

Olympiades d'Informatique de l'Océan Indien

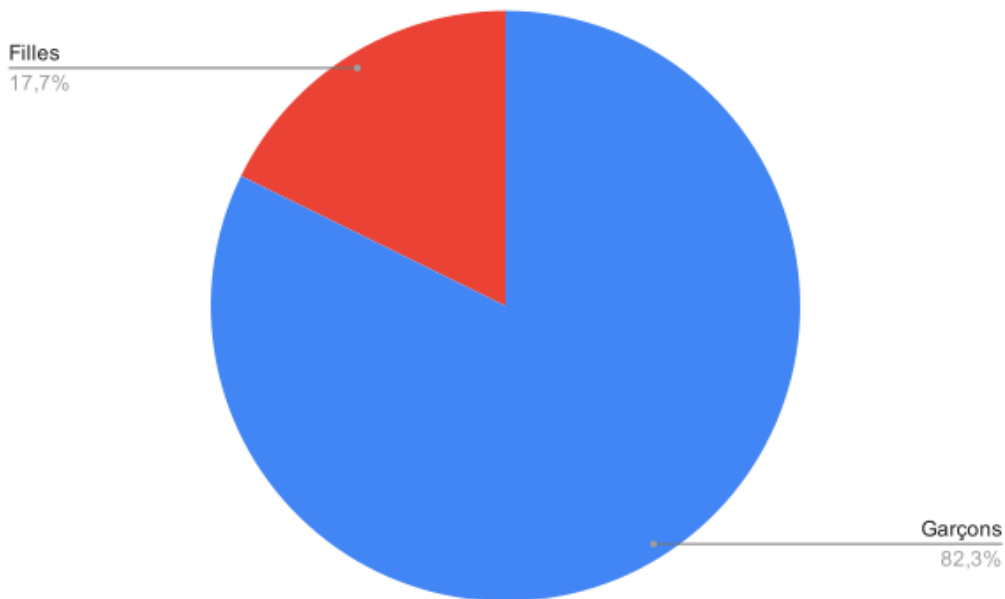
Édition 2026

1

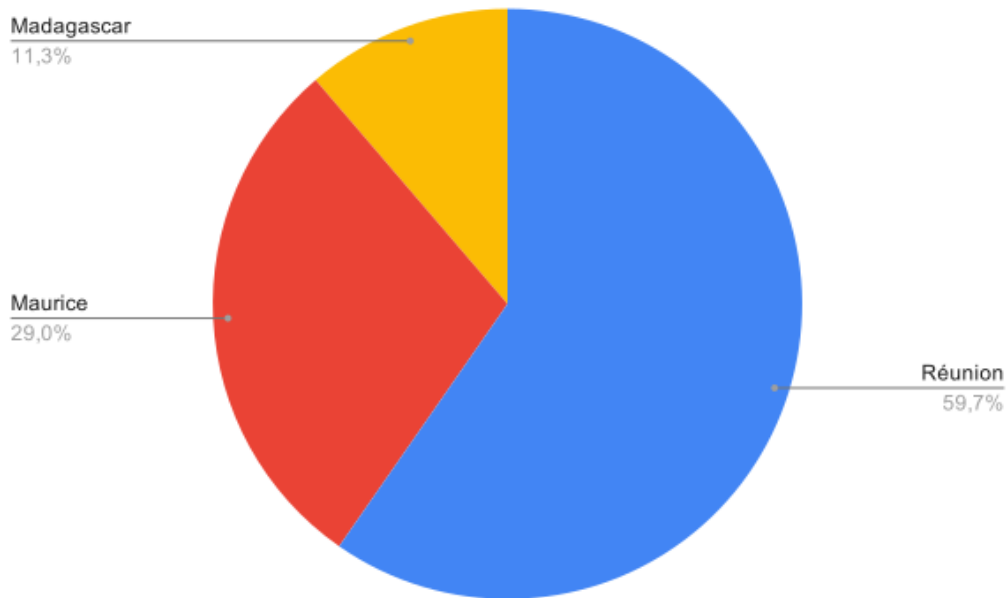
Statistiques

Les participants

- 62 participants : 11 filles et 51 garçons



■ 10 lycées dont 2 de l'île Maurice et 1 de Madagascar

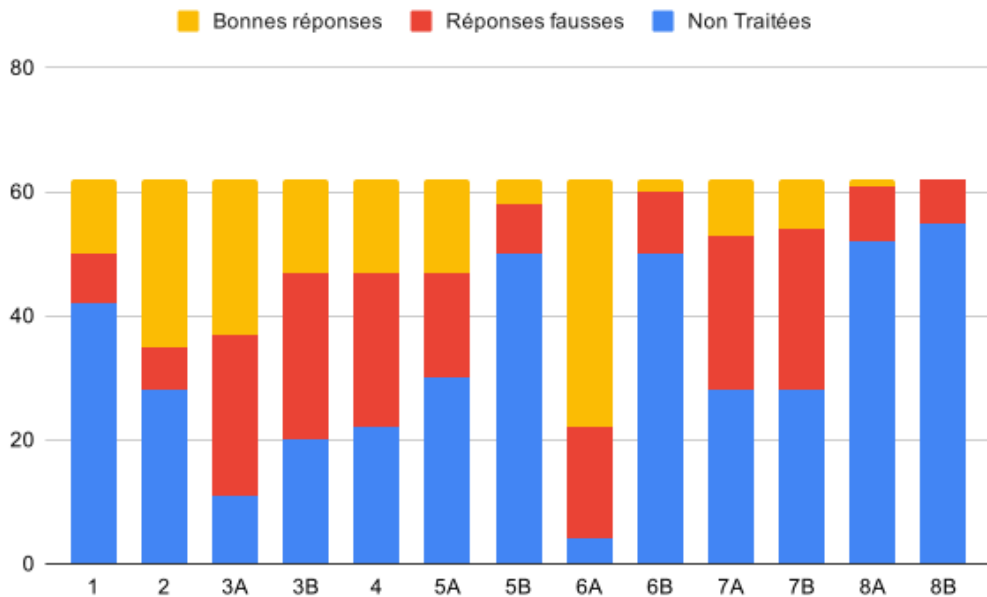


Le sujet

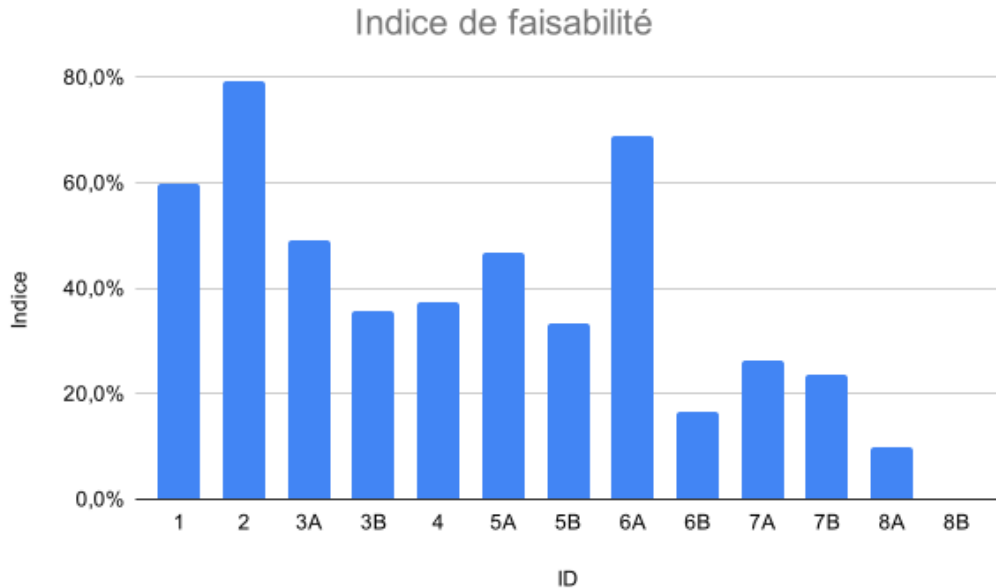
- 8 énigmes : 13 questions
- Entièrement résolubles avec les connaissances de 1ère NSI
 - ▶ Memento python fourni aux élèves pour l'épreuve
- Difficulté progressive des exercices
- Volonté de susciter une démarche de réflexion
 - ▶ étude et modélisation d'une situation
 - ▶ résolution algorithmique
 - ▶ mise en œuvre en Python
- **Nouveauté** : la réponse attendue est toujours 1 mot ou 1 valeur numérique.
 - ▶ Faciliter la transmission des réponses
 - ▶ Éviter les erreurs de format

Les réponses

■ Réponses reçues



■ Faisabilité des énigmes (bonnes reponses / nombre de tentatives)



2

Exemples d'énigmes

Facile : l'auteur caché

- On donne un fichier source Python contenant une longue chaîne de caractères :

```
mystere = "[6 Lcan <:8sSc@akiTeLM (...) ngu]ottnWjghWK4c50A6h"
```

- On demande de reformer le texte obtenu en ne prenant que les caractères dont l'indice est un nombre premier (définition rappelée)
- Difficultés
 - ▶ Décomposer en sous-problèmes : écrire d'abord une fonction pour tester si un nombre est premier.
 - ▶ Gérer la reconstruction dynamique du texte déchiffré (l'opération append sur les listes n'étant pas au programme de 1ère NSI) grâce à la concaténation de chaînes de caractères

