

DDK1 PC2

COLLEGE Mgr François-Xavier VOGT		Année scolaire 2024-2025
Département d'Informatique		Date : Novembre 2024
MINI SESSION EPREUVE D'INFORMATIQUE Classe : PCD Durée : 02 heures		

Aucun document ou matériel en dehors de ceux remis aux candidats par l'examineur n'est autorisé.

I. ÉVALUATION DES RESSOURCES /13 PTS

Exercice 1/ 04 PTS : Questions du cours.

1. Définir les concepts suivants :
 - a. Procédure 0,5 pt
 - b. Structure de données 0,5 pt
2. Donner une différence pertinente entre une PILE et une FILE. 1 pt
3. Donner deux (02) éléments caractéristiques d'un tableau. 1 pt
4. Lister quatre (04) types de données élémentaires utilisés en algorithmique. 1 pt

Exercice 2/ 09 PTS : Ecriture et exécution des algorithmes

1. Donner les instructions algorithmiques qui correspondent aux expressions suivantes :
 - a. Déclarer la constante N de valeur 50 ; 0,5 pt
 - b. Récupérer une valeur au clavier et la sauvegarder dans 3^{ème} case du tableau NOTE ; 0,5 pt
2. Soit l'algorithme ci-dessous :

Algorithme ndam

Var i : entier ;

Trouvé : booléen ;

Tab : Tableau [N] de chaîne de caractères ;

MOT : chaîne de caractères ;

Debut

Ecrire ('entrez une donnée : ');

Lire (MOT);

i ← 1;

Trouvé ← FAUX;

Tant que ((i ≤ N) ET (Trouvé=FAUX)) faire

Si (Tab[i] = MOT) alors

Trouvé ← VRAI;

Finsi

i ← i+1;

FinTantque

Ecrire (Trouvé);

Fin

- a. Dire dans quels cas cet algorithme s'arrête. (Il y en a 2) 1 pt
- b. Donner les valeurs possibles de la variable Trouvé. 0,5 pt
- c. Récrire cet algorithme en utilisant la boucle Répéter jusqu'à à la place de Tant que... Faire 1,5 pt
- d. Exécuter cet algorithme pour le tableau suivant : Tab 2 pts

PA	PCD	PTI	PSES
----	-----	-----	------

et les MOT = "PCD" et MOT = "PCE".

3. Ecrire un algorithme qui demande à un utilisateur de saisir un nombre et ne s'arrête que si le nombre est > 0 . Ensuite, calcule et affiche la factorielle du nombre. Pour rappel, la factorielle d'un nombre N est $1 \times 2 \times 3 \times 4 \times \dots \times N$ **1,5 pt**
4. Ecrire un algorithme nommé **mon-tableau** qui déclare un tableau de 100 entiers. Recherche et affiche toutes les valeurs paires contenues dans ce tableau **1,5 pt**

II. ÉVALUATION DES COMPÉTENCES / 07 PTS

Votre professeur de Français voudrait automatiser le traitement des notes de ses élèves de la classe de PCD. Il souhaiterait produire les statistiques attendues par sa hiérarchie. Vous êtes contacté pour apporter une solution à ce problème. Votre camarade propose la structure de données dont la déclaration est la suivante : **var NOTES : tableau [50] de réel ;**

1. Ecrire **portion de code (ou la partie de l'algorithme)** qui enregistre les notes des élèves dans le tableau NOTES. C'est-à-dire demande à l'utilisateur de saisir les valeurs par numéro d'ordre. Les notes saisies sont comprises entre 0 et 20. **1 pt**
2. En supposant les notes déjà saisies.
 - 2.1. Ecrire **une portion de code** affiche les notes saisies. **1,5 pt**
 - 2.2. Ecrire un algorithme ou **une portion de code** qui calcule et affiche la somme des notes et la moyenne générale de la classe. **1,5 pt**
3. Ecrire la partie de l'algorithme permettant de compter et d'afficher le nombre de moyennes supérieures ou égales à 10 ainsi que le taux de réussite. **1,5 pt**
Indication : parcourir les notes du tableau, si une note est ≥ 10 , utiliser une variable qu'on incrémentera. Cette variable contiendra à la fin du parcours, le nombre de notes ≥ 10 .
1. Ecrire la partie de l'algorithme permettant de ramener à 10 toutes notes comprises entre 8 et 9.99. **1,5 pt**

CORRECTION DE L'ÉPREUVE D'INFORMATIQUE

Collège Mgr François-Xavier VOGT
Année scolaire 2024-2025 – Novembre 2024

Département d'Informatique

Durée : 02 heures

Aucun document ou matériel en dehors de ceux remis aux candidats par l'examineur n'est autorisé.

Table des matières

1	ÉVALUATION DES RESSOURCES (13 PTS)	1
1.1	Exercice 1 : Questions du cours (04 PTS)	1
1.2	Exercice 2 : Écriture et exécution des algorithmes (09 PTS)	2
2	ÉVALUATION DES COMPÉTENCES (07 PTS)	5

1 ÉVALUATION DES RESSOURCES (13 PTS)

1.1 Exercice 1 : Questions du cours (04 PTS)

Correction

1. Définir les concepts suivants : (1 pt)

- **a. Procédure** : C'est un sous-programme (ou sous-algorithme) qui regroupe un ensemble d'instructions permettant d'effectuer une tâche spécifique. Contrairement à une fonction, une procédure ne renvoie pas de valeur de retour.
- **b. Structure de données** : C'est une manière d'organiser, de stocker et de manipuler des données en mémoire afin de les utiliser efficacement (exemples : tableau, liste, pile, file, arbre).

