

TEST PROG1 – G1

Decembre 2021 – Semaine 4

Exo 1 :

1. Expliquez ce que va produire le code suivant **1pt** :

```
for(let i = 1; i<=3; i++)  
    console.log(i);  
  
console.log(i)
```

2. Même question si on remplace *let* par *var*? **1pt**
3. Que fait la fonction *slice* dans un tableau ? Et la fonction *filter* ? **2pt**

Exo 2 :

On souhaite créer un outil statistique pour calculer des indicateurs basiques tels que la variance et l'écart-type. Pour cela créez les fonctions suivantes :

- La fonction **sum** qui, étant donné un tableau de nombres, retourne la somme des éléments de cette liste, ou 0 si la liste est vide **(2pts)**;
- la fonction **mean** qui, étant donné un tableau non vide de nombres, retourne la moyenne des éléments de cette liste **(2pts)**;
- la fonction **square** qui, étant donné un tableau de nombres L, retourne la liste des carrés des éléments de L **(2pts)**;
- la fonction **variance** qui étant donné un tableau non vide de nombres, renvoie la variance (la différence entre la moyenne des carrés des éléments de la liste, et le carré de la moyenne des éléments de la liste)

(2pts);

- la fonction **standardDeviation** qui étant donné une liste non vide de nombres, renvoie l'écart-type de la liste (racine carrée de la variance) **1pt**.

Exo 3 : 3pts

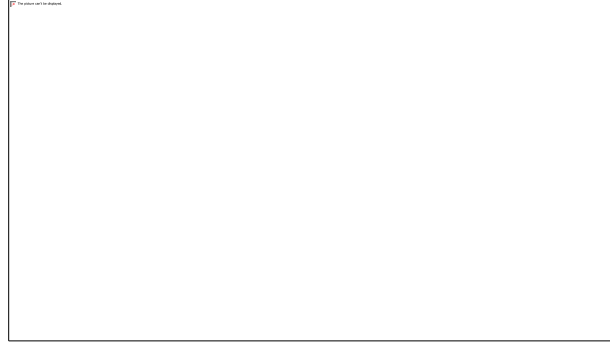
Créez une fonction qui calcule le PGCD de deux nombres. Une des formules possibles pour calculer le PGCD est la suivante :

- $\text{pgcd}(a, b) = \text{pgcd}(a - b, a)$, si $a > b$
- $\text{pgcd}(a, b) = \text{pgcd}(a, b - a)$, si $b > a$
- $\text{pgcd}(a, b) = a$, si $a = b$

Exo 4 : (4 pts)

Pour convertir une feuille Excel sur les étudiants en base de données, il faut qu'il n'y ait aucun doublons sur les numéros qui les identifie. [1, 2, 3, 4, 4] par exemple contient un doublon, mais [5,6,3] non.

Créez une fonction **hasDuplicate** qui verifie si un tableau contient des doublons. Elle retournera *true* si elle contient des doublons, et *false* sinon.



TEST PROG1 – G2

Decembre 2021 – Semaine 4

Exo 1 :

1. Décrire ce que va produire le code suivant

```
console.log(value);  
  
var value = "some value";
```

2. Même question si l'on changeait *var* par *let* ?

3. Que fait la fonction *filter* dans un tableau ? Et la fonction *indexOf* ?

Exo 2 :

On souhaite créer un outil littéraire pour déterminer les caractéristiques d'un mot. Pour cela créez les fonctions suivantes :

- **letterCount** : qui soit un string et un caractère, retourne le nombre d'apparition du caractère dans ce string;
- **vowelCount** : qui soit un string, va compter le nombre de voyelles que l'on limite à : **a, o, i, e, u, y, w** pour cet exercice
- **consonantCount**: qui soit un string, va compter le nombre de consonnes en supposant que le string ne contient que des voyelles et des consonnes
- **isIncluded** : qui soient deux string passées vérifie si le premier est inclus dans le deuxième. Ex : **isIncluded**(«sac», « sachant ») retourne *true*;
- **removeAccents** : qui soit un string, va remplacer [é,è,ê] par e, à par a et retourne le nouveau string produit. Attention, les strings en Javascript sont immuables : Pour `let s = "hello"`, on ne peut pas faire `s[0] = "c"`.

Exo 3:

Selon des biologistes, chaque être vivant est organisé selon des principes mathématiques bien précis : la suite de Fibonacci. C'est une suite mathématique où, à l'exception des deux premiers, chaque terme est égal à la somme des deux termes qui le précèdent immédiatement. Tout nombre (à partir du troisième) est donc égal à la somme des deux précédents. Formellement, on a :

- $\text{Fibonacci}(0) = 0$ et $\text{Fibonacci}(1) = \text{Fibonacci}(2) = 1$;
- $\text{Fibonacci}(n+2) = \text{Fibonacci}(n+1) + \text{Fibonacci}(n)$, avec $n > 0$

Créez une fonction récursive **fibonacci** qui permet de calculer $\text{Fibonacci}(x)$.

Exo 4 :

Avant de publier les résultats finaux de la L1, on vérifie si les étudiants sont bel et bien classés dans l'ordre décroissant selon leur moyennes finales.

Pour cela, écrivez une fonction **isDecreasing** qui vérifie si un tableau de nombre est trié dans l'ordre décroissant. Elle retourne *true* si le tableau est trié dans l'ordre décroissant, et *false* sinon.