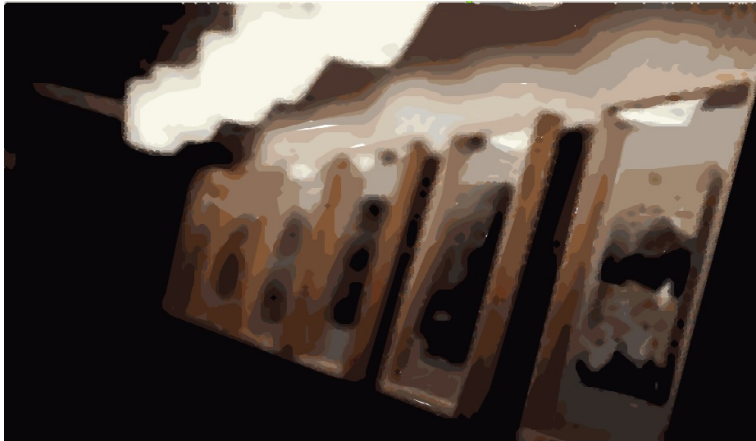


Challenge *Élyseum*  
*Une Exposition Optimisée pour des T.A.*  
Semaine Interchallenges  
Algorithmique + Programmation En C  
S.I-A.P.E.C.  
— ESIEA — 1A — 2022-2023  
LAVAL + PARIS

R. ERRA  
& G. HEISS (merci pour la plate-forme!)  
& et l'équipe INF1031

Janvier 2023



- 1 Introduction
  - Le contexte
  - Votre mission

## Le contexte :

- ❶ Les *Intelligences Artificielles* (*I.A.* ) "à la retraite" se trouvent dans un endroit baptisé *Élyseum* sur la planète *Asimov*
- ❷ Pour les occuper, un challenge est organisé régulièrement, challenge adapté aux *I.A.* orientées **Algorithmique**
- ❸ Ces *I.A.* ont un goût particulièrement prononcé pour les problèmes **difficiles** mais **fun**.
- ❹ Pour satisfaire ces *I.A.* , vous devez suivre (exactement) certaines règles (voir la suite)
- ❺ Une des ces règles, depuis la promulgation des lois 4, 5 et 6 de la "robotique", qui complète les 3 lois de la robotique édictées par *Asimov* est qu'au moins un humain doit **concourir** !
- ❻ Votre mission, si vous l'acceptez (figure de style), consiste à participer au challenge.
- ❼ Bonne chance !

- 1 Introduction
  - Le contexte
  - Votre mission

## Votre mission :

- 1 Vous allez travailler pour une galerie virtuelle qui "recevra" une exposition de peintures numériques demain soir
- 2 Vous donc devez "préparer" la galerie pour demain soir (18h)
- 3 Vous devez pour cela "ordonner" les "peintures numériques" dans un couloir virtuel où les *I.A.* pourront les "regarder".
- 4 Pour satisfaire les *I.A.* , et essayer de les battre, il faut maximiser un critère : le *Score Global De Satisfaction* des *I.A.* !
- 5 Ce *Score Global De Satisfaction* se définit à partir du *Score Local De Satisfaction*

## 2 Description du Problème

- Le goût des  $\mathcal{I.A.}$
- Cadre virtuel

## *I.A.*?

- En vertu de la 4ème loi de la "robotique" (qui on vous le rappelle reste inconnue des humains et connue seulement par les *I.A.* ) : on ne discute pas cette règle des *nombres clés* !
- Il y a deux types de peintures :
  - 1 "Paysage" (Landscape)
  - 2 "Portrait" (Portrait).
- Dans la suite on prendra :
  - 1 "L" pour "Landscape" (Landscape)
  - 2 "P" pour "Portrait" (Portrait).
- Toutes les images, quel que soit leur type, ont des *nombres clés*. Le mystère reste complet quant à leur signification.



