
Fiche de TD : Base du langage C

Exercice - 1 *Premier programme*

Soit la fonction mathématique f définie par $f(x) = (2x + 3)(3x^2 + 2)$

Q-1 : Écrire un programme C qui calcule l'image d'un nombre saisi au clavier.

Q-2 : Une approximation de la dérivée f' de la fonction f est donnée en chaque point x , pour h assez petit (proche de 0) par :

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h}$$

Écrire un programme C qui calcule et affiche une approximation de la dérivée de f en un point x entré au clavier. On pourra faire saisir le paramètre h au clavier.

Exercice - 2 *Type de données simples*

Écrire un programme qui affiche le chiffre des dizaines d'un nombre saisi au clavier. Même question pour les centaines.

Exercice - 3 *Entrées - sorties : `stdio.h`*

Écrire un programme qui :

1. lit deux entiers n_1, n_2 au clavier ;
2. affiche la partie entière de leur quotient ;
3. affiche la partie fractionnaire $frac$ de leur quotient ;
4. lit un nombre réel l au clavier ;
5. calcule la partie entière du produit de l par $frac$ modulo 256, puis convertit la résultat en caractère ;
6. affiche le caractère obtenu.

Tester le programme écrit avec $n_1 = 1321, n_2 = 500, l = 500$.

Exercice - 4 *Sélections : `if-then`, `if-then-else` et `switch`*

Q-1 : Une entreprise X vend deux types de produits. Les produits de type A qui donnent lieu à une TVA à 5.5%, et les produits de type B , qui donnent lieu à une TVA à 19.6%. Écrire un programme qui lit au clavier le prix hors taxe d'un produit, saisit au clavier le type du produit et affiche le taux de TVA et le prix TTC du produit.

Q-2 : Écrire un programme qui lit deux nombres entiers a et b et donne le choix à l'utilisateur :

1. de savoir si la somme $a + b$ est paire ;
2. de savoir si le produit ab est pair ;
3. de connaître le signe de la somme $a + b$;
4. de connaître le signe du produit ab .

Exercice - 5 Structuration d'un programme en C

Soit la fonction mathématique f définie par $f(x) = \frac{(2x^2+3)(x^2-1)}{3x^2+1}$.

Q-1 : Écrire une fonction C qui retourne la valeur de $f(x)$ pour un point x passé en paramètre.

Q-2 : Une approximation de la dérivée f' de la fonction f est donnée en chaque point x , pour h assez petit (proche de 0) par :

$$f'(x) \simeq \frac{f(x+h) - f(x-h)}{2h}$$

Écrire une fonction C qui calcule une approximation de la dérivée f' de f en un point x entré au clavier. On passera la valeur de h en paramètre de la fonction.

Q-3 : La dérivée seconde de f est la dérivée de la dérivée. Écrire une fonction qui calcule une approximation de la dérivée seconde de f'' de f en un point x entré au clavier. On passera la valeur de h en paramètre de la fonction.

Q-4 : Écrire une fonction C qui détermine le signe de la dérivée seconde de f en fonction de x . On pourra faire un programme principal qui lit x entré au clavier et affiche le résultat.

Q-5 : Écrire une fonction C qui donne le choix à l'utilisateur d'afficher la valeur de la fonction f , de sa dérivée première ou de sa dérivée seconde en un point x lu au clavier.

Exercice - 6 Structures

Définir une structure `NombreRationnel` permettant de coder un nombre rationnel, avec numérateur et dénominateur. On écrira des fonctions de saisie, d'affichage, de multiplication et d'addition de deux rationnels. Pour l'addition, pour simplifier, on ne cherchera pas nécessairement le plus petit dénominateur commun.

Exercice - 7 Boucle While et boucle For

Q-1 : Écrire une fonction qui calcule la factorielle $n!$ d'un entier n passé en paramètre. (*Rappel : $n! = 1 \times 2 \times \dots \times n$*). On en fournira deux versions, une avec la boucle `while` et l'autre avec la boucle `for`).

Q-2 : La suite de Fibonacci est définie par récurrence comme suit :

$$u_0 = 1, u_1 = 1, \text{ et } u_n = u_{n-1} + u_{n-2}, \text{ pour } n \geq 2.$$

Écrire une fonction qui calcule le n ième terme de la suite de Fibonacci. On utilisera la boucle `for`.

Exercice - 8 Tableaux

- Q-1** : Écrire une fonction qui prend en paramètre un tableau d'entier et calcule la somme de ses éléments.
- Q-2** : Écrire une fonction qui prend en paramètre un tableau de taille `NB_MAX`, son nombre d'éléments $n < \text{NB_MAX}$, un entier $i \leq n$, un entier m . La fonction doit insérer l'élément m en position i dans le tableau (sans supprimer d'élément).

Exercice - 9 Fichier texte

Une série statistique donne, pour chaque poids : $1kg, 2kg, 3kg, \dots, 200kg$, le nombre de personnes de la population qui pèse ce poids. Ces données sont stockées dans un fichier. La i ème ligne du fichier contient le nombre de personnes qui pèsent ikg .

n_1
 n_2
 \dots
 n_{200}

- Q-1** : Écrire un programme qui charge ces données en mémoire centrale et calcule le poids moyen dans la population.
- Q-2** : Écrire un programme qui calcule le poids moyen dans la population sans charger les données en mémoire.
- Q-3** : Reprendre les questions précédentes en mettant le résultat du calcul en fin du fichier lu.

