

TP N° 7 : Les fonctions

Objectifs

Résoudre quelques problèmes classiques en utilisant les fonctions avec et sans valeurs de retour.

Exercice 1

1. Ecrire une fonction **MIN** et une fonction **MAX** qui déterminent le minimum et le maximum de deux nombres réels. Ecrire ensuite un programme se servant des fonctions MIN et MAX pour déterminer le minimum et le maximum de quatre nombres réels entrés au clavier.
2. Ecrire la fonction **Impaire** qui permet de déterminer si une valeur est impaire.
3. La fonction **Absolue** qui retourne la valeur absolue d'un nombre entier.

Exercice 2

Ecrire une fonction **Puissance** qui calcule la valeur x^N pour une valeur réelle x (type float) et une valeur entière positive N (type int).

Exercice 3

Ecrire une fonction **Fact** qui reçoit la valeur N (type **int**) comme paramètre et qui fournit la factorielle de N comme résultat. Ecrire un petit programme qui teste la fonction Fact.

```
0! = 1
n! = n*(n-1)*(n-2)* ... * 1 (pour n>0)
```

Exercice 4

En utilisant les deux fonctions créées dans l'exercice 2 et 3, écrivez la fonction **func(x, n)** qui permet de retourner la valeur de l'expression suivante :

$$\sum_{i=0}^n \frac{x^i}{i!}$$

Ecrire le programme principal qui saisit les valeurs de x et de n pour calculer **func(x, n)** et affiche par la suite le résultat.

Exercice 5

1. Ecrire une fonction **Position** qui prend en paramètres un tableau d'entiers et un entier et retourne la position de cet entier dans le tableau. S'il n'existe pas, cette fonction doit retourner la valeur 0.
2. Ecrire une fonction **Supprimer** qui prend en paramètres un tableau et un indice k et supprime l'élément de position k du tableau.

Dans la fonction principale **main()** vous devez utiliser les deux fonctions Position et Supprimer pour supprimer un élément d'un tableau d'entiers autant de fois qu'il existe.

Exercice 6

On appelle permutation d'ordre n la suite des n entiers positifs 1,2,.....n dans un ordre quelconque. Par exemple, les suites suivantes sont respectivement des permutations d'ordre 5, 3 et 8.

- 1 2 3 4 5
- 3 1 2
- 4 8 2 3 6 1 5 7

Ecrire une fonction **VérifierPermutation** qui permet de vérifier si une suite est une permutation. (Utilisez un tableau d'entiers)

Exercice 7

1. Ecrire une fonction **LIRE_TAB** qui permet de remplir un tableau **TAB** de dimension maximale **NMAX** et de dimension effective **N**, donnés comme paramètres.
2. Ecrire une fonction **ECRIRE_TAB** qui permet d'afficher un tableau **TAB** de dimension **N**.
3. Ecrire une fonction **SOMME_TAB** qui retourne la somme des éléments du tableau **TAB**, de dimension **N**.
4. Ecrire un programme principale qui permet de tester ces différentes fonctions sur un tableau d'entiers **T** de taille maximale 100, et de taille effective **L** donné par l'utilisateur.

Exercice 8

Ecrire la fonction **strlen** qui calcule et retourne la longueur d'une chaîne de caractères entrée comme paramètre.

Exercice 9

Ecrire une fonction **Apparition** qui prend en paramètres une chaîne de caractères et une lettre et calcule le nombre d'apparition de la lettre dans cette chaîne.

Exercice 10

1. Ecrire une fonction **LONG_CH** qui retourne la longueur d'une chaîne de caractères CH donnée comme paramètre.
2. Ecrire une fonction **AJOUTE_CH** qui retourne une chaîne de caractères CH après l'ajout d'une chaîne CH1 à la fin d'une chaîne CH2 données comme paramètre.
3. Ecrire une fonction **PERMUTE_CH** qui permute deux caractères char * CH1 et char * CH2.
4. Ecrire une fonction **INVERSE_CH** qui permet d'inverser une chaîne de caractères CH donnée en paramètre en utilisant la fonction **PERMUTE_CH**.
5. Ecrire un programme principale qui permet de tester ces différentes fonctions sur deux chaînes de caractères données par l'utilisateur et de taille maximale 100.

Exercice 11

Ecrire une fonction **SuppOcc** qui a comme paramètre deux chaînes de caractères CH1 et CH2 et supprime la première occurrence de CH2 dans CH1.

Exemples: Alphonse phon ==> Alse
 totalement t ==> otatement
 abacab aa ==> abacab

Exercice 12

On désire modéliser des opérations dans un établissement bancaire à l'aide de tableaux à une dimension. On considère pour cela un tableau (tabCpt) d'entiers où les éléments représentent des numéros de compte, et un tableau (tabSoldes) de réels représentant les soldes des comptes. Ainsi, un élément d'indice i du tableau tabSoldes représente le solde du compte de la case i du tableau tabCpt. (On suppose que les tableaux ont une capacité maximale définie par une constante max donnée).

On demande d'écrire les fonctions suivantes :

- **SaisirComptes** qui permet de saisir à partir du clavier les deux tableaux en même temps (chaque itération permet de saisir un numéro de compte et un solde et de les stocker au même indice dans les deux tableaux).
 - **AffichComptes**, qui permet d'afficher à l'écran les deux tableaux en même temps (chaque itération permet d'afficher un numéro de compte et le solde correspondant).
 - **Rechercher** qui permet de rechercher dans le tableau tabCpt un compte de numéro num et retourne sa position dans le tableau s'il existe, -1 sinon.
-

-
- **Ajouter**, qui permet d'ajouter un nouveau compte. Cette opération consiste à ajouter le numéro num à la fin du tableau tabCpt et le solde s à la fin du tableau tabSoldes.
 - **Deposer** , qui permet d'ajouter un montant donné au solde d'un compte de numéro donné (n représente la taille des deux tableaux, num représente le numéro du compte sur lequel on désire effectuer le dépôt, montant représente le montant à déposer). Cette fonction procède de la manière suivante : elle effectue une recherche du numéro de compte dans le tableau tabCpt, s'il existe alors elle ajoute le montant au solde correspondant, s'il n'existe pas elle affiche un message d'erreur.
 - **Retirer**, qui permet de retirer un montant donné au solde d'un compte de numéro donné (n représente la taille des deux tableaux, num représente le numéro du compte dont on désire retirer de l'argent, montant représente le montant à retirer). Cette fonction procède de la manière suivante : elle effectue une recherche du numéro de compte dans le tableau tabCpt, s'il existe alors elle retire le montant du solde correspondant (si le solde le permet), s'il n'existe pas ou si le solde est insuffisant elle affiche un message d'erreur.
-