

## TD N°1 : LES SOUS PROGRAMMES

### Fonction et procédures

**Exercice 1 :** Ecrire une procédure Echange1 qui inverse le contenu de deux valeurs entières passées en argument

**Exercice 2 :** Ecrire l'Algorithme qui affiche le maximum et le minimum de deux entiers saisis au clavier en organisant le travail selon la politique suivante :

- 1) Procédure Min de 2 entiers et procédure Max de 2 entiers.
- 2) Procédure MinMax de 2 entiers, qui appelle les procédures Min et Max.
- 3) Algorithme Principale qui lit 2 entiers, appelle MinMax et affiche les résultats.

**Exercice 3 :** Écrire les procédures suivantes

- 1) Procédure Echange2 sur 2 réels a; b qui inverse éventuellement a et b, pour que l'état de sortie soit  $a \leq b$ . Ecrire un l'Algorithme d'appel.
- 2) Procédure Echange3 sur 3 réels a; b; c qui appelle Echange2. L'état de sortie est  $a \leq b \leq c$ .  
Ecrire l'Algorithme principale.

**Exercice 4 :** Ecrire une fonction *puissance* qui calcule  $a^b = a \times a \times a \dots \times a$  (b fois); a et b étant des entiers positifs.

**Exercice 5 :** Sachant qu'un nombre premier est un nombre qui n'accepte aucun diviseur excepté 1 et lui-même.

1. Ecrire une fonction *premier* qui indique si un nombre entier N est premier ou pas.
2. Construire l'algorithme qui nous donne les N premiers nombres premiers.

**Exercice 6 :** A partir d'un nombre entier N on voudrait obtenir deux autres nombres N1 et N2. Le premier (N1) sera constitué par les chiffres pairs de N et le second (N2) par les chiffres impairs.

Exemples :

N = 25461327	N1 = 2462	N2 = 5137
N = 42613786	N1 = 42686	N2 = 137
N = 240682	N1 = 240682	N2 = 0

1. Ecrire un algorithme qui permet de retourner les nombre N1 et N2 à partir d'un nombre N.

**N.B :** La solution doit comporter au moins une fonction et une procédure.

**Exercice 7 :** Soit A un temps en secondes.

1. Ecrire une **procédure** qui convertit A en heures, minutes et secondes.

**Exercice 8 :** Etant donné deux nombres N1 et N2.

1. Ecrire une **fonction** qui permet de fusionner les chiffres des deux entier N1 et N2 de manière alternée.

Exemples : N1 = 381                      N2 = 946                      N = 398416

**N.B :** La solution doit comporter au moins deux fonctions.

**Exercice 9 :** (Examen rattrapage Init.Algo S1 2015/2016)

1. Ecrire une procédure **Insert\_Chiff (c, p, N)** qui permet d'insérer un chiffre c dans le nombre N à la position p (le résultat est N lui-même).

Exemple : Insert\_Chiff(5, 3, N) avec N = 14721 donné le résultat N = 147521.

**Exercice 10 :** Ecrire une fonction permettant de convertir un nombre N écrit sous la base x ( $x \leq 10$ ) à la base 10.

Exemple :  $(7643)_8 = (4003)_{10}$

1. Généraliser la solution précédente pour convertir un nombre écrit sous la base x à un nombre écrit sous la base y ( $x \leq 10, y \leq 10$ )

**Exercice 11 : (Examen rattrapage Init.Algo S1 2013/2014)**

1. Ecrire une fonction **SommeDiv(n)** qui retourne la somme des diviseurs propres (sauf le nombre lui-même) d'un nombre entier N.
2. Ecrire un algorithme qui permet d'afficher si un nombre entier X est abondant, parfait ou déficient en utilisant la fonction **SommeDiv(n)**.

Lorsque la somme des diviseurs propres d'un nombre lui est supérieure, le nombre est dit *abondant*. Lorsque la somme lui est égale, le nombre est dit *parfait*. Lorsque la somme lui est inférieure, le nombre est dit *déficient*.

**Exercice 12 :** Deux entiers naturels strictement positifs m et n sont dits nombre amis si et seulement si :

- la somme des diviseurs de m sauf lui-même est égale à n
- et la somme des diviseurs de n sauf lui-même est égale à m.

Exemple : 220 et 284 sont deux nombres amis.

1. Écrire un algorithme qui permet d'afficher si deux entiers naturels donnés m et n sont amis ou non.

**N.B :** La solution doit comporter au moins une fonction.

**Exercice 13 :**

1. Ecrire une fonction permettant de vérifier si un mot est palindrome ou non.

Exemple : mots palindromes : RADAR, ICI, ELLE, ....

3. Ecrire un algorithme permettant d'afficher tous les mots palindromes dans un texte donné.

**Exercice 14 :**

1. Ecrire un algorithme qui permet de saisir deux mots non vides MOT1 et MOT2 puis de déterminer si MOT2 est une anagramme de MOT1.

Une anagramme est un mot obtenu par transposition des lettres d'un autre mot (par exemple chien, chine sont des anagrammes du mot niche).

**N.B :** La solution doit comporter au moins deux fonctions.

**Exercice 15: (Examen PSD S2 2013/2014)**

Soit T un tableau à une dimension de N valeurs entières ( $N \leq 100$ )

1. Ecrire une fonction **Tab\_Rech (T, N, val)** qui permet de rechercher une valeur entière **val** dans le tableau **T** et retourne son indice dans le tableau **T** si elle existe (retourne l'indice de la première occurrence) sinon, retourne une valeur négative.
2. Ecrire une procédure **Tab\_sans\_double (T1, N, T2)**, en utilisant la fonction précédente, qui permet de construire un tableau **T2** qui ne contient qu'une seule occurrence de chaque nombre dans **T1** de N éléments.