## *I.A.* ?

- ➊ À chaque photo numérique, sont associés des "*nombre*s clés" que seule les *I.A.* apprécient.
- ➋ Ces *nombre*s clés ne sont en fait que des nombres entiers ( $\geq 0$ ) qui représentent quelque chose pour les *I.A.*
- ➌ Ainsi si à la photo d'indice 0 on associe les entiers 100, 200 et 300, même si pour nous cela ne représente rien, pour les *I.A.* cela veut dire quelque chose.
- ➍ Pour simplifier, dans un fichier il n'y aura que des photos avec le même nombre de "*nombre*s clés" :
  - ➊ Le nombre maximum de *nombre*s clés est de 50
  - ➋ Le nombre maximum de photos est de 1000000 (un million) (ce lundi)
  - ➌ Pour mardi ? *surprise*.

Exemple : à gauche un paysage (L) et à droite un portrait (P)



## 2 Description du Problème

- Le goût des  $\mathcal{I.A.}$
- Cadre virtuel

## Aie : ça se complique

- ❶ Pour des raisons que nous ne discuterons pas ici, les *I.A.* détestent les "portraits" (on devine pourquoi !)
- ❷ Et il y a donc une règle de plus : **les portraits doivent être associés par deux !**
- ❸ On définit ainsi un "cadre virtuel" (*cadre* dans la suite) :
  - Un cadre ne contient qu'une photo si celle-ci est de type "L" (paysage/landscape)
  - Un cadre doit impérativement contenir deux photos si celles-ci sont de type "P" (portrait).

- 3 Fichier "input"
  - Fichiers d'entrainement

Pour vous aider à trouver le meilleur algorithme d'ordonnancement on vous propose plusieurs fichiers d'entrainement avec toutes les informations nécessaires sur les "peintures numériques" :

- Chaque fichier d'input (le "jeu de données") est un fichier texte (uniquement des caractères ASCII)
- Toutes les lignes se terminent par un unique '\n' caractère (UNIX- style line endings).
- La 1ère ligne contient un entier  $N$  : c'est le nombre de photos du jeu de données ( $1 \leq N \leq 10^6$ ) que vous devez traiter.

On rappelle que chaque *nombre clé* est un entier. L'ordre des *nombres clés* n'a aucune importance.

Suite)

- Après la 1ère ligne, **suivent  $N$  lignes** :
- La  $i$ ème ligne  $i$  contient une description de la peinture avec au début son **identifiant** qui est un entier  $i$  ( $0 \leq i < N$ )
- Suivent ensuite **sur la même ligne (séparés par un espace)** :
  - 1 Un caractère "L" si la photo est un paysage/landscape
  - 2 Ou un caractère 'P' si c'est un portrait.
  - 3 Ensuite : Un entier  $M_i$  ( $1 \leq M_i \leq 100$ ) : c'est le **nombre de nombres clés** pour cette peinture.
  - 4 Suivent  $M_i$  entiers : ce sont les *nombres clés* de la peinture  $i$ .

*Rappel : L'ordre des nombres clés n'est pas important.*

### Un exemple naïf de fichier "input"

Les commentaires ne font évidemment pas partie du fichier

```
4 // Nombre de photos
L 3 1 2 3 // Photo de type L avec 3 nombres clés
P 4 12 13 14 15 // Photo de type P avec 4 nombres clés
P 3 4 5 6 // Photo de type P avec 2 nombres clés
L 2 10 1 // Photo de type L avec 2 nombres clés
```



### Un exemple naïf de fichier "input"

Un exemple de "vrai" fichier (même nombre de *nombres clés* par photo), tels que ceux que vous aurez sur la plate-forme :

4

L 4 1 2 13 17

P 4 12 13 14 15

P 4 14 5 6 16

L 4 10 12 19 20

## Fichiers d'input et d'output ?

Faites simple ! Ne perdez pas de temps avec les fichiers :

- ➊ Nous vous recommandons fortement d'utiliser les **redirections**
- ➋ Rappel : < pour le fichier d'entrée , > pour le fichier de sortie !
- ➌ Exemple :
- ➍ `./a.out < 100.input > 100.output`

