

Centres étrangers – 2023 – sujet1 - Correction

Exercice 3 (6 points)

1. Fonction `ajout(f)`

On doit :

- tirer un indice aléatoire entre 0 et 3
- ajouter la couleur correspondante à la file

Correction:

```
def ajout(f):  
    couleurs = ("bleu", "rouge", "jaune", "vert")  
    indice = randint(0, 3)  
    enfiler(f, couleurs[indice])  
    return f
```

👉 *Commentaire :*

- *randint(0,3) car le tuple contient 4 éléments indexés de 0 à 3.*
 - *on ajoute en queue car la file est FIFO.*
-

2. Fonction `vider(f)`

On doit vider la file **sans la renvoyer**.

Correction:

```
def vider(f):  
    while not est_vide(f):  
        defiler(f)
```

👉 *Commentaire :*

- *on retire tous les éléments un par un.*
- *on ne retourne rien car la consigne précise « sans la renvoyer ».*

3. Fonction `affich_seq(sequence)`

Objectif :

- ajouter une couleur
- afficher chaque couleur
- ne pas vider la file à la fin

Il faut donc utiliser une file temporaire `stock`.

Correction:

```
def affich_seq(sequence):
    stock = creer_file_vider()
    ajout(sequence)

    while not est_vide(sequence):
        c = defiler(sequence)
        affichage(c)
        time.sleep(0.5)
        enfiler(stock, c)

    while not est_vide(stock):
        enfiler(sequence, defiler(stock))
```

👉 *Commentaire :*

- on vide temporairement `sequence` pour l'afficher.
- `stock` permet de restaurer la file dans le bon ordre.
- sans cette restauration, la séquence serait perdue.

4. Fonction `tour_de_jeu(sequence)`

On doit :

- ajouter et afficher la séquence
 - comparer les saisies joueur
 - arrêter si erreur
 - conserver la file en cas de réussite
-

4.a Complétion logique du script

Les bonnes correspondances sont :

- **ZONE A** → `affich_seq(sequence)`
 - **ZONE B** → `while not est_vide(sequence) :`
 - **ZONE C** → `defiler(sequence)`
 - **ZONE D** → `enfiler(stock, c_seq)`
 - **ZONE E** → `vider(sequence)`
 - **ZONE F** → `while not est_vide(stock) :`
-

Version cohérente complete:

```
def tour_de_jeu(sequence):
    stock = creer_file_vide()

    affich_seq(sequence)

    while not est_vide(sequence):
        c_seq = defiler(sequence)
        enfiler(stock, c_seq)

        c_joueur = saisie_joueur()

        if c_joueur != c_seq:
            vider(sequence)
            return False

    while not est_vide(stock):
        enfiler(sequence, defiler(stock))

    return True
```

👉 *Commentaire :*

- *on compare au fur et à mesure.*
 - *en cas d'erreur, la séquence est vidée immédiatement.*
 - *en cas de réussite, la séquence est restaurée.*
-

4.b Modification pour répétition automatique

On peut entourer la logique d'une boucle :

```
def jeu():  
    sequence = creer_file_vide()  
  
    while True:  
        if not tour_de_jeu(sequence):  
            vider(sequence)
```

👉 *Commentaire :*

- si le joueur gagne, une nouvelle couleur est ajoutée au tour suivant.

- si le joueur perd, on vide puis on recommence.
