

 **achref** (sallemi.achref@gmail.com)

Campagne : JavaScript, HTML, CSS, Angular 2+, Java, Spring Fram... - Expert

Domaine(s) : Angular 2+, CSS, HTML, Java, JavaScript, Spring Framework

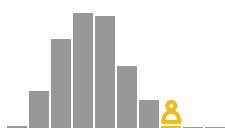
Langage : Français

Date : 29/09/2022

MEILLEUR QUE

>99%

des professionnels



RANG

1 / 1



DURÉE

0h51 / 1h47



SCORE

2 010 / 2 660 (76%)

Angular 2+ 300 / 300pts (100%)

MEILLEUR QUE >99% des professionnels

Connaissance du langage



300 / 300pts

CSS 200 / 200pts (100%)

MEILLEUR QUE >99% des professionnels

CSS



200 / 200pts

HTML 140 / 160pts (88%)

MEILLEUR QUE 61% des professionnels

HTML



140 / 160pts

Java 1 000 / 1 450pts (69%)

MEILLEUR QUE **>99%** des professionnels

Fiabilité



181 / 206pts

Résolution de problèmes



819 / 1 244pts

JavaScript 350 / 530pts (66%)

MEILLEUR QUE **88%** des professionnels

Connaissance du langage



219 / 339pts

Fiabilité



19 / 19pts

Modélisation



0 / 60pts

Résolution de problèmes



112 / 112pts

Spring Framework

20 / 20pts (100%)

MEILLEUR QUE **>99%** des professionnels

Spring Boot



20 / 20pts

Question 1: Utilisation des pipes (percent, currency, date)



Angular 2+



09:59 / 20:00



8x (28 sec)



300 / 300 pts



Question

On vous demande de terminer le composant `TransactionDetailsComponent`. Vous devez uniquement modifier la partie template du composant.

L'objectif de ce composant est d'afficher la date, le montant, la devise et les frais associés à une transaction, chacun dans un format spécifique.

Vous devez afficher 3 divs :

Le div "Fee" (frais)

L'id de ce div doit être `fee`. Il affiche les frais `fee` associés à la transaction sous forme de pourcentage : S'il y a moins de 2 chiffres pour la partie entière de la valeur en pourcentage, vous devez combler avec des zéros à gauche. S'il y a moins de 2 chiffres dans la partie décimale, vous devez combler avec zéros à droite. S'il y a plus de 3 chiffres dans la partie décimale, vous devez arrondir à 3 chiffres. On utilise un point `.` en tant que séparateur décimal.

Par exemple `0.031234` s'affiche `03.123%`.

Le div "Amount" (montant)

L'id de ce div doit être `amount`. Il affiche les frais `amount` et la devise `currency` de la transaction. Le symbole de la devise associée au code `currency` (par exemple `€` pour `EUR`) est affiché avant les frais. Les frais sont formatés comme suit :

S'il y a moins de 9 chiffres pour la partie entière, vous devez combler avec des zéros à gauche. Le séparateur des milliers doit être une virgule `,`. S'il y a moins de 2 chiffres dans la partie décimale, vous devez combler avec zéros à droite. S'il y a plus de 2 chiffres dans la partie décimale, vous devez arrondir à 2 chiffres.

Par exemple `currency=EUR`, `amount=312.562` s'affiche `€000,000,312.56`.

Le div "Time" (date)

L'id de ce div doit être `time`. La date et l'heure de la transaction doivent être affichés dans ce format inhabituel : 'ww: yyyy MMMMM dd hh-mm-ss'

Notes : Un bloc de "Preview" est disponible pour vous permettre de déboguer votre code. Il affiche votre composant ci-dessous. Vous pouvez ouvrir ce bloc et le modifier à votre guise. Le bloc de "Preview" n'est pas pris en compte dans le calcul du score de votre code.



