

Algorithmique et structures de données – ASD1
Travaux Pratiques – Série d'exercices 04

Exercice 1

1. Écrire une fonction en langage C qui vérifie si un entier positif est premier ou non.
2. Écrire une procédure en langage C qui vérifie si un nombre est premier ou non.
3. Écrire un programme principal en langage C qui affiche la liste des nombres premiers compris entre deux valeurs positives entrées par l'utilisateur.

Exercice 2

1. Écrire une procédure en langage C qui permet de résoudre une équation du second degré.
2. Écrire une fonction en langage C qui assure le même traitement.

Exercice 3

1. Écrire une fonction en langage C qui vérifie si un nombre est pair ou non.
2. Écrire un programme principal en langage C qui affiche le nombre et le pourcentage de nombres impairs dans une suite de nombres entiers positifs introduite par l'utilisateur et qui termine par 0.

Exercice 4

1. Écrire une fonction récursive en langage C qui calcule le terme de rang n de la suite définie par:
$$\begin{cases} U_0 = 56 \\ U_n = U_{n-1} - 0.8715 \end{cases}$$
2. Écrire un programme principal qui utilise la fonction de la question 1 pour afficher les M premiers termes de cette suite; M est entré par l'utilisateur.
3. Écrire un programme principal qui affiche les termes strictement positifs de cette suite.

Exercice 5

1. En langage C, écrire une fonction itérative factIter qui calcule la factorielle d'un entier positif.
2. En langage C, écrire une fonction récursive factRec qui calcule la factorielle d'un entier positif.
3. Écrire un programme principal qui Lit deux valeurs entières positives A et B , et ensuite calcule et affiche le résultat: $C = A!(A - B)!/B!$.
4. En langage C, écrire une fonction Puissance qui permet de calculer B^P , l'élevation d'un nombre B non nul à la puissance P , tel que P est un nombre **entier**.
5. Utiliser la fonction factIter (ou factRec) avec la fonction Puissance dans un programme principal pour afficher le résultat de la formule:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^9}{9!}$$

6. Utiliser la fonction factIter (ou factRec) avec la fonction Puissance dans un programme principal pour afficher le résultat de la formule:

$$Formule = (1!)^9 + (2!)^8 + (3!)^7 + (4!)^6 + \dots + (9!)^1$$

Exercice 6

1. Écrire une fonction itérative en langage C qui affiche le résultat du calcul suivant:

$$f(M) = \sum_{n=0}^M \frac{(-1)^n}{2n}$$

2. Écrire une fonction réursive en langage C qui assure le même traitement.

Exercice 7

Le nombre D_n de dérangements de n éléments vérifie l'étonnante récurrence suivante :

$$\begin{aligned} D_1 &= 0 \\ D_n &= n * D_{n-1} + (-1)^n \end{aligned}$$

1. En langage C, écrire une fonction réursive DérangementRec qui calcule D_n .
2. En langage C, écrire une fonction itérative DérangementIter qui calcule D_n .
3. Exécuter la fonction réursive, DérangementRec, pour $n = 4$.

Exercice 8

1. En langage C, écrire une fonction réursive compRec qui détermine le maximum de deux entiers positifs donnés, mais sans utiliser l'opération de comparaison.
2. En langage C, écrire un programme principal qui exécute la fonction compRec pour les deux nombres 3 et 7.