

Série d'exercices N° 7

RECURSIVITE

---

**1. Calcul de la factorielle :**

Calcul de la factorielle d'un nombre N donné par l'utilisateur

- a. version itérative
- b. version récursive

**2. Calcul de  $C_n^p$**

Le nombre de combinaisons de n éléments pris p à p ( $0 \leq p < n$ ) peut être calculé en se basant sur les formules récursives suivantes :

$$C_n^0 = 1$$

$$C_n^n = 1$$

$$C_n^p = C_{n-1}^{p-1} + C_{n-1}^p$$

- a. Ecrire une fonction récursive qui calcule  $C_n^p$ .
- b. Dessiner l'arbre d'exécution pour un appel à cette fonction avec  $n = 3$  et  $p = 2$ .

**3. Nombre de Fibonacci :  $F_n = F_{n-1} + F_{n-2}$**

Calcule le Nième nombre de Fibonacci, N étant donné en paramètre

- a. version récursive
- b. version itérative

**4. Palindrome**

Un palindrome est un mot, une phrase, un nombre qui, si l'on ne tient compte ni des espaces ou apostrophes ni des signes de ponctuation, peut être lu de droite à gauche ou de gauche à droite en gardant la même signification. (Exemple : eve, laval, un roc cornu, 239932 sont des palindromes).

Ecrire un programme permettant de déterminer si un mot donné est ou non un palindrome. On supposera que les mots saisis ne comportent que des caractères significatifs (pas de blanc ou de signe de ponctuation).

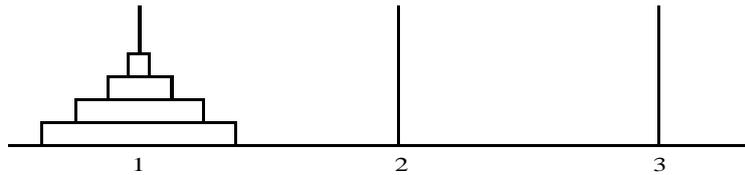
**5. Somme des éléments d'un tableau**

On dispose d'un tableau non contraint dont la déclaration est la suivante :

```
type Tab is array(natural range <>) of float ;
```

Ecrire une fonction récursive qui permet de calculer la somme des éléments d'un tableau de ce type.

## 6. Tour de Hanoi



Ecrire une procédure qui transfère un nombre de disques N du piquet Origine 1 sur le piquet Destination 3 en utilisant le piquet Auxiliaire 2 comme piquet intermédiaire

## 7. Recherche de la sortie dans un labyrinthe

On considère un labyrinthe modélisé par un tableau à 2 dimensions contenant des caractères. On adopte les conventions suivantes ( cf exemple ) :

- le caractère 'H' représente un mur ;
- le caractère '•' représente un passage.

Il s'agit de trouver, à partir d'une position donnée, un chemin quelconque vers une sortie (pas nécessairement le chemin le plus court, ni le plus direct). On ne considère que des déplacements horizontaux ou verticaux (pas en diagonale).

Ecrire un sous-programme qui, pour un labyrinthe et une position de départ donnés, indique s'il existe un chemin vers une sortie. Si un tel chemin existe, le chemin trouvé doit être matérialisé par des caractères \* .

### Exemple de labyrinthe :

```

H  H  H  H  H  H  H
H  •  •  H  •  •  H
H  H  H  H  •  H  H
H  •  •  •  •  H  H
H  •  •  H  •  H  H
H  •  •  •  •  •  H
H  H  H  H  H  •  H
    
```

### Exécution pour différentes positions de départ :

**Position de départ :**  
*ligne 2 ; colonne 5*

**Sortie trouvée !**

**Position de départ :**  
*ligne 2 ; colonne 3*

**Il n'existe pas de chemin vers la sortie!**

```

H  H  H  H  H  H  H
H  •  •  H  *  •  H
H  H  H  H  *  H  H
H  *  *  *  *  H  H
H  *  •  H  •  H  H
H  *  *  *  *  *  H
H  H  H  H  H  *  H
    
```

Chemin vers la sortie

## 8. Calcul de la factorielle : devoir à rendre

Il s'agit d'écrire un programme permettant d'évaluer la fonction exponentielle(x) en utilisant la formule suivante (développement en série).

$$\text{Exp}(x) = 1 + x^1/1! + x^2/2! + \dots + x^n/n! \quad \text{Où } n! \text{ représente la factorielle de } n$$

Calcul de la factorielle :

$$\begin{array}{ll} n! = 1 & \text{si } n = 0 \\ n! = n * (n-1)! & \text{si } n > 0 \end{array}$$

### Remarque :

Si le nombre est trop grand, il y a **Constraint\_Error** car les résultats dépassent **Integer'Last**.

Si le nombre est beaucoup plus grand, il y a **Storage\_Error** à cause de la pile des appels.

Par test, on détermine que 13 est déjà trop grand pour qu'on puisse calculer sa factorielle.

Cahier des charges :

1 Dans le corps de la procédure principale, uniquement des appels de procédures et fonctions

2 Arrêter le calcul itératif sitôt que le terme  $|x^n/n!|$  de la série est inférieur à  $10^{-5}$

3 Le calcul de n! doit être fait à l'aide d'une fonction récursive

4 comparer votre résultat à celui fourni par la fonction exp(x). Pour cela, il est nécessaire d'inclure le package Ada.Numerics.Elementary\_Functions;

Implémentez les procédures et fonctions suivantes :

**procedure** Calcul\_Expo ;

**procedure** Presentation ;

**function** Menu **return** T\_Menu;

**function** Factorielle(N : Natural) **return** Natural;

**function** Exponentielle(X : Float) **return** Float;