Réponse

```
1 // Angular 8.x code
2 import { Component, Input, NgModule } from '@angular/core';
3 import { CommonModule } from '@angular/common';
4
5 @Component({
6   selector: 'transaction-component',
7   template: `
8     <!-- Your code goes here -->
9     <div id="fee">{{ fee | percent:'2.2-3' }}</div>
10    <div id="amount">{{ amount | currency:currency: 'symbol':'9.2-2'}}</div>
11    <div id="time">{{ timeOfTransaction | date:'ww: yyyy MMMMM dd hh-mm-ss' }}</div>
12  `
13 })
14 export class TransactionDetailsComponent {
15
16   @Input()
17   public currency: string;
18
19   @Input()
20   public timeOfTransaction: Date;
21
22   @Input()
23   public amount: number;
24
25   @Input()
26   public fee: number;
27
28 }
29
30 // #region Preview
31 @Component({
32   template: `<transaction-component [fee]=0.02 [amount]=123.45 [currency]='EUR' "
33   [timeOfTransaction]='getPresetDate()'></transaction-component>`
34 })
35 export class PreviewComponent {
36   public getPresetDate(){
37     return new Date(1997,6,1,12,32);
38   }
39 }
40 // #endregion Preview
41
42 // #region Module declaration - Do not Change
43 @NgModule({
44   imports: [CommonModule],
45   declarations: [PreviewComponent, TransactionDetailsComponent],
46   entryComponents: [PreviewComponent]
47 })
48 export class PreviewModule { }
49 // #endregion Module declaration
```

Résultat

-  Compile
Connaissance du langage +0pts
-  Affichage pourcentage
Connaissance du langage +60pts
-  Affichage de date
Connaissance du langage +60pts
-  Affichage EUR
Connaissance du langage +60pts
-  Affichage JPY
Connaissance du langage +60pts
-  3 décimales pour le pourcentage
Connaissance du langage +60pts

Question 2: [CSS] Syntaxe



CSS



00:15 / 00:45



20 / 20 pts



Question

Quelle syntaxe CSS est correcte ?

(Une seule réponse attendue)



Réponse



`{body; color:black}`



`body:color=black;`



`{body:color=black;}`



`body {color:black;}`



`body {color=black;}`



Résultat



Réponse correcte

CSS +20pts

Question 3: [CSS] z-index



CSS



00:16 / 00:40



40 / 40 pts



Question

En CSS, la propriété `z-index` permet de :



Réponse

- ☐ Spécifier la position d'un élément dans une liste
- ☐ Affiner la position d'un élément dans la page
- ☒ Spécifier la position d'empilement d'un bloc par rapport à un autre
- ☐ Spécifier l'ordre d'accès des champs d'un formulaire (touche tabulation)



Résultat



Réponse correcte

CSS +40pts

Question 4: [CSS] Padding



CSS



00:17 / 00:30



40 / 40 pts



Question

En CSS, padding est utilisé pour :



Réponse

- ☒ les marges interieures
 - ☐ les marges extérieures
 - ☐ l'alignement intérieur
 - ☐ l'alignement extérieur
-



Résultat



Réponse correcte

CSS +40pts

Question 5: [CSS] Couleurs



CSS



00:22 / 00:25



40 / 40 pts



Question

En CSS, dans quel format est spécifiée la couleur #A1B2C3 ?



Réponse

- ☐ Cyan Magenta Jaune Noir hexadécimal
 - ☐ Rouge Vert Bleu décimal
 - ☒ Rouge Vert Bleu hexadécimal
 - ☐ Cyan Magenta Jaune Noir octal
-



Résultat



Réponse correcte

CSS +40pts

Question 6: [CSS] Sélecteurs



CSS



00:28 / 00:45



60 / 60 pts

? Question

À partir du code HTML suivant :

```
<div id="container">
  <div>
    I am a sub container
  </div>
</div>
```

quel selecteur(s) CSS peut-on utiliser pour sélectionner l'élément dont l'id est égal à "container" ?

Cochez toutes les solutions possibles.

📝 Réponse

- ☐ .container
- ☐ container
- ☒ #container
- ☐ @container
- ☒ [id = "container"]

> Résultat

✅ Réponse correcte
CSS +60pts

Question 7: [HTML] Commentaires



HTML



00:10 / 00:30



0 / 20 pts



Question

Parmi les options suivantes, laquelle est un commentaire HTML valide ?



