

Corrigé sujet **13** - Année : 2022

Sujet 13 - 20222



1. Exercice 1

```
1 def rendu(somme_a_rendre):
2     n1 = somme_a_rendre//5
3     somme_a_rendre = somme_a_rendre%5
4     n2 = somme_a_rendre//2
5     n3 = somme_a_rendre%2
6     return [n1,n2,n3]
```

Commentaire

- Bien qu'au programme de première ([algorithme glouton](#)), cet exercice est bien plus difficile que ceux proposés habituellement en exercice 1 (recherche simple, recherche de maximum, calcul de moyennes, ...).
- La correction proposée ici utilise les trois variables `n1`, `n2` et `n3` proposées dans l'énoncé. Dans un cadre plus général, une boucle travaillant sur une liste de pièces serait préférable. Voir par exemple la [correction proposée ici](#).

2. Exercice 2

```

1  class Maillon :
2      def __init__(self,v) :
3          self.valeur = v
4          self.suivant = None
5
6  class File :
7      def __init__(self) :
8          self.dernier_file = None
9
10     def enfile(self,element) :
11         nouveau_maillon = Maillon(element) #(1)
12         nouveau_maillon.suivant = self.dernier_file
13         self.dernier_file = nouveau_maillon #(2)
14
15     def est_vide(self) :
16         return self.dernier_file == None
17
18     def affiche(self) :
19         maillon = self.dernier_file
20         while maillon != None : #(3)
21             print(maillon.valeur)
22             maillon = maillon.suivant #(4)
23
24     def defile(self) :
25         if not self.est_vide() :
26             if self.dernier_file.suivant == None :
27                 resultat = self.dernier_file.valeur
28                 self.dernier_file = None
29                 return resultat
30             maillon = self.dernier_file #(5)
31             while maillon.suivant.suivant != None :
32                 maillon = maillon.suivant
33             resultat = maillon.suivant.valeur
34             maillon.suivant = None
35             return resultat
36         return None

```

1. Le constructeur de la classe `Maillon` prend en argument la valeur `v` (appelée ici `element`)
2. Comme indiqué dans l'énoncé, l'attribut `dernier_file` doit contenir le dernier maillon enfilé.
3. La fin de file est atteinte lorsque le maillon vaut `None`
4. Passage au maillon suivant
5. On commence au dernier maillon en s'assurant avant que le maillon suivant n'est pas `None`

Attention

1. L'implémentation d'une file proposée ici est problématique, pour défiler, il faut partir de la fin (seul accès à la file), la remonter en entier afin de défiler le premier.
2. Faire un schéma de l'implémentation proposée peut aider à la compréhension du code donné dans l'énoncé.