- 4 Fichiers d'input et d'output ?
  - Fichier de sortie/output
  - Exemple de fichier de sortie

Le fichier de sortie de votre code a une structure simple :

- ❶ La 1ère ligne c'est  $C$  le nombre de "cadre" de votre fichier output/résultat (donc  $C \leq$  au nombre de photos du fichier d'input)
- ❷ Cette 1ère ligne sera suivie de  $C$  lignes
- ❸ Chacune des ces lignes comporte soit un entier (indice de la photo de type  $L$ )
- ❹ soit deux entiers (séparés par un espace) : les indices des deux portraits "P" que vous avez appariés.
- ❺ chaque ligne sera suivi d'un retour à la ligne (soit :le caractère '\n' — UNIX- style line endings)
- ❻ Chaque "peinture" ne doit évidemment apparaître qu'une seule fois ! sinon 0 points.

- 4 Fichiers d'input et d'output ?
  - Fichier de sortie/output
  - Exemple de fichier de sortie

Un exemple de fichier de output/sortie :

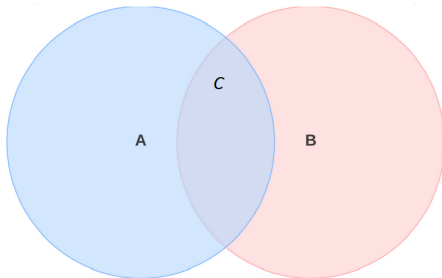
F<sub>0</sub>F<sub>1</sub>F<sub>2</sub>

Fichier soumis	Explications
3	3 "cadres"
0	1er "cadre": photo 0
1 2	2ème "cadre": photos 1 et 2
3	3ème "cadre": photo 3

Un exemple de fichier "soumis"

- 5 Score Local De Satisfaction et Score Global De Satisfaction
  - Calcul du *Score Local De Satisfaction*
  - Score Global De Satisfaction

# Calcul du *Score Local De Satisfaction*



On définit **3 sous-ensembles** à partir de deux images :

- 1 A : *nombres clés* dans image de gauche mais pas dans image de droite
- 2 B : *nombres clés* dans image de droite mais pas dans image de gauche
- 3 C : *nombres clés* dans image de gauche ET dans image de droite



## Score Local De Satisfaction

On définit maintenant **trois entiers** :

- ①  $\mathcal{A} = |A|$  : nombre de *nombres clés* dans image de gauche mais pas dans image de droite
- ②  $\mathcal{B} = |B|$  : nombre de *nombres clés* dans image de droite mais pas dans image de gauche
- ③  $\mathcal{C} = |C|$  : nombre de *nombres clés* dans image de gauche ET dans image de droite

Le *Score Local De Satisfaction* des  $\mathcal{I.A.}$  est alors défini par :

$$Score_{local} = \min(\mathcal{A}, \mathcal{B}, \mathcal{C}).$$

## Exemple de calcul du *Score Local De satisfaction*

- ❶ *nombres clés* de images de gauche : 100, 200, 300
- ❷ *nombres clés* de images de droite : 200, 400, 500
- ❸ Donc  $\mathcal{A} = 2$  car seul 200 fait partie des *nombres clés* de de l'image de gauche et de celle de droite mais ni 100 ni 300
- ❹  $\mathcal{B} = 2$  car seul 200 fait partie des *nombres clés* de l'image de gauche et de celle de droite mais ni 400 ni 500
- ❺  $\mathcal{C} = 1$  car 200 fait partie des *nombres clés* de l'image de gauche et de celle de droite.