Réponse



`<-- Comment -->`



`<!-- Comment --!>`



`<!-- Comment -->`



`/* Comment */`



Résultat



Réponse incorrecte

HTML +20pts

Question 8: [HTML] Disabled



HTML



00:29 / 00:30



20 / 20 pts



Question

En HTML, que faut-il ajouter à un champ de formulaire pour le désactiver ?



Réponse

- ☐ Une classe disabled
 - ☒ Un attribut disabled
 - ☐ Une classe read-only
 - ☐ Un attribut read-only
 - ☐ Aucune des propositions
-



Résultat



Réponse correcte

HTML +20pts

Question 9: [HTML] Liste déroulante



HTML



00:12 / 00:30



40 / 40 pts



Question

En HTML, quel élément permet de créer une liste déroulante ?



Réponse



<select>



<option>



<input type="dropdown">



<dropdown>



Résultat



Réponse correcte

HTML +40pts

Question 10: [HTML] Balises



HTML



00:21 / 00:30



40 / 40 pts



Question

En HTML, laquelle des réponses suivantes correspond à une hiérarchie correcte d'éléments ?



Réponse



`<section> <div> <p> `



`<div> <section> <p>`



`<div> <p> <section> `



Résultat



Réponse correcte

HTML +40pts

Question 11: [HTML] Liens



HTML



00:36 / 00:40



40 / 40 pts



Question

En HTML, que fait le lien suivant ?

```
<a href="fleurs.html#coquelicot">un lien...</a>
```



Réponse

- ☒ Il ouvre la page fleurs.html et nous amène à l'ancre "coquelicot".
- ☐ Il ouvre la page coquelicot.html et nous amène à l'ancre "fleurs".
- ☐ Il ouvre la page fleurs.html et nous amène à l'ancre "#coquelicot".
- ☐ Ce lien n'est pas valide.



Résultat



Réponse correcte

HTML +40pts

Question 12: Du désordre le plus grand gagne



Java



01:36 / 05:00



100 / 100 pts

? Question

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.

✎ Réponse

```
1 import java.util.Arrays;
2 class Algorithm {
3
4     /** @return the largest number of the given array */
5     static int findLargest(int[] numbers) {
6         Arrays.sort(numbers);
7         return numbers[numbers.length-1];
8     }
9
10 }
```

> Résultat



Fonctionne dans des cas simples

Résolution de problèmes +32pts



Fonctionne quand le tableau contient seulement Integer.MIN_VALUE

Fiabilité +58pts



Fonctionne quand le plus grand élément est à la position 0

Fiabilité +5pts



Fonctionne quand le plus grand élément est à la fin du tableau

Fiabilité +5pts

Question 13: [JS] Comparaison avec ===



JavaScript



00:13 / 00:30



20 / 20 pts



Question

En JavaScript, quel est le résultat de l'expression suivante :

```
"2" === 2;
```



Réponse



true



false



null



undefined



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 14: Méthodes de chaînes de caractères



JavaScript



00:36 / 00:40



0 / 40 pts



Question

Laquelle des options suivantes n'est pas une méthode de `String` valide ?



Réponse

- ☒ indexOf
- ☐ includes
- ☐ slice
- ☐ splice
- ☐ charAt



Résultat



Réponse incorrecte
Connaissance du langage ~~+40pts~~

Question 15: [JS] Conversion de type #1



JavaScript



00:09 / 00:20



40 / 40 pts



Question

En JavaScript, quel est le résultat de l'expression suivante :

```
true == 1;
```



Réponse



true



false



null



undefined



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 16: [JS] Événements



JavaScript



00:30 / 00:30



40 / 40 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.



Question

En JavaScript, quel est l'évènement qui est déclenché lorsqu'un élément `<button>` ou `<textarea>` perd le focus ?



