

## Mini projet 1 : jeux de 32 cartes

Récupérer le fichier **jeux\_de\_cartes.py** donné en ressources.

### Conventions :

- La classe **Carte** aura deux attributs privés de type **int** nommés **valeur** et **couleur** .
  - Correspondance entre les valeurs de l'attribut valeur et les valeurs des cartes :

Valeur de l'attribut valeur	Valeur de la carte
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	valet
12	dame
13	roi
14	as

- Correspondance entre les valeurs de l'attribut valeur et les valeurs des cartes :

Valeur de l'attribut couleur	Couleur de la carte
1	pique
2	coeur
3	carreau
4	trèfle

- La comparaison de deux cartes ne se fait que sur la valeur, on ne tient pas compte de la couleur. Ainsi le valet de trèfle et le valet de coeur ont la même valeur, ses deux cartes seront considérées comme égales avec l'opérateur surchargé `==` .

## Les différentes étapes du mini-projet 1

**Etape 1 :** Compléter tous les **pass** dans le code de la classe **Carte**.

Afin de vérifier en partie que le code semble bon, vous pouvez lancer la fonction **verification1()**.  
L'affichage attendu est suivant :

```
as de pique
8 de trèfle
valet de coeur
valet de carreau
True
False
True
False
True
```

**Etape 2 :** Compléter tous les **pass** dans le code de la classe **Jeux**.

Afin de vérifier en partie que le code semble bon, vous pouvez lancer la fonction **verification()**.  
L'affichage attendu est la liste en colonne de toutes les cartes du jeu de **max** dans un ordre aléatoire, suivi de toutes les du jeu de **bob** dans un autre ordre aléatoire.  
Attention aux effets de bord.

**Etape 3 :** Rédiger le code de la fonction bataille. La règle du jeu ici est la suivante. Chaque joueur joue ses cartes dans l'ordre de sa main. Celui qui a la valeur la plus élevée sans tenir compte de la couleur marque 1 point. Si les valeurs sont égales, personne ne marque de points.

Afin de vérifier votre code, lancer la fonction bataille(). Un affichage possible est suivant :

```
Score de max : 22
Score de bob : 26
Le vainqueur est bob
```

En cas de match nul, afficher 'Match nul' en troisième ligne.

**\* Etape 4 (bonus) :** Modifier le code de la fonction **bataille()** pour l'adapter aux vraies règles de la bataille.

**\* Etape 5 (bonus) :** Ajouter une animation graphique avec **tkinter** afin de visualiser l'évolution de la partie à l'écran.