

Programmation
Licence 1 UPEC 2022/2023
Travaux Machine 7 : Fonctions et tableaux

Ce TM porte sur la conception et la programmation des fonctions. Chaque point vous demande de programmer une fonction qui a un certain nombre de paramètres, chacun avec son type, et qui calcule quelque chose. Vous devez identifier, pour chaque point :

1. Le nombre de paramètres de la fonction.
2. Le type de chaque paramètre.
3. Dans le cas où la fonction demandée doit renvoyer une valeur, le type de cette valeur.

Vous devez ensuite créer un programme qui contient le code de la fonction demandée et le "programme principal" `main` où vous devez appeler votre fonction. Dans chaque cas, l'appel doit être précédé de la déclaration et l'initialisation de variables qui seront passées en paramètre à votre fonction, puis de l'affichage du résultat calculé par votre fonction. Cela représente le "test" de votre fonction. Il est conseillé de faire **au moins 5 tests par point**, avec différentes valeurs pour chacun des paramètres lors de chaque test.

La réponse à chaque point n'est pas complète tant que vous n'avez pas créé le programme principal `main` qui vous permet de tester votre fonction !

Exercices

Question 1 : Fonctions simples

1. Écrire une fonction `somme` qui prend en paramètre un tableau `t` d'entiers et qui renvoie la somme de toutes les valeurs qui se trouvent dans le tableau.
2. Écrire une fonction `appartient` qui prend en paramètre un tableau `t` d'entiers et un entier `n` et qui renvoie `true` si `n` est dans une des cases de `t`, et `false` sinon.
3. Écrire une fonction `paire` qui prend en paramètre un tableau `t` d'entiers et qui renvoie `true` si `t` contient deux cases de suite qui ont une même valeur, et `false` sinon.
4. Écrire une fonction `brelan` qui prend en paramètre un tableau `t` d'entiers et qui renvoie `true` si `t` contient *trois* cases de suite qui ont une même valeur, et `false` sinon.

Question 2 : Operations

1. Écrire une fonction `sommeMultiple` qui prend en paramètre un tableau `t` d'entiers et un entier `n` et qui renvoie la somme de toutes les valeurs qui se trouvent dans une case dont l'indice est un multiple de `n`.
2. Écrire une fonction `tabMultiple` qui prend en paramètre un tableau `t` d'entiers et un entier `n` et qui renvoie un nouveau tableau qui contient seulement les valeurs qui se trouvent dans une case de `t` dont l'indice est un multiple de `n`.
3. Écrire une fonction `tabHasard` qui prend en paramètre deux entiers `n` et `h` et qui renvoie un tableau de taille `n` contenant `n` nombres différents choisis au hasard entre 0 et `h`.
4. Écrire une fonction `sommeHasard` qui prend en paramètre un tableau `t` d'entiers et un entier `n` et qui renvoie la somme de `n` entiers choisis au hasard dans `t`.

Question 3 : Modification d'un tableau

1. Écrire une fonction `void augmente2(int[] tab)` qui prend en entrée un tableau d'entiers, et qui augmente de 2 la valeur de chaque case du tableau.
2. Écrire une fonction `void augmente(int[] tab, int n)` qui prend en entrée un tableau d'entiers et un entier n , et qui augmente de n la valeur de chaque case du tableau.
3. Écrire une fonction `void echange(int[] tab, int i, int j)` qui prend en entrée un tableau d'entiers et deux entiers i, j , et qui échange de place les valeurs des cases i et j . Votre procédure ne doit pas vérifier si les entiers correspondent réellement à des cases du tableau.
4. Écrire une fonction `void rotation(int[] tab)` qui prend en entrée un tableau d'entiers et qui déplace chaque valeur d'une case à la suivante. La valeur de la dernière case sera mise dans la case 0.
5. Écrire une fonction `void remplissage(int[] source1, int[] source2, int[] target)` qui prend en entrée trois tableaux d'entiers et qui commence à mettre les valeurs de `source1` dans `target` tant qu'il y a de la place. Ensuite elle met les valeurs de `source2` dans `target` tant qu'il y a de la place. S'il reste encore de la place dans `target`, on remplit les cases qui restent avec des 0.

Question 4 : Modification d'un tableau à deux dimensions** Écrire une procédure `transformeC(int[][] tab)` qui prend en entrée un tableau d'entiers à deux dimensions, et qui le transforme en un tableau carré rempli partout par des 0. Il n'y a aucun choix à faire : toutes les informations sont dans l'énoncé !

Modèle d'exécution

Question 1 On considère les deux programmes suivants, composés d'un `main` et d'une fonction. Pour chacun, décrire les affichages et le modèle d'exécution.

— Premier programme :

```
void main() {
    int a = 5;
    int b = 9;
    afficheSomme(a,b);
}

void afficheSomme(int x, int y) {
    int s = x+y;
    println("La somme est " + s + ".");
}
```

— Deuxième programme :

```
void main() {
    int[] tab = {4,5,6,7};
    cinq(tab);
}

void cinq(int[] tab) {
    for (int i=0; i<tab.length; i++) {
        tab[i]=5;
    }
}
```