

**Identifiant de l'étudiant****Nom:****Code permanent:****Consignes**

1. Toute remise en retard sera refusée et sera sanctionnée par la note zéro.
2. Vous avez le droit à une seule remise. Seule la première remise sera corrigée, toute autre remise complète ou partielle (une ou plusieurs feuilles) ne sera pas considérée pour la correction.
3. Vos réponses doivent être claires et lisibles. Une réponse qui n'est pas claire aura la note zéro.
4. La copie remise doit être agrafée sinon elle sera refusée.
5. Vous devez soit imprimer ce fichier et ensuite écrire vos réponses dans l'espace réservé, soit écrire directement vos réponses dans le document PDF et ensuite l'imprimer et le remettre.
6. La copie papier de votre travail doit être remis au département informatique en suivant la procédure indiquée sur la page du cours.
7. Vos réponses ne doivent pas dépasser l'espace qu'on vous a réservé.
8. Vous devez indiquer clairement sur la copie remise votre nom et votre code permanent.
9. Même si aucun point n'est attribué pour la qualité de la présentation du travail, ce critère sera considéré pour la correction et vous pouvez perdre jusqu'à 10% pour cela.
10. Les règlements sur le plagiat seront strictement appliqués.
11. Cochez la case suivante si vous avez lu et compris ces consignes :

Oui

## Question 1

Dans l'exemple d'exécution suivant, il y a des lignes qui correspondent à un affichage effectué par le programme et des lignes qui correspondent à des valeurs saisies par un utilisateur. Encerclez chaque ligne correspondant à une valeur saisie par l'utilisateur (veuillez répondre directement sur la copie).

```
1  Nombre de cartes pour la pile 1:
2  2
3  Nombre de cartes pour la pile 2:
4  2
5  Nombre de cartes pour la pile 3:
6  2
7  ** Etat des piles:
8  2:2:2
9  => Coup du joueur 1
10 Pile cible:
11 1
12 Nombre de cartes:
13 2
14 ** Etat des piles:
15 0:2:2
16 => Coup du joueur 2
17 Pile cible:
18 2
19 Nombre de cartes:
20 2
21 ** Etat des piles:
22 0:0:2
23 => Coup du joueur 1
24 Pile cible:
25 3
26 Nombre de cartes:
27 2
28 Gagnant: joueur 2
```

## Question 2

Complétez les lignes vides de l'exemple d'exécution suivant :

```
1      Nombre de cartes pour la pile 1:
2  À compléter →
3      Une pile doit avoir au moins 2 cartes.
4      Nombre de cartes pour la pile 1:
5  À compléter →
6      Nombre de cartes pour la pile 2:
7  À compléter →
8      Nombre de cartes pour la pile 3:
9  À compléter →
10     ** Etat des piles:
11     3:3:3
12     => Coup du joueur 1
13     Pile cible:
14  À compléter →
15     Numero de pile inexistant: choisir 1, 2 ou 3
16     Pile cible:
17  À compléter →
18     Nombre de cartes:
19  À compléter →
20     Vous devez retirer au moins une carte et au plus trois
21     Nombre de cartes:
22  À compléter →
23     ** Etat des piles:
24     0:3:3
25     => Coup du joueur 2
26     Pile cible:
27  À compléter →
28     Nombre de cartes:
29  À compléter →
30     ** Etat des piles:
31     0:0:3
32     => Coup du joueur 1
33     Pile cible:
34  À compléter →
35     Nombre de cartes:
36  À compléter →
37     Gagnant: joueur 2
```

## Question 3

Complétez les lignes vides de l'exemple d'exécution suivant :

1 Nombre de cartes pour la pile 1:  
2 5  
3 Nombre de cartes pour la pile 2:  
4 5  
5 Nombre de cartes pour la pile 3:  
6 5  
7 \*\* Etat des piles:  
8 5:5:5  
9 => Coup du joueur 1  
10 Pile cible:  
11 1  
12 Nombre de cartes:  
13 5

14 *À compléter →*  
15 *À compléter →*  
16 *À compléter →*  
17 *À compléter →*  
18 *À compléter →*  
19 *À compléter →*  
20 *À compléter →*  
21 *À compléter →*  
22 *À compléter →*  
23 *À compléter →*  
24 *À compléter →*  
25 *À compléter →*

26 => Coup du joueur 1  
27 Pile cible:  
28 2  
29 Nombre de cartes:  
30 3  
31 \*\* Etat des piles:  
32 0:2:5  
33 => Coup du joueur 2  
34 Pile cible:  
35 2  
36 Nombre de cartes:  
37 3

38 *À compléter →*  
39 *À compléter →*  
40 *À compléter →*  
41 *À compléter →*  
42 *À compléter →*  
43 *À compléter →*  
44 *À compléter →*

45 \*\* Etat des piles:  
46 0:0:2  
47 => Coup du joueur 2

48  
49  
50  
51  
52

Pile cible:

3

Nombre de cartes:

2

*À compléter →*

## Question 4

Identifiez les cas qui feront planter le programme (une phrase en une ligne pour chaque cas).

