



C5 Listes, Piles et Files

Exercice 1

Énoncé

Exercice 5 du sujet [Centres Étrangers 1 - 2021](#)

Corr. Q1



Corr. Q2



Corr. Q3

Script Python

```

1  def maximum(P):
2      if est_vide(P):
3          return None
4      m = depile(P)
5      while not est_vide(P):
6          val = depile(P)
7          if val > m:
8              m = val
9      return m

```

Avec le code ci-dessus, la pile `p` est vide à la fin de l'exécution. Pour éviter cela, on peut par exemple créer une pile `q` temporaire qui recevra les éléments de `p`, avant de retransférer à la fin du programme les éléments de `q` dans `p`.

Script Python

```

1  def maximum(P):
2      Q = creer_pile()
3      if est_vide(P):
4          return None
5      m = depile(P)
6      empile(Q, m)
7      while not est_vide(P):
8          val = depile(P)
9          empile(Q, val)
10         if val > m:
11             m = val
12     while not est_vide(Q):

```

```
13     empile(P, depile(Q))  
14     return m
```

Corr. Q4

Q4a. On va vider la pile `p` dans une pile `q` tout en comptant le nombre d'éléments dépliés dans une variable `t`. On redonne ensuite à `p` son état initial en vidant `q` dans `p`.

Q4b

Script Python

```
1 def taille(P):  
2     if est_vide(P):  
3         return 0  
4     Q = creer_pile()  
5     t = 0  
6     while not est_vide(P):  
7         empile(Q, depile(P))  
8         t += 1  
9     while not est_vide(Q):  
10        empile(P, depile(Q))  
11    return t
```

 Exercice 2

Énoncé

Exercice 1 du sujet [La Réunion J2 - 2022](#)

Corr. Q1



Corr. Q2a

La variable `temp` contient la valeur 25.

Corr. Q2b

`p1` est identique, elle contient toujours les valeurs 25, 3 et 7.

Corr. Q3

Script Python

```

1 def addition(p):
2     nb1 = depiler(p)
3     nb2 = depiler(p)
4     empiler(p, nb1 + nb2)

```

Corr. Q4

Script Python

```

1 p = pile_vide()
2 empiler(p, 3)
3 empiler(p, 5)
4 addition(p)
5 empiler(p, 7)
6 multiplication(p)

```

Exercice 3

Énoncé

Exercice 2 du sujet [Métropole Candidats Libres J1 - 2021](#)

Corr. Q1a

Script Python

```

1 pile1 = Pile()
2 pile1.empiler(7)
3 pile1.empiler(5)
4 pile1.empiler(2)

```

Corr. Q1b

L'affichage produit est 7, 5, 5, 2.

Corr. Q2a

- Cas n°1 : 3, 2
- Cas n°2 : 3, 2, 5, 7
- Cas n°3 : 3
- Cas n°4 : «pile vide»

Corr. Q2b

La fonction `mystere` permet d'obtenir la pile retournée jusqu'à un élément particulier (s'il existe).

Corr. Q3

Script Python

```

1 def etendre(pile1, pile2):
2     while not pile2.est_vide():
3         val = pile2.depiler()
4         pile1.empiler(val)

```

Corr. Q4

Script Python

```

1 def supprime_toutes_occurrences(pile, element):
2     p_temp = Pile()

```

```
3     while not pile.est_vide():
4         val = pile.depiler()
5         if val != element:
6             p_temp.empiler(val)
7         while not p_temp.est_vide():
8             val = p_temp.depiler()
9             pile.empiler(val)
```

 Exercice 4

Énoncé

Exercice 5 du sujet Amérique du Nord J1 - 2021

Corr. Q1a

Le contenu de la pile P sera

 Script Python

```
| "rouge" |
| "vert"  |
| "jaune" |
| "rouge" |
| "jaune" |
```

Corr. Q1b

 Script Python

```
1 def taille_file(F):
2     """File -> Int"""
3     F_temp = creer_file_vide()
4     n = 0
5     while not est_vide(F):
6         enfiler(F_temp, defiler(F))
7         n += 1
8     while not est_vide(F_temp):
9         enfiler(F, defiler(F_temp))
10    return n
```

Corr. Q2

 Script Python

```
1 def former_pile(F):
2     """File -> Pile"""
3     P_temp = creer_pile_vide()
4     P = creer_pile_vide()
5     while not est_vide(F):
6         empiler(P_temp, defiler(F))
7     while not est_vide(P_temp):
8         empiler(P, depiler(P_temp))
9     return P
```