Réponse



onblur



onclick



ondblclick



onfocus



onfocuslost



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 17: [JS] Etapes de boucle



JavaScript



00:12 / 00:35



40 / 40 pts



Question

En JavaScript, quel mot clé permet de passer directement au tour suivant d'une boucle ?



Réponse

continue



Résultat



Réponse correcte

Connaissance du langage +40pts



Réponse(s) correcte(s)

- continue;
- continue

Question 18: Portée de bloc



JavaScript



00:27 / 00:30



0 / 20 pts



Question

Quels mots-clés fournissent une portée de bloc (block scope) ?

Sélectionnez toutes les bonnes réponses.



Réponse



const



let



var



Résultat



Réponse incorrecte
Connaissance du langage +20pts

Question 19: [JS] Comparaison de strings



JavaScript



00:30 / 00:30



60 / 60 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.



Question

En JavaScript, qu'affiche ce code ?

```
var x = "2", y = "10";  
alert(x < y);
```



Réponse

- ☐ true
- ☒ false
- ☐ "2<10"
- ☐ null
- ☐ Une erreur JavaScript



Résultat



Réponse correcte

Connaissance du langage +60pts

Question 20: [JS] Paramètres



JavaScript



00:38 / 00:40



0 / 60 pts



Question

En JavaScript, quel est le mot clé utilisé pour accéder au tableau des arguments d'une fonction ?



Réponse

var



Résultat



Réponse incorrecte
Connaissance du langage ~~+60pts~~



Réponse(s) correcte(s)

- arguments

Question 21: [JS] Prototypes



JavaScript



01:02 / 01:30



0 / 60 pts



Question

Soit un objet `obj` défini par le code suivant :

```
MyClass = function() {  
  this.a = 'hello';  
  this.b = 'world';  
};  
  
MyClass.prototype.a = 'john';  
MyClass.prototype.c = 12;  
  
obj = new MyClass();
```

Parmi les propositions suivantes, cochez celles qui sont vraies.

Plusieurs réponses attendues.



Réponse

- ☒ `obj.a` s'évalue à `'hello'`
- ☐ `obj.a` s'évalue à `'john'`
- ☐ `obj.b` s'évalue à `undefined`
- ☒ `obj.b` s'évalue à `'world'`
- ☒ `obj.c` s'évalue à `undefined`
- ☐ `obj.c` s'évalue à `12`
- ☒ `obj.d` s'évalue à `null`

> Résultat

✖ Réponse incorrecte
Modélisation ~~+60pts~~

Question 22: [JS] Propriétés d'Objets

JavaScript 02:44 / 03:00 1x (14 sec) 150 / 150 pts

? Question

Implémentez la fonction `helloProperties(obj)`.

Cette fonction prend en paramètre un objet `obj` comme celui-ci:

```
{  
  john: 12,  
  brian: true,  
  doe: 'hi',  
  fred: false  
}
```

Et renvoie une array contenant ses noms de propriétés, préfixés par "Hello-", comme ceci:

```
['Hello-john', 'Hello-brian', 'Hello-doe', 'Hello-fred'];
```

`obj` est toujours défini et est toujours un objet JavaScript



Réponse

```
1 // JavaScript code below
2 // Use printErr(...) to debug your solution.
3
4 function helloProperties(obj) {
5   var a = [];
6   // Your code goes here
7   Object.keys(obj).forEach(value => {
8     a.push('Hello-' + value);
9   });
10  return a;
11 }
```



Résultat



La solution fonctionne pour un cas standard
Résolution de problèmes +112pts



La solution fonctionne pour un objet vide
Fiabilité +19pts



La solution utilise Object.keys
Connaissance du langage +19pts

Question 23: [Boot] @SpringBootTest



Spring Framework



00:08 / 00:45



20 / 20 pts



Question

Quel est l'intérêt de l'ajout de l'annotation `@SpringBootTest` à une classe de test ?



