

# TP Placement aléatoire d'étudiants en salle d'examen

## Consignes pour le TP :

- Désactiver Copilot ou tout autre IA, pas d'utilisation d'IA autorisée.
  - Vous pouvez faire le TP en binôme => un seul repo Git nécessaire.
  - Créer un dossier TP1\_S2 dans votre répertoire Git du TP1.
  - Créer un fichier nommé main.c dans le dossier TP1\_S2.
  - Faire un **commit & push** de ce fichier
  - Vérifier sur Github que cela a fonctionné.
  - Faire régulièrement des **commit & push**. Vos **messages** de commit doivent être **explicite** !!
- 

Junia souhaite un outil permettant de générer automatiquement un plan de salle pour un examen. Les étudiants doivent être placés de façon aléatoire tout en respectant des objectifs de répartition :

- avoir le moins d'étudiants possible sur une même rangée
- espacer au maximum les étudiants entre eux (éviter les voisins directs quand c'est possible)

## Données fournies au programme

1. Un fichier texte .txt contenant la liste des élèves  
Format : une ligne par élève, sous la forme : prénom nom
2. Le nombre de rangées dans la salle
3. Le nombre de tables par rangée

Ces informations seront saisies par l'utilisateur lors de l'exécution du programme.

## Résultat attendu

### Sortie

Le programme doit afficher un plan de salle lisible, montrant pour chaque rangée et chaque table :

- soit l'élève affecté
- soit une case vide si aucune personne n'est placée

## Exemple d'affichage :

```
Entrez nb rangee : 5
Entrez nb table par rangee : 10
Entrez le chemin du fichier texte (il doit être au même niveau que l'exe): liste.txt
fichier : liste.txt

Liste des eleves:
1) Florian BAILLY
2) Jed BENSLAMA
3) Mathieu BERTRON
4) Louis BLANC
5) Louis BORDELAIS
6) Lucas COELHO
7) Amal EL
8) Louis GUERIN
9) Malo HAMON
10) Adam JAKUBOWSKI
11) Wilhelm MACAIGNE
12) Hatem MARHRAOUI
13) Anatole MAURIAC
14) Alexandre MOULIS
15) Edgar PULLES
16) Anaëlle SOUINI-RENAUDON
17) Lucas VERRON

Valeurs saisies:
Nb rangee : 5
Nb table par rangee : 10
Nb étudiants : 17

      1      2      3      4      5      6      7      8      9      10
1 Alexandre MO      X      X      Lucas CO      X      Wilhelm MA      X      X      Adam JA
2 Louis BO      X      X      Anatole MA      X      Jed BE      X      X      Anaëlle SO
3 Malo HA      X      X      X      Lucas VE      X      X      Hatem MA      X
4 Amal EL      X      X      X      Louis BL      X      X      Florian BA      X
5 Mathieu BE      X      X      X      Edgar PU      X      X      Louis GU      X

Entrez nb rangee : 4
Entrez nb table par rangee : 9
Entrez le chemin du fichier texte (il doit être au même niveau que l'exe): liste.txt
fichier : liste.txt

Liste des eleves:
1) Florian BAILLY
2) Jed BENSLAMA
3) Mathieu BERTRON
4) Louis BLANC
5) Louis BORDELAIS
6) Lucas COELHO
7) Amal EL
8) Louis GUERIN
9) Malo HAMON
10) Adam JAKUBOWSKI
11) Wilhelm MACAIGNE
12) Hatem MARHRAOUI
13) Anatole MAURIAC
14) Alexandre MOULIS
15) Edgar PULLES
16) Anaëlle SOUINI-RENAUDON
17) Lucas VERRON

Valeurs saisies:
Nb rangee : 4
Nb table par rangee : 9
Nb étudiants : 17

      1      2      3      4      5      6      7      8      9
1 Malo HA      X      Adam JA      X      Lucas CO      X      Mathieu BE      X      Wilhelm MA
2 Hatem MA      X      X      Amal EL      X      X      Anaëlle SO      X      X
3 Jed BE      X      X      Louis BO      X      X      Anatole MA      X      X
4 Alexandre MO      X      X      Louis BL      X      X      Louis GU      X      X
```

## Objectifs de placement

Votre placement doit être :

1. Aléatoire : deux exécutions successives ne doivent pas forcément donner le même plan.
2. Réparti : éviter de remplir une rangée pendant qu'une autre est vide, si c'est possible.
3. Espacé : éviter de placer deux élèves sur deux tables adjacentes dans une même rangée, si des alternatives existent.

## Contraintes techniques obligatoires

### Structures

Vous devez créer 2 structures :

1. **Eleve** contenant au minimum *prenom* et *nom*
2. **Place** contenant au minimum *numéro de rangée*, *numéro de table* et l'indicateur *occupée*

Vous devrez manipuler ces structures via des tableaux.

### Fonctions et pointeurs

- Le programme doit être découpé en fonctions.
- La fonction main doit quasiment uniquement appeler des fonctions.
- Vous devez utiliser des pointeurs pour manipuler les tableaux et passer les structures aux fonctions.

## Etapes que vous pouvez suivre

Étape 1: lecture des données

- demander `nb_rangees` et `nb_tables_par_rangee`
- demander le nom du fichier
- lire la liste des élèves dans un tableau de `Eleve`

Étape 2: création de la salle

- générer un tableau de `Place` correspondant à toutes les places disponibles
- initialiser chaque place comme libre

Étape 3: placement aléatoire avec contraintes

- placer chaque élève sur une place libre
- essayer de:
  - répartir le nombre d'élèves sur les rangées
  - laisser un maximum d'espaces (par exemple: ne pas utiliser deux tables consécutives si possible)

Étape 4: affichage

- afficher le plan final, rangée par rangée
- indiquer clairement les places vides

## Industrialisation

### Etape 1 – Fichier en sortie :

Maintenant que votre code fonctionne et produit ce qui est attendu, il va falloir le rendre exploitable par quelqu'un de non technique.

Faites en sorte que votre programme crée un fichier .txt contenant le plan de salle avec les étudiants positionnés (comme dans la sortie console), ainsi que la liste des étudiants avec leur numéro de rangée et de table (cf. fichier sortie.txt).

### Etape 2 – Interface minimaliste :

Vous allez maintenant essayer de créer une petite interface graphique afin que l'utilisateur n'ait pas à passer par des lignes de commande et le terminal.

Je vous fournis dans le fichier interface.c le code minimal permettant de créer une fenêtre vide.

la ligne de commande à utiliser est :

```
gcc interface.c -o interface.exe -mwindows
```

À partir de maintenant, vous pouvez utiliser toutes les ressources que vous souhaitez pour y ajouter :

- un bouton permettant de sélectionner le fichier liste.txt,
- deux champs pour saisir les valeurs « nombre de rangées » et « nombre de tables par rangée »,
- ainsi qu'un bouton « Générer le plan d'examen ».

Il faudra ensuite connecter votre code à ces boutons.

Dans cette partie vous pouvez demander de l'aide aux IA, mais ne lui faites pas faire à votre place...