

**Sujet d'épreuves de la 48^e Compétition Nationale
des Métiers**

MÉTIER N°09 SOLUTIONS LOGICIELLES EN ENTREPRISE

Module 5

Soumis par :

Paul NGO, Expert WorldSkills France

Geoffrey DA ENCARNACAO, Expert Adjoint WorldSkills France

EXPLICATION DU SPEED MODULE

DUREE TOTALE DE L'ÉPREUVE	1 heure
DIFFUSION DU SUJET	Découvert le jour de la compétition

1. INTRODUCTION

Le Speed module est conçu pour tester vos capacités d'analyse, de résolution de problèmes et d'implémentation rapide sous contrainte de temps. Il est composé de 4 épreuves indépendantes, visant à évaluer :

- la rapidité d'exécution
- l'agilité intellectuelle
- la réactivité algorithmique

Objectifs

- Résoudre un maximum d'épreuves dans un temps imparti.
- Manipuler des fichiers de données structurées (.json).
- Échanger avec un système de validation automatisé via des endpoints HTTP.
- Produire des réponses au format JSON strictement conforme.

Affichage dynamique en temps réel

- Les résultats seront projetés en direct sur un écran public. L'état d'avancement sera visible avec un code couleur explicite.

Informations pratiques

- Vous avez 4 épreuves indépendantes à résoudre.
- Aucun accès à Internet n'est autorisé pendant l'épreuve.
- Toutes les données sont fournies dans le dossier **C:\Users\<VotreNom>\Bureau\ressources**.
- Chaque réponse doit être soumise au format JSON via une requête POST vers l'endpoint indiqué dans chaque épreuve. Identifiez-vous avec votre **student_id**.
- Réponse du serveur :

```
{
  "valid": true // ou false
}
```

- Base URL du serveur : **http://localhost:8000**

2. ÉPREUVES

Épreuve 1 : Tri rapide (QuickSort)

- **Fichier à charger** : epreuve1_data.json
- **Objectif** : Trier 10 000 entiers entre 1 et 1 000 000.
- **Afficher** les 10 plus petits.
- **Endpoint** : POST /epreuve1/
- **Format JSON attendu** :

```
{
  "student_id": "abc123",
  "result": [12, 34, 56, 78, 90, 123, 145, 200, 212, 300]
}
```

- **Validation** : L'affichage passe au vert dès réussite.

Épreuve 2 – Statistiques (moyenne, médiane, variance)

- **Fichier à charger** : epreuve2_tailles.json (Données triées par ordre croissant)
- **Objectif** : Calculer la moyenne, la médiane et la variance d'une liste de 1000 tailles.
- **Arrondir** à deux décimales.
- **Endpoint** : POST /epreuve2/
- **Format JSON attendu** :

```
{
  "student_id": "abc123",
  "moyenne": 172.53,
  "mediane": 171.00,
  "variance": 36.28
}
```

- **Validation** : L'affichage passe au vert dès réussite.
- **Rappel des formules** :

- **Moyenne**

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- **Médiane**

La médiane est une mesure de tendance centrale utilisée en statistique pour décrire la valeur "centrale" d'une série de données. Voici comment procéder :

- a) Trier la liste par ordre croissant. (epreuve2_tailles.json est déjà trié par ordre croissant)
- b) Identifier le nombre d'éléments :
 - Si le nombre de données (n) est impair, la médiane est la valeur du milieu.
Ex. : Données = [2,4,5,7,10] \Rightarrow La médiane est 5.
 - Si le nombre de données (n) est pair, la médiane est la moyenne des deux valeurs centrales.
Ex. : Données = [2,4,5,7] \Rightarrow La médiane est 4.5 ; $\frac{(4+5)}{2} = 4.5$

- Variance de population

La variance mesure à quel point les données d'un ensemble complet s'écartent de leur moyenne.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Où :

- x_i : chaque valeur individuelle
- \bar{x} : la moyenne des valeurs
- n : le nombre total de valeurs

Voici comment procéder :

- a) Trier la liste par ordre croissant. (epreuve2_tailles.json est déjà trié par ordre croissant)
- b) Calculer la moyenne :

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

- c) Calculer l'écart à la moyenne :

$$(x_i - \bar{x})$$

- d) Faire la somme des carrés :

$$\sum_{i=1}^n (x_i - \bar{x})^2$$

- e) Diviser par n .

