

2024 sujet 14

Exercice 1

```
def min_et_max(tab):
    dico = {"min": tab[0], "max": tab[0]}
    for e in tab:
        if e < dico["min"]:
            dico["min"] = e
        if e > dico["max"]:
            dico["max"] = e
    return dico
```

Exercice 2

```
class Carte:
    def __init__(self, c, v):
        """Initialise les attributs couleur (entre 1 et 4),
        et valeur (entre 1 et 13). """
        self.couleur = c
        self.valeur = v

    def recuperer_valeur(self):
        """ Renvoie la valeur de la carte :
        As, 2, ..., 10, Valet, Dame, Roi """
        valeurs = ['As', '2', '3', '4', '5', '6', '7', '8',
                   '9', '10', 'Valet', 'Dame', 'Roi']
        return valeurs[self.valeur - 1]

    def recuperer_couleur(self):
        """ Renvoie la couleur de la carte
        (parmi pique, coeur, carreau, trèfle). """
        couleurs = ['pique', 'coeur', 'carreau', 'trèfle']
        return couleurs[self.couleur - 1]

class Paquet_de_cartes:
    def __init__(self):
        """ Initialise l'attribut contenu avec une liste des 52
        objets Carte possibles rangés par valeurs croissantes en
        commençant par pique, puis cœur, carreau et trèfle. """
        # Les exemples montrent que les cartes sont rangées par couleur
        # *puis* par valeurs croissantes. La formulation n'est pas claire
```

```

self.contenu = []
for couleur in range(1, 5):
    for valeur in range(1, 14):
        self.contenu.append(Carte(couleur, valeur))

def recuperer_carte(self, pos):
    """ Renvoie la carte qui se trouve à la position pos
    (entier compris entre 0 et 51). """
    assert pos >= 0 and pos <= 51, "paramètre pos invalide"
    return self.contenu[pos]

```