**Test Java**

**La méthode sumRange devrait retourner la somme des entiers compris entre 10 et 100 inclusifs contenus dans le tableau passé en paramètre.**

Corrigez la méthode sunRange

**Note :** le paramètre ints n’est jamais null

public class Solution {

static int sunRange(int[] ints) {

int sum = 0;

for (int i = 1; i < ints.length; i++) {

int n = ints[i];

if (n>=10 || n<=100) {

sum +=n;

}

}

return sum;

}

}

**Code de test :**

int[] ints = {1, 20, 3, 10, -2, 100};

int result = Solution.sunRange(ints);

System.out.println("result : "+result);

System.out.println("expected result : 130");

**Transformez la méthode Counter.increment() pour qu’elle supporte l’accès concurrentiel de plusieurs threads**

public class Counter {

private static int count = 0;

/\*

\* Increments count in a thread-safe manner

\*/

public static int increment() {

count +=1;

return count;

}

}

**Code de test**

Counter.increment()

**En base 2 (binaire), combien font 0001 & 0001 ?**

**Le Garbage collector garanti qu’il y a suffisamment de mémoire pour exécuter un programme java ?**

**Implémentez la méthode Algorithm.findLargest(int[] numbers) afin qu’elle retourne le plus grand nombre dans numbers.**

Note :le tableau contient toujours au moins un nombre

public class Algorithm {

public static int findLargest(int[] numbers) {

//Your code

return 0;

}

}

**Code de test**

int[] numbers = {1,-28,88,300,600};

int result = Algorithm.findLargest(numbers);

System.out.println(result);

**Parmi ces deux procédures de développement, laquelle préconisez-vous ?**

**Procédure #1**

1. Ecrire des tests pour la nouvelle fonctionnalité “F”
2. Tester que “F” ne fonctionne pas
3. Implémenter “F”
4. Tester que “F” fonctionne correctement

**Procédure #2**

1. Implémenter la nouvelle fonctionnalité “F”
2. Ecrire des testes pour “F”
3. Tester que “F” fonctionne correctement

**Que se passe-t-il lors d’un merge “fast-forward” ?**

**Quelle commande permet d’effacer la branche distante nommée “branfix” ?**

**Un dépôt distant à pour URL https//github.com/exp/hello.git**

Ecrivez la commande utilisée pour créer une copie locale de ce dépôt dans le répertoire courant

**La commande permet de récupérer les nouvelles données d’un dépôt distant sans pour autant combiner ces données avec les fichiers de votre espace de travail ?**

**Quelle fonctionnalité de Github met à disposition une interface web conviviale pour notifier les membres d’une équipe projet de la fin d’un développement et permet ensuite que les changements proposés soient revus puis intégrés dans la branche master ?**

**Quel est l’usage typique de la commande git bisect ?**

**Quelle commande utilisant l’option —mount permet de monter un volume en lecture seule (dans docker) ?**

**Pour investiguer un problème, vous avez besoin de lister les ports publiés par un container.**

Quelle commande pourriez-vous utiliser dans cette situation ?

**Pour investiguer un problème, vous avez besoin de lister les ports publiés par un container.**

Quelle commande pourriez-vous utiliser dans cette situation ?

**Y-a-t-il des différences entre les instructions CMD et ENRYPOINT d’un dockerfile ?**

**Quelle commande permet de créer un container avec pour un serveur DNS personnalisé 8.8.8.8 ?**

**Quel est le rôle de l’instruction EXPOSE dans un Dockerfile ?**

**Scrum a-t-il un rôle appelé “Project Manager”**

Ecrivez une expression en SpEL (Spring Expression Language) en remplacement de XXX dans le code ci-dessous.

L’objectif est d’injecter la valeur de la propriété système user.dir en tant que valeur la propriété defaultDir

<bean id="csClient" class="org.example.CsClient">

<property name="defaultDir" value="XXX"/>

</bean>

**Quelles différences entre @RequestParam et @PathVariable ?**

**Observez les deux façons d’injecter un service dans un bean:**

@Component

public Class MyService {

private AnotherService anotherService;

@Autowired

public MyService(AnotherService anotherService) {

this.anotherService = anotherService;

}

}

@Component

public Class MyService {

@Autowired

private AnotherService anotherService;

public MyService(AnotherService anotherService) {

this.anotherService = anotherService;

}

}

Le résultat est le même pour les deux codes Mais y en a-t-il un qui est préférable à l’autre ?

**Quelle annotation peut-on utiliser pour injecter un bean Spring ?**