Réponse

- ☐ Cela est nécessaire pour autoriser les tests unitaires sur des services simulés
- ☒ **Cela met en place un contexte d'application complet pour les tests d'intégration**
- ☐ Cela permet le démarrage de l'application si un cas de test échoue
- ☐ Cela démarre un Spring Boot REPL pour l'examen manuel de l'application



Résultat



Réponse correcte
Spring Boot +20pts

Question 24: Approximation de π



Java



10:28 / 12:00



10x (2 min)



300 / 300 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

? Question

Dans cet exercice nous allons calculer une estimation du nombre π (Pi).

La technique est la suivante :

On prend un point P au hasard de coordonnées (x, y) tel que $0 \leq x \leq 1$ et $0 \leq y \leq 1$. Si $x^2 + y^2 \leq 1$, alors le point est à l'intérieur du quart de disque de rayon 1, sinon le point est à l'extérieur.

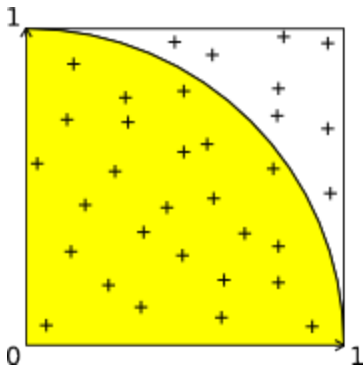


Fig 1. Exemple avec 33 points aléatoires.

On sait que la probabilité que le point se situe à l'intérieur du quart de disque est égale à $\pi/4$.

Écrivez la méthode `double approx(double[][] pts)` qui va utiliser les points `pts` (tirés au hasard) pour retourner une estimation du nombre π .

Données :

Chaque item de `pts` contient un point. Un point est représenté par un tableau contenant exactement deux nombres, respectivement `x` et `y` tels que $0 \leq x \leq 1$ et $0 \leq y \leq 1$. `pts` n'est jamais null et contient toujours au moins un item.

Réponse

```
1  /**
2   * This class defines a method to approximate pi
3   */
4  class Pi {
5
6      /**
7       * Approximate pi using the given points.
8       */
9      static double approx(double[][] pts) {
10         double m = 0d;
11         double pi;
12
13         for(int i = 0; i < pts.length; i++){
14             if( (pts[i][0]*pts[i][0]) + (pts[i][1]*pts[i][1]) <= 1 ){
15                 m = m + 1;
16             }
17         }
18         pi = (m * 4) / pts.length;
19         return pi;
20     }
21 }
```

Résultat

- ✓ L'estimation de π est valide (liée aux points fournis)
Résolution de problèmes +257pts
- ✓ Le point P(1, 0) est à l'intérieur du quart de disque
Fiabilité +43pts

Question 25: Point de jointure



Java



07:41 / 30:00



17x (58 sec)



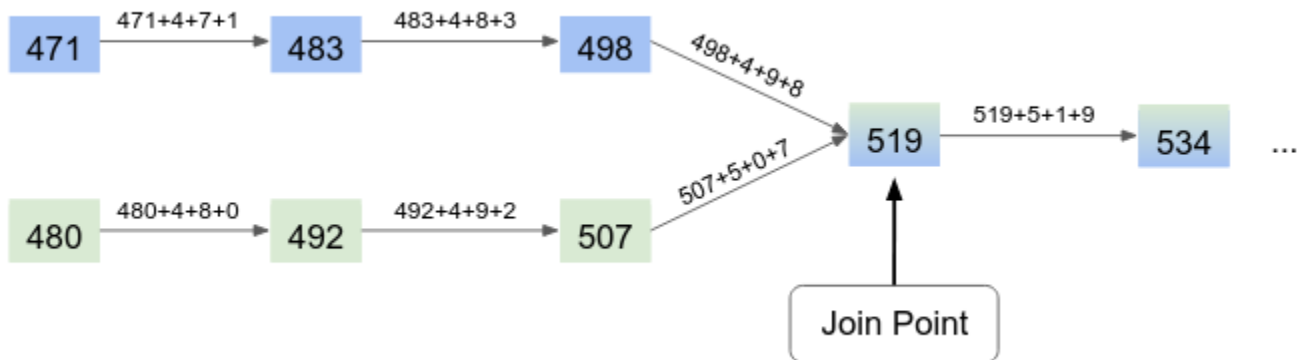
600 / 600 pts

? Question

On considère la suite de nombres dans laquelle un nombre est suivi du même nombre plus la somme de ces chiffres.

