

Développement orienté objet - TypeScript

Tableaux

Exercice 1

Ecrire une fonction qui prend un entier **n** en paramètre. La fonction demande **n** nombres entiers à un utilisateur. Lorsque l'utilisateur a fini, la fonction affiche les nombres pairs.

Exercice 2

1. Ecrire une fonction **ajouterUn** qui prend en paramètre un tableau de nombres et qui ajoute un à tous les éléments du tableau. La fonction doit modifier le tableau passé en paramètre et ne pas créer une copie. Par exemple, si on fait l'appel :

```
let x = [1, 2, 3];
ajouterUn(x);

console.log("x =", x);
```

```
x = [ 2, 3, 4 ]
```

2. Ecrire une fonction **plusUn** qui prend en paramètre une liste d'entiers et qui retourne une copie de la liste où on a ajouté à tous les éléments. La liste passée en paramètre ne doit pas être modifiée. Par exemple, si on l'appelle :

```
let x = [1, 2, 3];
let y = plusUn(x);

console.log("x =", x);
console.log("y =", y);
```

```
x = [ 1, 2, 3 ]
y = [ 2, 3, 4 ]
```

Exercice 3

Écrire une fonction qui effectue une rotation d'un tableau passé en paramètre : chaque élément est décalé d'une case vers la droite et le dernier élément de la table occupe la première place. Il faut que la fonction

modifie le tableau passé en paramètre et ne crée pas une copie du tableau. Écrire un appel de cette fonction.

Par exemple, si avant la rotation, on a la liste :

indice	0	1	2	3	4
valeur	23	28	6	49	50

après l'appel, on aura :

indice	0	1	2	3	4
valeur	50	23	28	6	49

Exercice 4

Ecrire une fonction qui prend un tableau en paramètre et qui inverse l'ordre des éléments. Il faut que la fonction modifie le tableau passé en paramètre et ne crée pas une copie du tableau. Ecrire un appel à la fonction.

Exercice 5

1. Écrire un programme qui affiche le nombre de tirages de la somme de de deux dés à 6 faces sur 10000 tirages. Le résultat devrait ressembler à :

```
somme = 2 : 290 tirages
somme = 3 : 593 tirages
somme = 4 : 827 tirages
somme = 5 : 1137 tirages
somme = 6 : 1411 tirages
somme = 7 : 1648 tirages
somme = 8 : 1343 tirages
somme = 9 : 1115 tirages
somme = 10 : 794 tirages
somme = 11 : 579 tirages
somme = 12 : 263 tirages
```

Consigne : vous n'avez le droit de déclarer qu'une seule variable pour stocker les nombres de tirages.

Indication : Vous pouvez vous utiliser du code ci-dessous pour générer un entier aléatoire entre deux valeurs **lb** et **ub**.

```
function randInt(lb: number, ub: number): number {
    return Math.floor(Math.random() * (ub - lb + 1)) + lb;
}
```

2. Modifier le programme précédent pour afficher l'histogramme graphiquement avec des étoiles. On utilisera l'échelle "une étoile = 100 tirages". Le résultat devrait ressembler à :

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