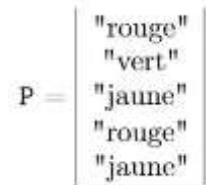


Correction

NSI - 2021 Amérique (21-NSIJ1AN1)

Exercice 5 - Programmation, Piles, Files

1. a. La file F sera vide tandis que la pile P contiendra : "jaune", "rouge", "jaune", "vert" et "rouge" (du fond vers le sommet de la pile).



1. b.

Solution : Programmation Procédurale du TAD

```
def taille_file(F):
    """ File -> Int """
    total = 0
    F_2 = creer_file_vide()
    while not est_vide(F):
        total += 1
        enfiler(F_2, defiler(F))
    F = F_2 # restore la file
    return total
```

Solution : Programmation Orientée Objet du TAD

```
def taille_file(F):
    """ File -> Int """
    total = 0
    F_2 = creer_file_vide()
    while not F_2.est_vide():
        total += 1
        F_2.enfiler(F.defiler())
    F = F_2 # restore la file
    return total
```

2.

Solution : Programmation Procédurale du TAD

```
def former_pile(F):
    """ File -> Pile """
    P = creer_pile_vide()
    while not est_vide(F):
        empiler(P, (defiler(F)))
    # Renversement de la pile
    P_2 = creer_pile_vide()
    while not est_vide(P_2):
        empiler(P_2, depiler(P))
    return P_2
```

Solution : Programmation Orientée Objet du TAD

```
def former_pile(F):
    """ File -> Pile """
    P = creer_pile_vide()
    while not F.est_vide():
        P.empiler(F.defiler())
    # Renversement de la pile
    P_2 = creer_pile_vide()
    while not P_2.est_vide():
        P_2.empiler(P.depiler())
    return P_2
```

Correction

NSI - 2021 Amérique (21-NSIJ1AN1)

3.

Solution : Programmation Procédurale du TAD

```
def nb_elements(F, elt):
    """ File, element -> Int """
    total = 0
    F_2 = creer_file_vide()
    while not est_vide(F):
        temp = defiler(F):
        if elt == temp:
            total += 1
        enfiler(F_2, temp)
    F = F_2 # restore la file
    return total
```

Solution : Programmation Orientée Objet du TAD

```
def nb_elements(F, elt):
    """ File, element -> Int """
    total = 0
    F_2 = creer_file_vide()
    while not F.est_vide():
        temp = F.defiler():
        if elt == temp:
            total += 1
        F_2.enfiler(temp)
    F = F_2 # restore la file
    return total
```

4.

```
def verifier_contenu(F, nb_rouge, nb_vert, nb_jaune):
    """ File, Int, Int, Int -> Bool """
    r, v, b = nb_element(F, 'rouge'), nb_element(F, 'vert'), nb_element(F, 'jaune')
    return r <= nb_rouge and v <= nb_vert and j <= nb_jaune
```