Le *Score Local De Satisfaction* des  $\mathcal{I.A.}$  est donc donné par :

$$Score_{local} = \min(\mathcal{A}, \mathcal{B}, \mathcal{C}) = \min(2, 1, 2) = 1$$

- 5 Score Local De Satisfaction et Score Global De Satisfaction
  - Calcul du *Score Local De Satisfaction*
  - Score Global De Satisfaction

- 1 Le **Score Global De Satisfaction** se calcule simplement :
- 2 On ajoute les **Score Locaux de Satisfaction**
- 3 Ainsi, pour  $N$  "cadres" il y a  $N - 1$  Score Locaux de Satisfaction
- 4 ...dont la somme donne le **Score Global de Satisfaction**

- 6 Contraintes de temps et d'espace
  - Contraintes
  - Optimisation ?

## Contraintes de temps et d'espace

Pour éviter de surcharger la plate-forme inutilement

- ❶ Pour chaque fichier il y a un *timeout* de 1mn sur la plate-forme
- ❷ Pour l'ensemble des fichiers il y a un *timeout* de 15mn
- ❸ Vous êtes limité en RAM : 50Mo
- ❹ Votre code est testé sur chaque fichier "public" et sur chaque fichier "caché"
- ❺ Votre **score total** est juste la somme du score de chaque fichier
- ❻ ...disons, *si tout se passe bien*

- 6 Contraintes de temps et d'espace
  - Contraintes
  - Optimisation ?

## Quelques trucs (FORGET THE BRUTE FORCE ATTACK)

- ❶ L'exhibition est demain soir ! Nous avons besoin de votre aide !
- ❷ Trouver l'ordre "optimal" est un problème difficile, votre algorithme doit donc être rapide.
- ❸ Conseil : commencez avec une solution simple et rapide !
- ❹ ...puis, essayez d'optimiser mais pas trop car sinon vous serez bloqué par le timeout, c'est un Challenge !
- ❺ Utilisez des fonctions et procédures dans votre code !
- ❻ Écrivez des commentaires mais ne perdez pas trop de temps
- ❼ Essayez pas à pas d'optimiser vos résultats pour obtenir le meilleur score total possible.
- ❽ *Remarque : obtenir le "meilleur" Score Total (optimal) est un problème difficile, donc pensez "approximation".*



- 7 La plate-forme du challenge
  - Créez votre équipe

# Plate-forme du challenge

- 1 Vous devrez créer **une équipe** : 2 ou 3 élèves
- 2 Et non : pas 4 ni 5 etc.
- 3 Pour ce faire : un élève crée l'équipe [vous avez droit à un pseudo mais pas de trash !]
- 4 et les autres membres le rejoignent **après**

Et ce, avant 10h45

<https://sic.esiea.fr/>

## SIC 2023 - Elyseum

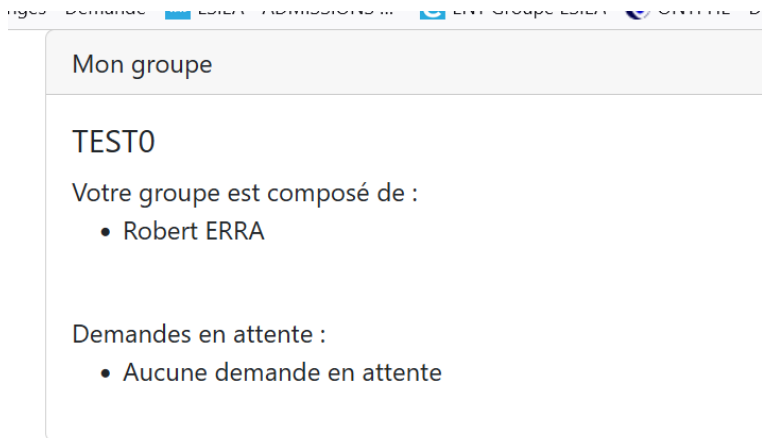
Classement

[Voir le classement complet](#)

Tentative

[Nouvelle tentative](#)