Exemple complet :

Jeu de données :

[1, 3, 5, 7, 9]

Moyenne :

$$\bar{x} = \frac{1 + 3 + 5 + 7 + 9}{5} = \frac{25}{5} = 5$$

Écart à la moyenne :

x_i	$(x_i - \bar{x})$
1	$1 - 5 = -4$
3	$3 - 5 = -2$
5	$5 - 5 = 0$
7	$7 - 5 = 2$
9	$9 - 5 = 4$

Carré des écarts :

x_i	$(x_i - \bar{x})^2$
1	$(1 - 5)^2 = 16$
3	$(3 - 5)^2 = 4$
5	$(5 - 5)^2 = 0$
7	$(7 - 5)^2 = 4$
9	$(9 - 5)^2 = 16$

Sommes des carrés :

$$16 + 4 + 0 + 4 + 16 = 40$$

Variance :

$$\sigma^2 = \frac{40}{5} = 8$$

Résultat final : Variance = 8

Épreuve 3 – Conversion en chiffres romains

- **Fichier à charger :**
 - epreuve3_baseromains.json
 - epreuve3_nombresaconvertir.json
- **Objectif :** Convertir une série d'entiers naturels en nombres romains en respectant les règles classiques d'écriture, et soumettre l'ensemble des résultats dans une seule requête.
- **Endpoint :**
 - POST /epreuve3/
 - Soumettre la conversion.

```
{
  "student_id": "candidat_id",
  "conversions": [
    {"nombre": 1450, "chiffre_roman": "MCDL"},
    {"nombre": 2024, "chiffre_roman": "MMXXIV"},
    {"nombre": 3999, "chiffre_roman": "MMMCMXCIX"},
    // ... 15 objets au total
  ]
}
```

Principe de base :

Lettre	Valeur
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Règles d'écriture :

- Addition : si une lettre est suivie d'une lettre de valeur inférieure ou égale, on les additionne.

Ex. : VI = 5 + 1 = 6

- Soustraction : si une lettre est suivie d'une lettre de valeur supérieure, on les soustrait.

Ex. : IV = 5 - 1 = 4

- Les symboles sont combinés de manière à exprimer le nombre de façon la plus concise et correcte possible.

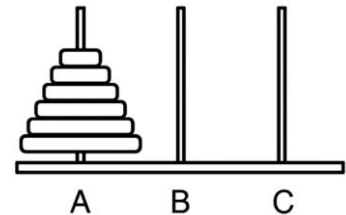
Épreuve 4 – Tour d'Hanoï

Présentation :

La Tour de Hanoï est un jeu de réflexion mathématique inventé par le mathématicien Édouard Lucas en 1883. C'est un casse-tête qui consiste à déplacer des disques d'une tige à une autre en suivant des règles précises.

Règles du jeu

- Vous disposez de 3 tiges verticales (appelées A, B, et C)
- Tous les disques sont initialement empilés sur la tige d'origine « A », du plus grand (en bas) au plus petit (en haut).
- Votre objectif est de déplacer tous les disques vers la tige cible « C », en utilisant la tige auxiliaire « B » comme tige intermédiaire.



Contraintes :

- Un seul disque à la fois peut être déplacé.
- Un disque ne peut jamais être posé sur un plus petit.
- La tige auxiliaire doit être utilisée si nécessaire.
- Le nombre de mouvements doit être optimal :
 - Nombre minimal de déplacement = $2^n - 1$
 - Pour 5 disques, cela donne 31 mouvements

- **Fichier à charger** : epreuve4_hanoi.json

```
{
  "disques": 5,      // Nombre de disques à déplacer
  "origine": "A",    // Tige de départ (où sont tous les disques)
  "cible": "C",      // Tige d'arrivée (où doivent finir tous les disques)
  "auxiliaire": "B"  // Tige intermédiaire (pour les déplacements temporaires)
}
```

- **Objectif** : Générer les mouvements pour déplacer 5 disques de la tour A à la tour C.
- **Endpoint** : POST /epreuve4/
- **Format JSON attendu** :

```
{
  "student_id": "abc123",
  "result": [
    ["A", "C"],
    ["A", "B"],
    ["C", "B"],
    "... (28 autres mouvements) ..."
  ]
}
```

- « result » doit contenir la séquence complète, ordonnée, et exacte de tous les déplacements effectués.
- Chaque déplacement est une chaîne de caractères indiquant le mouvement d'une tige source vers une tige cible, sous forme ["A", "C"].