Corr. Q3

 Script Python

```

1 def nb_elements(F, elt):
2     """File, Int -> Int"""
3     F_temp = creer_file_vide()
4     n = 0
5     while not est_vide(F):
6         val = defiler(F)
7         if val == elt:
8             n += 1
9         enfiler(F_temp, val)
10    while not est_vide(F_temp):
11        enfiler(F, deFiler(F_temp))
12    return n

```

Corr. Q4

 Script Python

```

1 def verifier_contenu(F, nb_rouge, nb_vert, nb_jaune):
2     """File, Int, Int, Int -> Bool"""
3     return nb_elements(F, "rouge") <= nb_rouge and \
4            nb_elements(F, "vert") <= nb_vert and \
5            nb_elements(F, "jaune") <= nb_jaune

```

 Exercice 5

Énoncé

Exercice 2 du sujet Centres Étrangers J1 - 2022

Corr. Q1

Il faut écrire l'instruction :

 Script Python

```
panier_1.enfile((31002, "café noir", 1.50, 50525))
```

Corr. Q2

 Script Python

```
1 def remplir(self, panier_temp):
2     while not panier_temp.est_vide():
3         article = panier_temp.defile()
4         self.enfile(article)
```

Corr. Q3

 Script Python

```
1 def prix_total(self):
2     total = 0
3     panier_temp = Panier()
4     while not self.est_vide():
5         article = self.defile()
6         total += article[2]
7         panier_temp.enfile(article)
8     self.remplir(panier_temp)
9     return total
```

Corr. Q4

 Script Python

```
1 def duree_passage_en_caisse(self):
2     if self.est_vide():
3         return None
4     horaire_premier = self.defile()[3]
5     while not self.est_vide():
6         horaire_dernier = self.defile()[3]
7     return horaire_dernier - horaire_premier
```

0.1. Exercice 6

Exercice 6

énoncé

Exercice 3 du sujet [Centres Etrangers J1 - 2023](#)

Jeu du Simon

Correction Q1.

Script Python

```

1 def ajout(f):
2     couleurs = ("bleu", "rouge", "jaune", "vert")
3     indice = randint(0, 3)
4     enfiler(f, couleur[indice])
5     return f

```

Correction Q2.

Script Python

```

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

```

Correction Q3.

Script Python

```

1 def affich_seq(sequence):
2     stock = creer_file_vide()
3     ajout(sequence)
4     while not est_vide(sequence):
5         c = defiler(sequence)
6         affichage(c)
7         time.sleep(0.5)
8         enfiler(stock, c)
9     while not est_vide(stock):
10        enfiler(sequence, defiler(stock))

```

Correction Q4.a.

Script Python

```

1 def tour_de_jeu(sequence):
2     affich_seq(sequence)
3     stock = creer_file_vide()
4     while not est_vide(sequence):
5         c_joueur = saisie_joueur()
6         c_seq = defiler(sequence)

```

```

7     if c_joueur == c_seq:
8         enfiler(stock, c_seq)
9     else:
10        vider(sequence)
11    while not est_vide(stock):
12        enfiler(sequence, defiler(stock))

```

Correction Q4.b.

Question bizarre...

 Script Python

```

1 def tour_de_jeu_modifie(sequence):
2     while True:
3         affich_seq(sequence)
4         stock = creer_file_vide()
5         while not est_vide(sequence):
6             c_joueur = saisie_joueur()
7             c_seq = defiler(sequence)
8             if c_joueur == c_seq:
9                 enfiler(stock, c_seq)
10            else:
11                vider(sequence)
12                vider(stock)
13            while not est_vide(stock):
14                enfiler(sequence, defiler(stock))

```

ou bien

 Script Python

```

1 def tour_de_jeu_modifie(sequence):
2     affich_seq(sequence)
3     stock = creer_file_vide()
4     while not est_vide(sequence):
5         c_joueur = saisie_joueur()
6         c_seq = defiler(sequence)
7         if c_joueur == c_seq:
8             enfiler(stock, c_seq)
9         else:
10            vider(sequence)
11            print("Perdu ! On rejoue !")
12            tour_de_jeu_modifie(sequence)
13        while not est_vide(stock):
14            enfiler(sequence, defiler(stock))
15        tour_de_jeu_modifie(sequence)

```