

# Projets

## 1 Instructions générales

Ces projets se font en binôme. Chaque binôme doit choisir un projet parmi les projets 1 et 2 puis y résoudre le problème proposé. Le projet 3 est obligatoire. Il y a donc deux projets en tout à réaliser : (1 ou 2) et 3.

Le tout devra être rendu dans un dossier au format Nom1\_Nom2 et les fichiers seront nommés selon le numéro du projet, par exemple, si vous avez choisi le projet 1, votre fichier se nommera projet1.py. Tout manquement à ces consignes sera sanctionné par un retrait de points.

Les programmes rendus doivent être personnels et de votre niveau, tout programme trop proche de celui d'un autre élève, d'un autre trouvé sur internet ou d'un niveau supérieur à celui de première se verra attribué automatiquement la note 0 plus un rapport de fraude.

Les critères d'évaluations sont les suivants :

- la réalisation de l'objectif du projet (et donc le rendu d'un programme fonctionnel) ;
- l'efficacité du programme ;
- la clarté et la structure du programme, l'usage de commentaires ;
- l'usage adéquat de fonctions, boucles, tests, etc ;
- le nommage des variables.

## 2 Projet 1 : BUSE

C'est la fin de la cinquième année pour Harry, Hermione, Ron, Neville et Malefoy qui doivent donc passer leur Brevet Universel de Sorcellerie Élémentaire (BUSE). Ils doivent tous passer cinq épreuves notées entre 0 et 20 pour les besoins de cet exercice.

1. Créer un tableau dont chaque ligne sera au format

```
['nom du personnage',note 1,...,note 5]
```

Tous les personnages susmentionnés devront apparaître dans le tableau mais vous pouvez en ajouter d'autres si vous voulez. Les notes des personnages devront être prises aléatoirement entre 0 et 20 sauf pour Hermione dont elles seront entre 18 et 20.

2. Ajouter trois colonnes au tableau. Une pour la moyenne, une pour le minimum et une dernière pour le maximum de chaque élève.
3. Ajouter une ligne au tableau. Elle contiendra les moyennes des notes à chaque épreuve, la moyenne générale, la note minimale et la note maximale de toutes les épreuves.
4. Un élève obtient une BUSE dans une matière si sa note à l'examen de celle-ci est supérieure ou égale à 10. Faire en sorte que votre programme affiche pour chaque élève une phrase de la forme

```
nomÉlève a eu nombreBUSE ; moyenne : valeurMoyenne ; note minimale :  
noteMin ; note maximale : noteMax
```

### 3 Projet 2 : le jeu de la vie

Le jeu de la vie est un automate cellulaire créé par John Conway en 1970. On pourra regarder la vidéo de Sciences Étonnantes pour une présentation plus complète. Il s'agit d'une grille (ou tableau) sur laquelle évoluent des cellules; une case du tableau contient un 1 si une cellule est présente et un 0 s'il n'y a pas de cellule.

Le jeu se décompose en une succession de tours et la vie des cellules obéit à deux règles à chacun d'entre eux :

**survie** : si une cellule à 2 ou 3 cellules voisines, elle survit au tour prochain, sinon elle meurt ;

**naissance** : si une case vide a exactement 3 cellules voisines, une cellule naît sur celle-ci au prochain tour.

**Exemples** : on se concentre dans chacun des exemples suivants sur la case centrale.

#### Survie

Tour actuel :

1	1	0
0	1	0
1	0	0

Tour suivant :

1	1	0
0	1	0
1	0	0

#### Mort par surpopulation

Tour actuel :

1	1	0
0	1	1
1	0	0

Tour suivant :

1	1	0
0	0	1
1	0	0

#### Mort par sous-population

Tour actuel :

0	0	0
0	1	0
1	0	0

Tour suivant :

0	0	0
0	0	0
1	0	0

#### Naissance

Tour actuel :

0	0	1
0	0	1
0	1	0

Tour suivant :

0	0	1
0	1	1
0	1	0

Votre mission, si vous l'acceptez (en fait vous n'avez pas le choix), est de programmer le jeu de la vie. Vous êtes libre de choisir la taille de la grille et la configuration de départ des cellules (vous pouvez choisir une configuration aléatoire). On représentera le tableau à l'aide de cases blanches et noires dans matplotlib.

## 4 Projet 3 : le sac à dos

### 4.1 Algorithme glouton

Un algorithme **glouton** est souvent une solution intéressante pour un problème d'optimisation. Ces problèmes ont deux caractéristiques : une fonction que l'on doit maximiser ou minimiser et une série de contraintes auxquelles il faut satisfaire. Dans les algorithmes gloutons, on fait toujours le choix qui semble meilleur sur le moment. Autrement dit, on fait un choix optimal localement dans l'espoir que ce choix mènera à la solution optimale globalement. Les algorithmes gloutons n'aboutissent pas toujours à des solutions optimales, mais ils y arrivent dans de nombreux cas.

### 4.2 Le sac à dos

Catwoman est en train de dévaliser le manoir du richissime Bruce Wayne. Pour emporter son butin, elle ne dispose que d'un sac à dos dans lequel elle ne peut pas mettre une charge supérieure à 5kg ; elle serait trop ralentie après et les risques de tomber sur le Batman ne sont pas négligeables. Voici les objets qu'elle est susceptible de voler :

Objet	Poids (en kg)	Valeur
Glaive en argent et rubis	4	1 000 000
Masque de samouraï d'époque	0,5	3 000 000
Sculpture en bronze	2	250 000
Petit tableau	1	1 500 000
Parchemin antique	0,1	2 000 000

Catwoman souhaite évidemment maximiser son gain dans la limite des 5kg de son sac à dos. Elle doit donc réfléchir à la meilleure combinaison d'objets possibles.

Ce problème fait partie des 21 problèmes NP-complets identifiés par Richard Karp en 1972. Il s'agit de remplir son sac à dos le mieux possible mais il y a évidemment des contraintes à respecter :

1. Les objets que l'on souhaite placer dans le sac à dos possèdent chacun une valeur et un poids.
2. La charge du sac à dos ne peut pas dépasser un poids maximal.
3. La masse totale des objets est supérieure à celle autorisée dans le sac.
4. Quels objets faut-il mettre dans le sac à dos de manière à maximiser la valeur totale des objets sélectionnés sans dépasser la charge maximale du sac ?

Écrire un programme prenant en données une liste de nom d'objets et pour chacun d'entre eux un poids et une valeur. Il déterminera ensuite quels sont ceux Catwoman devra voler afin de maximiser son gain dans la limite des 5kg.