

Algorithmique et programmation procédurale - TD No 2

Fonctions et procédures

Exercice 1. Écrire une procédure qui permet de trouver la résolution d'une équation bicarrée ($ax^4 + bx^2 + c = 0$).

Exercice 2. Écrire une fonction booléenne qui permet de tester si un entier donné est premier ou non.

Exercice 3. Écrire un programme qui permet d'afficher tous les nombres premiers inférieurs à un entier donné.

Exercice 4. Écrire une fonction qui teste si un nombre est parfait ou non. Un **nombre parfait** est un nombre naturel non nul qui est égal à la somme de ses diviseurs stricts (par exemple : le premier nombre parfait est 6, car 1, 2, et 3 sont les diviseurs stricts de 6 et $1 + 2 + 3 = 6$).

Exercice 5. **Cryptographie 1**

Un des plus anciens systèmes de cryptographie (aisément déchiffrable) consiste à décaler les lettres d'un message pour le rendre illisible. Ainsi, les A deviennent des B, les B des C, etc. Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui la code selon ce principe, puis affiche le résultat à l'écran.

Exercice 6. **Cryptographie 2 - le chiffre de César**

Une amélioration du principe précédent consiste à opérer avec un décalage non de 1, mais d'un nombre quelconque de lettres. Ainsi, par exemple, si l'on choisit un décalage de 12, les A deviennent des M, les B des N, etc.

Réalisez un algorithme sur le même principe que le précédent, mais qui demande en plus quel est le décalage à utiliser.