier était de ne recevoir jamais aucune chose pour vraie que je ne la connusse évidemment être telle ; c'est-à-dire, d'éviter soigneusement la précipitation et la prévention, et de ne comprendre rien de plus en mes jugements que ce qui me présenterait si clairement et si distinctement à mon esprit, que je n'eusse aucune occasion de le mettre en doute.

Le seconde, de diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait, et qu'il serait requis pour les mieux résoudre.

Le troisième, de conduire par ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu comme par degrés jusque à la connaissance des plus composés, et supposant même de l'ordre entre ceux qui ne se précèdent point naturellement les uns les autres.

Et le dernier, de faire partout des dénombrements si entiers et des revues si générales, que je fusse assuré de ne rien omettre. »

— Descartes, *Discours de la Méthode*

2 Exercices

Écrivez en pseudo-code (uniquement) les algorithmes définis ci-dessous, selon la convention donnée en cours.

2.1 Somme

Affiche la somme de deux nombres entiers saisis par l'utilisateur.

2.2 Permutation

Échange les valeurs de deux réels saisis par l'utilisateur.

2.3 Tarifs

Demande à l'utilisateur de saisir le prix de base du billet et l'âge d'un passager, puis affiche le prix après réduction.

Contraintes

Gratuit pour les enfants de moins de 2 ans ; moitié prix pour les enfants de moins de 10 ans ; réduction de 10 % pour les personnes de moins de 27 ans et celles de plus de 70 ans.

2.4 Minimum

Affiche le minimum d'une suite de dix réels saisis par l'utilisateur.

2.5 Changement de base

Affiche la valeur en base b ($b \in \mathbb{N}$) (compris entre 2 et 9) d'un entier n ($n \in \mathbb{N}$) décimal (en base 10).

2.6 Factoriel

Affiche le résultat de la fonction mathématique factorielle appliquée à n ($n \in \mathbb{N}$).

Contrainte

Faire un algorithme **récuratif**.

Équation 1 – Factorielle

$$n! = \prod_{k=1}^n k = 1 \times 2 \times 3 \times 4 \times \dots \times (n-1) \times n$$

2.7 Conjecture de Syracuse

Applique la « célèbre » fonction de conjecture de Syracuse à un nombre et s'arrête lorsque le calcul devient égal à 1.

La suite de Syracuse de N ($N \in \mathbb{N}^*$) est définie par récurrence, de la manière suivante :

Équation 2 – Conjecture de Syracuse

$$\forall n \in \mathbb{N} : \begin{cases} u_0 = N \\ u_{n+1} = \begin{cases} \frac{u_n}{2}, & \text{si } u_n \text{ est pair} \\ \frac{3u_n + 1}{2}, & \text{si } u_n \text{ est impair} \end{cases} \end{cases}$$

Faites plusieurs essais en changeant la valeur de test. Voyez-vous un problème algorithmique ?

2.8 Monnayeur

Affiche le nombre de chaque billets/pièces nécessaires pour composer une somme donnée en euros

sans centimes (par exemple 1949), grâce à un algorithme glouton (le nombre de billets/pièces ne sera donc pas forcément minimal).

2.9 Test de divisibilité par 11

Affiche le résultat du test de divisibilité suivant : en ajoutant et soustrayant en alternance les chiffres d'un nombre (en commençant par n'importe quelle extrémité) et ce, de manière itérative jusqu'à ce que la valeur absolue du résultat soit strictement inférieure à 11, nous pouvons affirmer que le nombre est divisible par 11 si le résultat est égal à 0 ou qu'il ne l'est pas sinon.

Exemples de nombres divisibles par 11 :

- 121 (+ 1 - 2 + 1 = 0) ;
- 9251 (-9 + 2 - 5 + 1 = - 11 [+1 - 1 = 0]).

2.10 Recherche dichotomique dans un tableau ordonné

Pour un tableau t de 100 nombres entiers deux à deux distincts rangés par ordre croissant et un nombre v , détermine soit le rang de v dans t , soit le fait que v ne figure pas dans t .

2.11 Tri par insertion

Trie par la méthode suivante (soit un tableau t de quinze nombres entiers) ; à l'étape i ($i \in \mathbb{N}^*$) :

- on suppose que les i premiers éléments de t sont triés ;
- on considère l'élément $t[i]$:
 - o l'insérer parmi $t[0], \dots, t[i - 1]$ à la bonne place (en décalant progressivement tous les éléments vers la droite) ;
 - o incrémenter i .