

```
# Jeu de cartes appelé "Pouilleux"

# L'ordinateur est le donneur des cartes.

# Une carte est une chaine de 2 caractères.
# Le premier caractère représente une valeur et le deuxième une couleur.
# Les valeurs sont des caractères comme '2','3','4','5','6','7','8','9','10','J','Q','K', et 'A'.
# Les couleurs sont des caractères comme : ♠, ♥, ♣, et ♦.
# On utilise 4 symboles Unicode pour représenter les 4 couleurs: pique, coeur, trèfle et carreau.
# Pour les cartes de 10 on utilise 3 caractères, parce que la valeur '10' utilise deux caractères.
```

```
import random
```

```
def attend_le_joueur():
    """()->None
    Pause le programme jusqu'au l'usager appui Enter
    """
    try:
        input("Appuyez Enter pour continuer. ")
    except SyntaxError:
        pass
```

```
def prepare_paquet():
    """()->list of str
    Retourne une liste des chaines de caractères qui représente tous les cartes,
    sauf le valet noir.
    """
    paquet=[]
    couleurs = ['\u2660', '\u2661', '\u2662', '\u2663']
    valeurs = ['2','3','4','5','6','7','8','9','10','J','Q','K','A']
    for val in valeurs:
        for couleur in couleurs:
            paquet.append(val+couleur)
    paquet.remove('J\u2663') # élimine le valet noir (le valet de trèfle)
    return paquet
```

```
def melange_paquet(p):
    """(list of str)->None
    Mélange la liste des chaines des caractères qui représente le paquet des cartes
    """
    random.shuffle(p)
```

```
def donne_cartes(p):
```

```
"""(list of str)-> tuple of (list of str,list of str)
```

Retourne deux listes qui représentent les deux mains des cartes.

Le donneur donne une carte à l'autre joueur, une à lui-même, et ça continue jusqu'à la fin du paquet p.

```
"""
```

```
donneur=[]
autre=[]
while len(p)>1:
    autre.append(p.pop())
    donneur.append(p.pop())
autre.append(p.pop()) # donne la dernière carte
return (donneur, autre)
```

```
def elimine_paires(l):
```

```
"""
```

```
(list of str)->list of str
```

Retourne une copie de la liste l avec tous les paires éliminées et mélange les éléments qui restent.

Test:

(Notez que l'ordre des éléments dans le résultat pourrait être différent)

```
>>> elimine_paires(['9♠', '5♠', 'K♦', 'A♣', 'K♣', 'K♥', '2♠', 'Q♠', 'K♠', 'Q♦', 'J♠', 'A♥',
'4♣', '5♣', '7♥', 'A♠', '10♣', 'Q♥', '8♥', '9♦', '10♦', 'J♥', '10♥', 'J♣', '3♥'])
['10♣', '2♠', '3♥', '4♣', '7♥', '8♥', 'A♣', 'J♣', 'Q♦']
>>> elimine_paires(['10♣', '2♣', '5♦', '6♣', '9♣', 'A♦', '10♦'])
['2♣', '5♦', '6♣', '9♣', 'A♦']
"""
```

```
resultat=[]
```

```
l.sort()
```

```
i=0
```

```
while i<len(l)-1:
```

```
    carte1=l[i]
```

```
    carte2=l[i+1]
```

```
    if carte1[0]==carte2[0] and carte1[1]==carte2[1]: # compare les premiers 2 caracteres si
c'est un 10
```

```
        i=i+1 # avance a la carte prochaine
```

```

elif carte1[0]==carte2[0]: # compare le premier caractere (c'est ne pas un 10)
    i=i+1 # avance a la carte prochaine
else:
    resultat.append(l[i])
    i=i+1
if i==len(l)-1: # True si la derniere carte n'a pas une paire
    resultat.append(l[i])

random.shuffle(resultat)
return resultat

def affiche_cartes(p):
    """
    (list)-None
    Affiche les éléments de la liste p séparées par d'espaces
    """

    for val in p:
        print(val, end=' ')
    print()

def entrez_position_valide(n):
    """
    (int)->int
    Retourne un entier du clavier, de 1 à n (1 et n inclus).
    Continue a demander si l'utilisateur entre un entier qui n'est pas dans l'interval [1,n]

    Precondition: n>=1
    """
    print("J'ai", n, "cartes. Si 1 est la position de ma première carte et")
    print(n, "est la position de ma dernière carte, laquelle de mes cartes vous voulez?")
    position=int(input("SVP entrer un entier de 1 à "+str(n)+" : ").strip())

    while not(position>=1 and position <=n):
        position=int(input("Position invalide. SVP enter un entier de 1 à "+str(n)+" : ").strip())
    return position

def joue():
    """()->None
    Cette fonction joue le jeu"""

    p=prepare_paquet()
    melange_paquet(p)

```

```

tmp=donne_cartes(p)
donneur=tmp[0]
humain=tmp[1]

print("Bonjour. Je m'appelle Robot et je distribue les cartes.")
print("Votre main est:")
affiche_cartes(humain)
print("Ne vous inquiétez pas, je ne peux pas voir vos cartes ni leur ordre.")
print("Maintenant défaussez toutes les paires de votre main. Je vais le faire moi aussi.")
attend_le_joueur()

donneur=elimine_paires(donneur)
humain=elimine_paires(humain)

ordinales=["ère", "ème"] # terminaisons pour les numérales ordinales en français

tour=0

while len(donneur)>0 and len(humain)>0:
    if tour==0: # donneur offres ses cartes
        print("*****")
        print("Votre tour.")
        print("Votre main est:")
        affiche_cartes(humain)

        position=entrez_position_valide(len(donneur))
        elem=donneur[position-1]
        donneur.remove(elem) # ca marche parce que chaque carte est unique

        if position>1:
            ord_index=1
        else:
            ord_index=position-1

        print("Vous avez demande ma "+str(position)+ordinales[ord_index]+" carte.")
        print("La voila. C'est un", elem)

        humain.append(elem)
        print("Avec", elem, "ajouté, votre main est:")
        affiche_cartes(humain)

        print("Après défaussé toutes les paires et mélanger les cartes, votre main est:")
        humain=elimine_paires(humain)
        affiche_cartes(humain)

```

```

attend_le_joueur()
tour=1

else:
    print("*****")
    print("Mon tour.")
    index=random.randint(0,len(humain)-1)
    elem=humain[index]
    humain.remove(elem)
    donneur.append(elem)
    donneur=elimine_paires(donneur)

    if index>1:
        ord_index=1
    else:
        ord_index=index

    print("J'ai pris votre "+str(index+1)+ordinales[ord_index]+" carte.")
    attend_le_joueur()
    tour=0

if len(donneur)==0:
    print("J'ai terminé toutes les cartes.")
    print("Vous avez perdu! Moi, Robot, j'ai gagné.")
else:
    print("*****")
    print("J'ai terminé toutes les cartes.")
    print("Félicitations! Vous, Humain, vous avez gagné.")

# programme principale
joue()

```