## Question 5

Identifiez les cas qui produiront des messages d'erreur (une phrase en une ligne pour chaque cas).

**Remarque importante :** En Pseudocode il n'y a pas de classe `DecimalFormat`, on affiche donc les nombres réels directement. De même, on n'a pas de classe `MessagesTp1` en Pseudocode. Notez aussi que vous ne devez pas déclarer les variables en Pseudocode, il suffit de les utiliser directement.

Vous devez respecter la syntaxe du Pseudocode vue dans le cours (notamment pour les symboles  $\leftarrow$ ,  $\neq$ ,  $\leq$ , etc.). Vos algorithmes doivent être correctement indentés.

## Question 6

Écrire en Pseudocode une boucle de validation pour le nombre initial de cartes dans la première pile. Utilisez la structure FAIRE-TANTQUE.

Plus spécifiquement, ce qu'on vous demande c'est d'écrire en Pseudocode la partie de votre programme qui sera chargée de demander à l'utilisateur de saisir le nombre de cartes pour la première pile et qui procèdera à la validation de ce nombre selon les spécifications de l'énoncé.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

## Question 7

Durant l'exécution de votre programme, on invite l'un des joueurs à saisir d'abord une pile cible ensuite le nombre de cartes à retirer depuis cette pile.

On vous demande d'écrire en Pseudocode la partie de votre programme qui sera chargée de demander au joueur de saisir le nombre de cartes à retirer et qui procèdera à sa validation. Dans cette première version, la validation se fera sans considération du nombre de cartes disponibles dans la pile cible (autrement dit, on n'affichera pas de message d'erreur dans le cas où la pile cible est vide). Utilisez la structure TANTQUE-FAIRE :

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

## Question 8

Cet exercice est supposé vous aider à coder une partie du programme demandé dans l'énoncé du TP. Il ne s'agit pas d'une question directement liée à votre programme mais il y a des similarités.

Dans cet exercice, on suppose qu'on a trois boîtes identifiées par les chiffres : 1, 3 et 7. Chaque boîte peut contenir un nombre indéterminé de *billes*.

On demande à l'utilisateur de saisir le nombre de billes que contient initialement chaque boîte (pas de validation, on accepte toutes les valeurs). Ensuite, on doit demander à l'utilisateur d'ouvrir une boîte non-vide. Le programme doit afficher un message d'erreur si l'identifiant saisi par l'utilisateur n'est pas valide et un message différent dans le cas où la boîte choisie est vide. Dans ces deux cas, le programme doit demander de saisir un nouvel identifiant de boîte et refaire la validation.

Si l'identifiant saisi est valide et que la boîte choisie n'est pas vide, le programme doit afficher le nombre de billes trouvées dans la boîte choisie avant de se terminer. Voici un exemple d'exécution :

```
Nombre de billes pour la boîte 1:
2
Nombre de billes pour la boîte 3:
0
Nombre de billes pour la boîte 7:
8
Choix de la boîte (1, 3 ou 7):
9
Boîte non trouvée.
Choix de la boîte (1, 3 ou 7):
3
Boîte vide. Choisir une boîte différente.
Choix de la boîte (1, 3 ou 7):
7
Nombre de billes trouvées:
9
```

Complétez le Pseudocode suivant pour répondre à cette question en respectant la contrainte suivante : « le nombre maximal d'instructions AFFICHER entre DEBUT et FIN est 8 ».

```
1  DEBUT
2      AFFICHER "Nombre de billes pour la boite 1:"
3      LIRE nbBillesBoite1
4      AFFICHER "Nombre de billes pour la boite 3:"
5      LIRE nbBillesBoite3
6      AFFICHER "Nombre de billes pour la boite 7:"
7      LIRE nbBillesBoite7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45      AFFICHER "Nombre de billes trouvées:"
46      AFFICHER nbBilles
47  FIN
```