

Introduction à la programmation

TD 3 - Fonctions

Polytech Marseille - 1ère année

Filière Informatique
Alexandra Bac - Benoît Favre

Exercice 1 (PREMIERES FONCTIONS).

- (i) Ecrire deux fonctions `minf` et `maxf` prenant en argument deux `float` x, y et *renvoyant* respectivement leur minimum et leur maximum
- (ii) Ecrire deux fonctions `print_min` et `print_max` prenant en argument deux `float` x, y et *affichant* respectivement leur minimum et leur maximum
- (iii) Enfin en faisant appel à certaines des fonctions précédentes, écrire deux fonctions `minf_ter` et `maxf_ter` prenant en argument trois `float` x, y, z et *renvoyant* respectivement leur minimum et leur maximum
- (iv) Testez ces fonctions depuis le `main`.

Exercice 2 (PUISSANCE D'UN NOMBRE - version itérative élémentaire). Ecrivez une fonction calculant de manière itérative x^n où x est un nombre réel et n un entier positif.

Exercice 3 (TABLEAUX). Ecrivez les fonctions suivantes :

- (i) une fonction affichant le contenu du tableau T (les valeurs des cases seront séparées par des espaces)
- (ii) une fonction inversant ses valeurs (la première case passe dans la dernière etc. ...)

Exercice 4 (ECRITURE BINAIRE D'UN NOMBRE). Ecrivez une fonction (itérative) qui calcule les chiffres de l'écriture en base 2 d'un entier N et les range dans un tableau de caractères passé en paramètre.

Exercice 5 (TERMES D'UNE SUITE). Ecrivez des fonctions calculant (de manière itérative) les n premiers termes de la suite :

$$u_{n+2} = u_n + u_{n+1} \text{ avec } u_0 = u_1 = 1 \text{ (suite de Fibonacci)}$$

- (i) Dans un premier temps, votre fonction affichera ces termes.
- (ii) Dans un deuxième temps, stocker ces n premiers termes dans un tableau T qui sera passé en argument à la fonction.

Exercice 6 (LES INEVITABLES C_n^p). On rappelle la formule :

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

Ecrivez une fonction qui garnit un tableau passé en argument avec les valeurs des n premières lignes du «triangle du Pascal» (voir ci-dessous), n étant également fourni en argument.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

Exercice 7 (RECURSIVITE). Ecrire des fonctions récursives (obligatoirement) :

```
void compteur_inverse (int n)
void compteur (int n)
```

affichant les entiers de n à 1 pour la première, et les entiers de 1 à n pour la seconde.

Exercice 8 (RECURSIVITE). Écrire une fonction récursive `draw_row` qui affiche n fois le caractère `c` sur une ligne. Écrire une fonction récursive `draw_diamond` qui utilise `draw_row` pour afficher un losange de hauteur et largeur m comme dans l'exemple ci dessous.

```
  *
 * * *
* * * * *
 * * *
  *
```

Exercice 9 (RECURSIVITE). Reprendre les exercices 2, 4, 5 et 6 en écrivant des fonctions **récursives** pour résoudre ces problèmes.

Exercice 10 (PRODUIT DE MATRICES.). Ecrivez une fonction qui calcule le produit C d'une matrice A ayant NLa lignes et Nca colonnes par une matrice B ayant Nlb lignes et Ncb colonnes. Ces trois matrices seront représentées par des tableaux statiques bidimensionnels de `double`.

On rappelle que le produit de $A_{nla,nca} \times B_{nlb,ncb}$ est une matrice $C_{nla,ncb}$ définie par :

$$C_{i,j} = \sum_{k=0}^{nca-1} A_{i,k} \times B_{k,j}$$

- (i) Quelles conditions faut-il vérifier sur les tailles de A et B ? (vérifier cette condition par un `assert` en début de fonction). Dans ce cas, quelle est la taille de C ?
 - (ii) Ecrire la fonction produit matriciel et la tester.
-

Bonus

Exercice 11 (PUISSANCE D'UNE MATRICE.). Ecrivez, en utilisant les exercices précédents, une fonction calculant X^n , où X est une matrice carrée et n un entier positif ou nul.