Exercice - 10 Adresses, pointeurs et passage par adresse

- Q-1** : Écrire une fonction qui initialise deux entiers et un réel à 0. Écrire le programme principal qui appelle cette fonction.
- Q-2** : Soit une structure `Point` contenant deux champs x et y de type `float`.
- Q-2-1** : Écrire une fonction qui échange deux structures de type `Point` passées par adresse.
- Q-2-2** : Écrire le programme principal qui saisit deux structures de type `Point` dans des variables, échange le contenu de ces variables en appelant la fonction et affiche le nouveau contenu des variables.

Exercice - 11 Allocation dynamique

- Q-1** : Écrire une fonction qui lit un entier n au clavier, alloue un tableau de n entiers initialisés à 0, et retourne n et le tableau. Écrire le programme principal qui teste cette fonction. On n'oubliera pas de libérer l'espace mémoire alloué.
- Q-2** : On se propose de réaliser une fonction de chargement en mémoire centrale sous forme de tableau de `float` d'un fichier texte. Le format du fichier est le suivant :

- la première ligne du fichier contient le nombre d'éléments du tableau ;
- les lignes suivantes contiennent chacune un nombre réel.

n
 f_1
 f_2
 \dots
 f_n

Q-2-1 : Réaliser une fonction de chargement dans un tableau dont la taille mémoire correspond exactement au nombre d'éléments du fichier.

Q-2-2 : Réaliser une fonction d'affichage du tableau.

Q-2-3 : Écrire le programme principal qui charge le fichier et affiche le tableau.

Q-3 : Soit la suite u_n définie par :

$$u_0 = 1, \text{ et } u_{n+1} = 3u_n^2 + 2u_n + 1, \text{ pour } n \geq 0.$$

Q-3-1 : Écrire une fonction qui prend en paramètre un entier n et qui retourne un tableau contenant les n premiers termes de la suite u_n . La fonction doit marcher quel que soit l'entier n rentré.

Q-3-2 : Écrire le programme principal qui saisit l'entier n et affiche les n premiers termes de la suite u_n en utilisant la fonction définie au Q-3-1.

Exercice - 12 Chaînes de caractères

Q-1 : Sans utilisation de la bibliothèque `string.h`

Q-1-1 : Écrire une fonction qui prend en paramètre une chaîne et qui retourne le nombre d'occurrences de la lettre f dans la chaîne.

Q-1-2 : Écrire une fonction qui renvoie la concaténation de deux chaînes c'est à dire une chaîne constituée des deux chaînes mises bout à bout.

Q-2 : Utilisation éventuelle de la bibliothèque `string.h`

Q-2-1 : Écrire une fonction qui ouvre un fichier texte dont le nom est passé en paramètre et qui calcule la somme des longueurs des lignes.

Q-2-2 : Écrire une fonction qui prend en paramètre le nom d'un fichier texte et détermine le nombre de lignes du fichier qui commencent par le mot "programmons".

Exercice - 13 Tableaux à double entrée

Q-1 : Cas statique

Q-1-1 : Écrire une fonction de saisie d'un tableau à double entrée statique de `int`. On fera saisir le nombre de lignes et de colonnes au clavier.

Q-1-2 : Écrire une fonction qui prend en paramètre un tableau statique à double entrée de type `int`, son nombre de lignes et son nombre de colonnes, et compte le nombre de fois qu'apparaît un nombre x passé en paramètre.

Q-1-3 : Écrire une fonction qui prend un tableau à double entrée `tab` et qui renvoie un tableau dont les éléments correspondent à la parité des éléments de `tab`.

Q-1-4 : Écrire un programme principal qui saisit un tableau à double entrée, et compte le nombre d'éléments qui sont pairs dans ce tableau.

Q-2 : Cas dynamique

On considère un tableau à m lignes et n colonnes entrées dans un fichier. La première ligne du fichier contient les nombres m et n . Les lignes suivantes du fichier contiennent les coefficients du tableau. Les colonnes sont séparées par des espaces.

m	n				
$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	\dots	$a_{0,n-1}$	
$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	\dots	$a_{0,n-1}$	
\dots	\dots	\dots	\dots	\dots	
$a_{m-1,0}$	$a_{m-1,1}$	$a_{m-1,2}$	\dots	$a_{m-1,n-1}$	

Q-2-1 : Écrire une fonction d'allocation du tableau de dimension 2 de m lignes et n colonnes, les nombres m et n sont passés en paramètres.

Q-2-2 : Écrire une fonction de libération de mémoire pour un tableau de dimension 2.

Q-2-3 : Écrire une fonction qui réalise le chargement du fichier dans un tableau de tableaux (ou *matrice*) A .

Q-2-4 : Écrire une fonction qui calcule la somme des coefficients de chaque ligne de la matrice A et met tous les résultats dans un tableau de m nombres. La fonction retournera ce tableau.

Q-2-5 : Écrire le programme principal qui affiche la somme des coefficients de chaque ligne d'une matrice stockée dans un fichier, et libère la mémoire.