

TD 03

Exercice 01

Suite Arithmétique $S = 3 + 7 + 11 + \dots + (4 * n - 1)$

- Écrire une fonction `CalculerTerme(k : Entier) : Entier` qui retourne la valeur du k-ième terme de la suite, définie par la formule $4k-1$.
- Écrire une fonction `SommerSuite(n : Entier) : Entier` qui utilise la fonction précédente pour calculer la somme totale des n termes.
- Écrire le programme principal qui lit la valeur de n au clavier, appelle la fonction `SommerSuite` et affiche le résultat final.

Exercice 02

$$e \approx \sum_{k=0}^n \frac{1}{k!}$$

- Écrire une fonction `Factorielle(k : Entier) : Entier` qui retourne le produit des entiers de 1 à k.
- Écrire une fonction `TermeEuler(k : Entier) : Réel` qui retourne la valeur de $1/k!$ en faisant appel à la fonction `Factorielle`.
- Dans le programme principal, lire la précision n demandée par l'utilisateur et réaliser une boucle qui cumule les résultats de `TermeEuler` pour afficher l'approximation de e.

Exercice 03

n utilisant le développement limité suivant, calculez une approximation de la fonction $\sin(x)$:

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

- Écrire une fonction `Puissance(base : Réel, exp : Entier) : Réel` qui calcule base^{exp} .
- Réutiliser ou réécrire la fonction `Factorielle(n : Entier)`.
- Écrire une fonction `TermeSinus(x : Réel, k : Entier) : Réel` qui calcule le k-ième terme du développement
- Concevoir le programme principal qui demande à l'utilisateur la valeur de x et le nombre de termes n, puis assemble ces composants pour afficher $\sin(x)$

Exercice 04

Calculez une approximation de la fonction exponentielle en utilisant le développement limité suivant :

$$e^x \approx 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

- Ecrire une fonction `Puissance(x : Réel, k : Entier) : Réel` qui calcule x à la puissance de k .
- Écrire une fonction `CalculerTermeExp(x : Réel, k : Entier) : Réel` qui calcule la fraction $(x^k/k!)$ en utilisant les fonctions `Puissance` et `Factorielle` définies précédemment.
- Écrire une procédure `AfficherApproximation(x : Réel, n : Entier)` qui calcule la somme des termes de 0 à n et affiche un message structuré présentant le résultat.
- Dans le programme principal, réaliser la lecture sécurisée de x et n , puis appeler la procédure `AfficherApproximation`.

Exercice 05

Écrire un programme qui calcule le produit :

$$P = \prod_{k=1}^n \left(1 + \frac{1}{k^2}\right)$$

- Écrire une fonction `CalculerTerme(k : Entier) : Réel` qui retourne la valeur de l'expression $(1 + 1/k^2)$.
- Écrire une fonction `CalculerProduitTotal(n : Entier) : Réel` qui réalise le produit cumulé des n termes en utilisant la fonction `CalculerTerme`.
- Écrire le programme principal qui lit l'entier n , appelle `CalculerProduitTotal` et affiche le résultat P .

Exercice 06

Calculer l'approximation de la valeur de π en utilisant la formule de Leibniz :

$$\pi \approx 4 \sum_{k=0}^n (-1)^k \frac{1}{2k+1}$$

- Écrire une fonction `PuissanceMoinsUn(k : Entier) : Entier` qui retourne 1 si k est pair et -1 si k est impair (pour simuler $(-1)^k$ à la puissance k).
- Écrire une fonction `TermeLeibniz(k : Entier) : Réel` qui retourne la valeur de $(-1)^k * 1/(2k+1)$.
- Écrire une fonction `ApproximationPi(n : Entier) : Réel` qui calcule la somme des n termes et la multiplie par 4.
- Écrire le programme principal pour lire la précision n et afficher la valeur de π obtenue.