# Plate-forme du challenge : vérifiez que votre équipe existe



# Plate-forme du challenge : historique

## Mes dernières tentatives

- [29 janvier 2023 18:55](#)
- [29 janvier 2023 18:55](#)
- [29 janvier 2023 18:40](#)
- [29 janvier 2023 18:40](#)
- [29 janvier 2023 18:39](#)
- [29 janvier 2023 18:38](#)
- [29 janvier 2023 18:38](#)
- [29 janvier 2023 18:37](#)
- [29 janvier 2023 18:34](#)
- [29 janvier 2023 18:32](#)
- [29 janvier 2023 18:32](#)
- [29 janvier 2023 18:28](#)

## SIC 2023 - Elyseum

### Score

Score total : 6313 points

Date : 29 janvier 2023 18:55

Fichier : [codebidon\\_2bkCJJw.c](#)

Auteur : Robert ERRA

Equipe : [TEST0](#)

Campus :

Position dans le classement : Hors classement

[Accueil](#)[Nouvelle tentative](#)[Classement](#)[Retour](#)

10 photos

[Voir les données](#)

5 points

[Voir le résultat](#)

Done

# OK ? les tests en temps réel

5 points

Voir le résultat

Done

100 photos

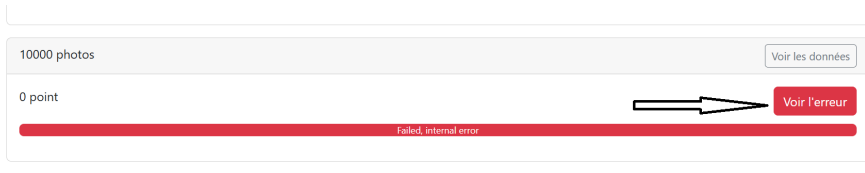
Voir les données

4 points

Voir le résultat

Done

# Une erreur ?





- 8 Avant de commencer
  - Votre Score Total Final
  - Dernières Remarques

## Votre Score Total Final

- ➊ Nous calculerons votre score pour chaque fichier du lundi
- ➋ De nouveaux fichiers d'entrée seront ajoutés entre le lundi et le mardi (par exemple : mardi il y aura des fichiers ayant des peintures de type "P" et "L" : votre code doit pouvoir le gérer
- ➌ Votre Score Total Final sera recalculé à chaque soumission mais la plate-forme garde toujours le meilleur résultat (Score Total)
- ➍ Ne *brute forcez* ni la plate-forme ni le problème, ça ne fonctionnera pas.
- ➎ Si vous insistez, votre score sera en fait divisé par 10 !

- 8 Avant de commencer
  - Votre Score Total Final
  - Dernières Remarques

## Dernières Remarques

- ❶ Un seul code (le dernier soumis) Lsera utilisé sur tous les fichiers
- ❷ Les fichiers seront soit "publics" soit "secrets"
- ❸ Fichiers "publics" : nombre de photos et de *nombres clés* connus
- ❹ Fichiers "secrets" : nombre de photos et de *nombres clés* inconnus
- ❺ Votre mission, si vous l'acceptez (c'est toujours une figure de style), consiste à participer au challenge. et à faire du mieux que vous pouvez.

Get the best score you can ! Obtenez le meilleur score possible !

...**Congratulations** et *bon courage* à toutes et à tous.

- 1 "Le défi de la programmation algorithmique est de créer des solutions efficaces avec des lignes de code minimalistes"
- 2 "La programmation algorithmique est un défi comme un labyrinthe, il faut trouver le chemin le plus court pour arriver à la sortie"
- 3 "Le défi en programmation algorithmique c'est de prendre des idées complexes et les traduire en instructions simples pour un ordinateur."
- 4 "La programmation algorithmique, c'est comme un match de boxe, il faut être rapide, précis et efficace pour gagner"

- ❶ "Le défi en programmation algorithmique est de simplifier la complexité pour créer des solutions élégantes."
- ❷ "La complexité est l'ennemi de la programmation algorithmique, mais c'est aussi ce qui rend le défi passionnant."
- ❸ "La programmation algorithmique est un défi qui consiste à prendre des idées complexes et à les rendre simples à comprendre et à exécuter."
- ❹ "Dans la programmation algorithmique, la complexité est le défi à relever et la simplicité est la récompense."