Par exemple 34 est suivi de 41 ($41 = 34 + (3 + 4)$). 41 est lui-même suivi de 46 ($46 = 41 + (4 + 1)$).

Deux séquences avec des points de départ différents peuvent finir par se rejoindre. Par exemple la séquence qui commence à 471 et la séquence qui commence à 480 partagent le nombre 519 (le point de jointure). Bien évidemment après le point de jointure, les séquences sont identiques.



Un exemple de deux séquences se rejoignant en 519.

Implémentez la méthode `int computeJoinPoint(int s1, int s2)` qui prend les points de départ de deux séquences et renvoie le point de jointure de ces deux séquences.

Contraintes:

Les deux séquences se rejoignent toujours $0 < s1, s2 < 20000000$ $0 < \text{joint point} < 20000000$



Réponse

```
1  import java.util.*;
2  import java.io.*;
3  import java.math.*;
4
5  class Solution {
6
7      public static int computeJoinPoint(int s1, int s2) {
8          // Write your code here
9          // To debug: System.err.println("Debug messages...");
10         if(s1 == 0 || s2==0 || s1>=20000000 || s2>=20000000) {
11             return 0;
12         }
13         while(s1 != s2) {
14             if (s1<s2) {
15                 s1 = computeSommePoint(s1);
16             } else {
17                 s2 = computeSommePoint(s2);
18             }
19         }
20         return s1;
21     }
22
23     public static int computeSommePoint(int s) {
24         int somme = s;
25         for(char c: String.valueOf(s).toCharArray()) {
26             somme += Character.getNumericValue(c);
27         }
28         return somme;
29     }
30
31     /* Ignore and do not change the code below */
32     // #region main
33     public static void main(String args[]) {
34         Scanner in = new Scanner(System.in);
35         int s1 = in.nextInt();
36         int s2 = in.nextInt();
37         PrintStream outStream = System.out;
38         System.setOut(System.err);
39         int res = computeJoinPoint(s1, s2);
40         System.setOut(outStream);
41         System.out.println(res);
42     }
43     // #endregion
44 }
```

Résultat

- ✓ $s1 < s2$
Résolution de problèmes +65pts
- ✓ $s2 > s1$
Résolution de problèmes +65pts
- ✓ Jointure rapide
Résolution de problèmes +65pts
- ✓ Plus grande jointure
Résolution de problèmes +65pts
- ✓ Jointure asymétrique 1
Résolution de problèmes +65pts
- ✓ Jointure asymétrique 2
Résolution de problèmes +65pts
- ✓ Nombres premiers
Résolution de problèmes +70pts
- ✓ Grands nombres
Résolution de problèmes +70pts
- ✓ Même point de départ
Fiabilité +70pts

Question 26: Détection de boucles



Java



08:40 / 25:00

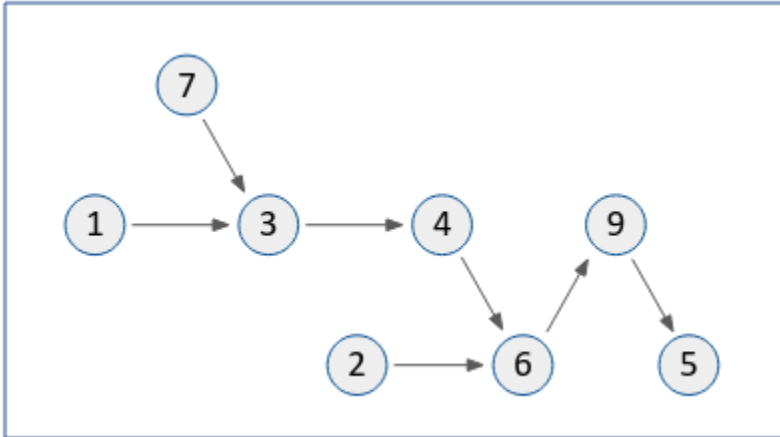


0 / 450 pts

? Question

L'objectif de cet exercice est de trouver le noeud terminal d'un réseau simple.