L'auteur caché : mise en œuvre en Python

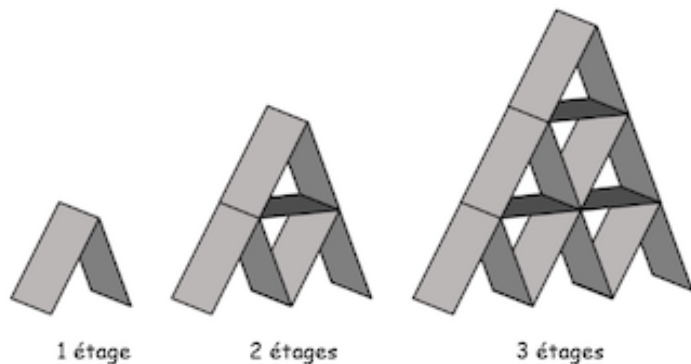
```
def est_premier(n):  
    if n < 2:  
        return False  
    for k in range(2,n):  
        if n % k == 0:  
            return False  
    return True
```

```
N = len(mystere)  
texte = ""  
for i in range(N):  
    if est_premier(i):  
        texte = texte + mystere[i]  
print(texte)
```

■ On trouve :

La science est ce que nous comprenons suffisamment bien pour l'expliquer à un ordinateur. L'art, c'est tout ce que nous faisons d'autre. signe : Knuth

Moyenne : le château de cartes



Le record du monde est de 54 étages de cartes.

1. Combien a-t-il fallu de cartes pour le réaliser ?
2. Avec 2026 cartes, quel est le plus grand château réalisable et combien de cartes sont laissées de côté ?

Château de cartes : analyse avec un tableau

Un exemple d'analyse possible :

étages	cartes	largeur	pilliers RDC
1	2	1	2
2	$7 = 2 + 1 + 4$	2	4
3	$15 = 7 + 2 + 6$	3	6
4	$26 = 15 + 3 + 8$	4	8

etc...

- Trouver une équation de récurrence était aussi une possibilité

Mise en œuvre en Python

```
# Initialisation
cartes = 2
largeur = 1
pilliers = 2

for etage in range(2, 55):
    pilliers = pilliers + 2
    cartes = cartes + largeur + pilliers
    largeur = largeur + 1

print(cartes)
```

- Pour la seconde question on utilise une boucle `while` au lieu d'une boucle `for` pour arrêter la construction dès que le nombre de cartes restantes ne suffit plus.

```
# Initialisation
```

```
cartes = 2
```

```
largeur = 1
```

```
pilliers = 2
```

```
etage = 1
```

```
while cartes <= 2026:
```

```
    etage = etage + 1
```

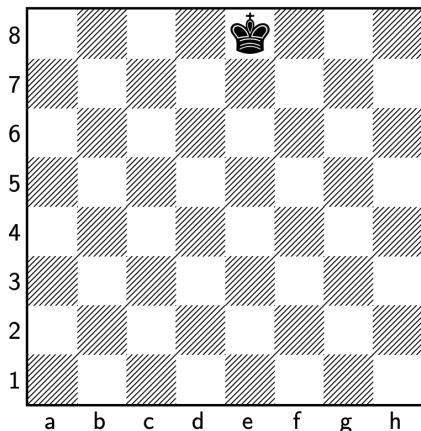
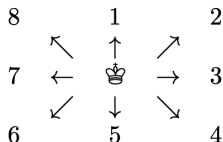
```
    pilliers = pilliers + 2
```

```
    cartes = cartes + largeur + pilliers
```

```
    largeur = largeur + 1
```

```
print(etage - 1)
```

Difficile : le roi en balade



- Une suite de 100 déplacements de roi est donnée sous forme d'une liste d'entiers.
- **Q1** : combien de cases de l'échiquier on été explorées ?
- **Q2** : quelle est la case la plus visitée (on garantit qu'il n'y en a qu'une)
- Difficultés :
 - ▶ Problème à 2 dimensions
 - ▶ Gérer les sorties d'échiquier
 - ▶ Structure matricielle : utilisation d'un dictionnaire.

Mise en œuvre en Python

```
pas = {1: (0, 1), 2: (1, 1), 3: (1, 0), 4: (1, -1), 5: (0, -1),  
6: (-1, -1), 7: (-1, 0), 8: (-1, 1)}
```

```
# Position initiale
```

```
posx = 4
```

```
posy = 7
```

```
# Initialisation du dictionnaire
```

```
visites = {}
```

```
for x in range(8):
```

```
    for y in range(8):
```

```
        visites[(x, y)] = 0
```

```
# Balade
```

```
for p in CENT_PAS:
```

```
    (dx, dy) = pas[p]
```

```
    newx = posx + dx
```

```
    newy = posy + dy
```

```
    if (0 <= newx and newx < 8 and 0 <= newy and newy < 8):
```

```
        posx, posy = newx, newy
```

```
        visites[(posx, posy)] += 1
```

■ Après quoi, les questions s'abordent facilement.

Merci !

- Nous remercions tous les participants, professeurs encadrants et organisateurs sans qui ces olympiades n'existeraient pas !
- Rendez-vous en 2027 !