

# FACULTÉ DES SCIENCES DE GABÈS

## DÉPARTEMENT INFORMATIQUE

*Algorithmique et Structures de Données1*

*TD3: Les procédures et les fonctions*

Sections: LGSSI1/LIRIS1

A.U: 2023/2024

### Exercice 1

Soit l'algorithme **ProduitIntervalle1** qui permet de calculer et d'afficher le produit **P** des entiers naturels successifs entre une valeur minimale  $B_i$  et une valeur maximale  $B_s$ .

$$P = \prod_{k=B_i}^{B_s} i \quad (1)$$

**Algorithme** ProduitIntervalle1

**Var**

Bi, Bs, P : **Entier**

**Début**

```
SaisieBornes(Bi, Bs)
CalculeProduit(Bi, Bs, P)
Ecrire("Borne inférieure Bi=", Bi)
Ecrire("Borne supérieure Bs=", Bs)
Ecrire("Le produit des entiers compris entre", Bi, "et", Bs, "est", P )
```

**Fin**

1. Ecrire la procédure **SaisieBornes** qui demande à l'utilisateur de fournir les deux valeurs Bi et Bs. Cette procédure doit garantir que Bi soit strictement positif et strictement inférieure à Bs.
2. Ecrire la procédure **CalculeProduit** à 3 paramètres Bi, Bs et P.
3. Nous voulons convertir les trois dernières lignes de l'algorithme **ProduitIntervalle1** en un sous programme appelé Affichage.
  - (a) Ce sous programme doit il être une procédure ou une fonction ? Argumenter brièvement votre réponse.
  - (b) Ecrire ce sous programme.
  - (c) Réécrire l'algorithme **ProduitIntervalle1** en utilisant les procédures et fonctions des questions 1, 2 et 3-b. Nous appelons cette version de l'algorithme **ProduitIntervalle2**
4. Nous voulons convertir la procédure **CalculeProduit** de l'algorithme **ProduitIntervalle2** en une fonction appelée **CalcProd**.
  - (a) Ecrire cette nouvelle fonction.
  - (b) Ecrire la nouvelle version de l'algorithme en utilisant les sous programmes des questions 1, 4-a et 3. Nous appelons cette version de l'algorithme **ProduitIntervalle3**.

### Exercice 2

1. Ecrire une fonction permettant de vérifier si un nombre entier n est premier ou non.
2. Ecrire une autre fonction permettant de vérifier si un nombre entier positif n est une puissance de 2 ou non.
3. Ecrire un algorithme utilisant ces deux fonctions pour dire si un nombre entré au clavier est premier et/ou une puissance de 2.

## Exercice 3

1. Ecrire une fonction MIN et une fonction MAX qui détermine le minimum et le maximum de deux nombres réels.
2. Ecrire un algorithme qui fait appel aux fonctions MIN et MAX pour déterminer le minimum et le maximum de quatre nombres entrés à partir du clavier.

## Exercice 4

Deux nombres sont dits amis, si et seulement si, la somme des chiffres du premier est égale à la somme des chiffres du second. Deux nombres sont dits cousins si le produit des chiffres non nuls du premier est égal au produit des chiffres non nuls du second.

**Exemples :**

1221 et 140 sont cousins puisque :  $1 * 2 * 2 * 1 = 1 * 4$

6002 et 143 sont cousins puisque :  $6 * 2 = 1 * 4 * 3$

105 et 150 sont cousins puisque :  $1 * 5 = 1 * 5$

On vous demande d'écrire les modules suivants :

1. Une procédure **SAISIE** permettant de saisir un entier strictement positif.
2. Une fonction **SOMME** permettant de calculer la somme des chiffres d'un entier.
3. Une fonction **PRODUIT** permettant de calculer et retourner le produit des chiffres non nuls d'un entier.
4. Ecrire un algorithme qui fait appel aux procédures ci-dessus pour saisir deux entiers A et B strictement positifs et d'afficher si A et B sont cousins ou non.

## Exercice 5

1. Ecrire une fonction INVERSION qui permet de retourner un nombre comprenant les mêmes chiffres que le nombre passé en paramètre mais dans l'ordre inverse.

**Exemple :** Si N = 1236 est passé en paramètre alors la fonction retourne 6321.

2. Ecrire une fonction PALINDROME qui vérifie et retourne :
  - (a) 1 si un nombre N est un palindrome et ceci en faisant appel à la fonction de la question précédente. Un palindrome est un nombre symétrique, par exemple 12321, 342243 et 77 sont des palindromes.
  - (b) 0 si non.
3. Ecrire un algorithme qui fait appel aux fonctions ci-dessus pour :
  - (a) saisir un entier positif N,
  - (b) et d'afficher si N est palindrome ou non.

## Exercice 6

Tout nombre positif composé de deux chiffres ab (tel que  $a \neq b$ ) possède une liste appelée "Liste vers 9".

Le principe est le suivant :

On calcule la différence (en valeur absolue) entre ab et son symétrique ba ; le résultat trouvé subit le même traitement ; on répète ce processus jusqu'à obtenir une différence = 9. L'ensemble constitué par le nombre initial et les résultats des différences est appelé "liste vers 9".

**Exemple :** Si X = 18, la liste vers 9 de 18 est la suivante : 18 63 27 45 9

On vous demande d'écrire les modules suivants :

1. Une fonction SAISIE permettant de saisir un entier positif X composé de deux chiffres obligatoirement différents.
2. Une procédure LISTE\_9 permettant de chercher et d'afficher la liste vers 9 de X.
3. Ecrire un algorithme qui fait appel aux procédures ci-dessus pour saisir un entier positif X composé de deux chiffres obligatoirement différents et d'afficher sa liste vers 9.

## Exercice 7

1. Ecrire une fonction **SOMME** qui retourne la somme suivante :

$$S = \sum_{j=1}^i 1/j \quad (2)$$

- 2- Ecrire une fonction **PRODUIT** qui retourne le produit P définie ci-après :

$$P = \prod_{i=1}^n \left( \sum_{j=1}^i 1/j \right) \quad (3)$$

- 3- Ecrire un algorithme qui fait appel aux fonctions ci-dessus pour afficher le produit P. (où n est un entier naturel saisi au clavier  $n \geq 0$ ).

## Exercice 8

On se propose de calculer la racine carrée d'un réel positif X donné en utilisant la suite suivante :

$$\begin{cases} U_0 = (1 + X)/2 \\ U_{n+1} = (U_n + X/U_n)/2 \end{cases}$$

On vous demande d'écrire les fonctions suivantes :

1. Une procédure **SAISIE** permettant de saisir un réel positif.
2. Une fonction **RACINE** permettant de calculer et retourner la racine carrée de X. Calculer les premiers termes jusqu'à ce que la différence entre deux termes successifs devient  $\leq 10^{-4}$ , le dernier terme calculé est une valeur approchée de racine de X.
3. Ecrire un algorithme qui fait appel aux procédures ci-dessus pour afficher la racine carrée d'un réel.

## Exercice 9

On considère la formule récurrente suivante :

$$\begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} (n > 1) \end{cases}$$

1. Ecrire une fonction permettant de calculer le terme  $U_n$  pour n donné.
2. Ecrire un algorithme qui utilise la fonction définie dans la question 1 et qui affiche le  $n^{\text{ème}}$  terme de cette suite, où n est un entier saisi à partir du clavier.