Dans ce réseau simple, chaque noeud est lié à au plus un noeud sortant de manière unidirectionnelle.



Un exemple de réseau simple

Implémentez la méthode `findNetworkEndpoint(startNodeId, fromIds, toIds)` qui retourne l'id du dernier noeud du réseau trouvé en partant du noeud `startNodeId` et en suivant les liens du réseau.

Dans l'exemple ci-dessus, le noeud terminal en partant du noeud n°2 (ou de n'importe quel noeud) est le noeud n°5.

`fromIds` et `toIds` sont deux tableaux de la même longueur qui décrivent les liens unidirectionnels du réseau (`fromIds[i]` est lié à `toIds[i]`). Dans l'exemple ci-dessus, `fromIds` est :

```
[1, 7, 3, 4, 2, 6, 9]
```

et `toIds` est :

```
[3, 3, 4, 6, 6, 9, 5]
```

Dans le cas où vous rencontrez une boucle lors du parcours du réseau, la méthode doit renvoyer l'id du dernier noeud traversé avant de clore la boucle.

Contraintes:

0 < nombre de liens < 10000 Un noeud ne peut pas être directement lié à lui-même.



Réponse

```
1 import java.util.*;
2 import java.io.*;
3 import java.math.*;
4
5 class Solution {
6
7     public static int findNetworkEndpoint(int startNodeId, int[] fromIds, int[] toIds) {
8         // Write your code here
9         // To debug: System.err.println("Debug messages...");
10
11         return -1;
12     }
13
14     /* Ignore and do not change the code below */
15     // #region main
16     public static void main(String args[]) {
17         Scanner in = new Scanner(System.in);
18         int startNodeId = in.nextInt();
19         int n = in.nextInt();
20         int[] fromIds = new int[n];
21         for (int i = 0; i < n; i++) {
22             fromIds[i] = in.nextInt();
23         }
24         int[] toIds = new int[n];
25         for (int i = 0; i < n; i++) {
26             toIds[i] = in.nextInt();
27         }
28         PrintStream outStream = System.out;
29         System.setOut(System.err);
30         int endPointId = findNetworkEndpoint(startNodeId, fromIds, toIds);
31         System.setOut(outStream);
32         System.out.println(endPointId);
33     }
34     // #endregion
35 }
```

Résultat

-  Réseau simple
Résolution de problèmes ~~+85pts~~
-  Grand réseau linéaire
Résolution de problèmes ~~+70pts~~
-  Simple boucle
Résolution de problèmes ~~+70pts~~
-  Réseau en arbre
Résolution de problèmes ~~+70pts~~
-  Cercle
Résolution de problèmes ~~+65pts~~
-  Réseaux disjoints
Résolution de problèmes ~~+65pts~~
-  Noeud isolé
Fiabilité ~~+25pts~~

Glossaire

Connaissance du langage

La mesure de cette compétence permet de déterminer l'expérience du candidat dans la pratique d'un langage de programmation. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui devra être rapidement opérationnel.**

Modélisation

Cette mesure fournit une indication sur la capacité du candidat à appliquer des solutions standard pour résoudre des problèmes récurrents. Un développeur ayant un bon niveau dans cette compétence augmentera la qualité (maintenabilité, évolutivité) de vos applications. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui sera amené à travailler sur les briques qui structurent vos applications, à anticiper les besoins de demain pour développer des solutions pérennes.**

Résolution de problèmes

Cette compétence correspond aux aptitudes du candidat à comprendre et à structurer son raisonnement pour trouver des solutions à des problèmes complexes. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vos applications ont une composante technique importante (R&D, innovation).**

Fiabilité

La fiabilité caractérise la capacité du candidat à réaliser des solutions qui prennent en compte les cas particuliers. Plus cette compétence est élevée, plus vos applications sont robustes (moins de bugs).