2. Donner une différence pertinente entre une PILE et une FILE. (1 pt)

- **PILE (LIFO – *Last In First Out*)** : Le dernier élément entré dans la structure est le premier à en sortir.
- **FILE (FIFO – *First In First Out*)** : Le premier élément entré dans la structure est le premier à en sortir.

3. Donner deux éléments caractéristiques d'un tableau. (1 pt)

- C'est une structure de données **statique** (sa taille est fixe et définie lors de la déclaration).
- Tous les éléments d'un tableau sont obligatoirement de **même type** (tous des entiers, tous des réels, etc.).
- Ses éléments sont accessibles et modifiables à l'aide d'un **indice** (ou index).

(Deux réponses parmi les trois suffisent)

4. Lister quatre types de données élémentaires utilisés en algorithmique. (1 pt)

- Entier (**Integer**)
- Réel (**Real / Float**)
- Booléen (**Boolean**)
- Caractère (**Char**)
- Chaîne de caractères (**String**)

(Quatre réponses parmi les cinq suffisent)

1.2 Exercice 2 : Écriture et exécution des algorithmes (09 PTS)

Correction

1. Instructions algorithmiques : (1 pt)

— a. Déclarer la constante N de valeur 5 :

1: **Const** N = 5 ;

— b. Récupérer une valeur au clavier et la sauvegarder dans la 3^{ème} case du tableau NOTE :

1: **Lire**(NOTE[3]) ;

2. Analyse de l'algorithme « ndam » : (5,5 pts)

a. Dire dans quels cas cet algorithme s'arrête. (0,5 pt)

L'algorithme s'arrête dans deux cas :

1. Le mot recherché (MOT) a été trouvé dans le tableau (Trouvé devient VRAI).
2. On a parcouru tout le tableau sans trouver le mot (l'indice i devient strictement supérieur à N).

b. Donner les valeurs possibles de la variable Trouvé. (1,5 pt)

Puisque Trouvé est déclarée comme un **booléen**, ses valeurs possibles sont uniquement : **VRAI** ou **FAUX**.

c. Réécrire cet algorithme en utilisant la boucle « Répéter... jusqu'à ». (2 pts)

Algorithm 1 Algorithme ndam (version Répéter...jusqu'à)

```
1: Var
2:    $i$  : entier ;
3:    $N$  : entier ;
4:    $Trouvé$  : booléen ;
5:    $Tab$  : tableau[1.. $N$ ] de chaîne de caractères ;
6:    $MOT$  : chaîne de caractères ;
7:
8: Début
9:   Écrire("entrez une donnée :");
10:  Lire( $MOT$ );
11:   $i \leftarrow 1$  ;
12:   $Trouvé \leftarrow \text{FAUX}$  ;
13:
14:  Répéter
15:    Si ( $Tab[i] = MOT$ ) alors
16:       $Trouvé \leftarrow \text{VRAI}$  ;
17:    FinSi
18:     $i \leftarrow i + 1$  ;
19:  Jusqu'à ( $(i > N)$  OU ( $Trouvé = \text{VRAI}$ ));
20:
21:  Écrire( $Trouvé$ );
22: Fin
```

d. Exécuter cet algorithme pour le tableau : $Tab[1]$ ="PA", $Tab[2]$ ="PCD", $Tab[3]$ ="PTI", $Tab[4]$ ="PSES" (Donc $N = 4$). (1,5 pt)

— **Cas 1 : $MOT = \text{"PCD"}$**

- $i = 1$: $Tab[1]$ ="PA" $\neq$ "PCD" $\Rightarrow$ $Trouvé$ reste FAUX, i passe à 2.
- $i = 2$: $Tab[2]$ ="PCD" = "PCD" $\Rightarrow$ $Trouvé$ devient VRAI, i passe à 3.
- La condition d'arrêt ($Trouvé = \text{VRAI}$) est vérifiée, la boucle s'arrête.
- **Résultat affiché : VRAI**

— **Cas 2 : $MOT = \text{"PCE"}$**

- $i = 1, 2, 3, 4$: Aucune case ne contient "PCE". $Trouvé$ reste FAUX à chaque itération.
- $i = 5$: La condition d'arrêt ($i > N$) soit ($5 > 4$) est vérifiée, la boucle s'arrête.
- **Résultat affiché : FAUX**

3. Algorithme de la factorielle (1,5 pt)

Algorithm 2 Algorithme Factorielle

```
1: Algorithme Factorielle
2: Var
3:    $N, i, Fact$  : entier ;
4:
5: Début
6:   // Demander un nombre tant qu'il n'est pas strictement positif
7:   Répéter
8:     Écrire("Entrez un nombre : ");
9:     Lire( $N$ );
10:  Jusqu'à ( $N > 0$ );
11:
12:  // Calcul de la factorielle
13:   $Fact \leftarrow 1$ ;
14:  Pour  $i$  de 1 à  $N$  faire
15:     $Fact \leftarrow Fact \times i$ ;
16:  FinPour
17:
18:  // Affichage du résultat
19:  Écrire("La factorielle de ",  $N$ , " est : ",  $Fact$ );
20: Fin
```

4. Algorithme « mon-tableau » (1,5 pt)

Algorithm 3 Algorithme mon_tableau

```
1: Algorithme mon_tableau
2: Var
3:    $Tab$  : tableau[1..100] de entier ;
4:    $i$  : entier ;
5:
6: Début
7:   // (On suppose ici que le tableau a déjà été rempli avec des valeurs)
8:
9:   // Recherche et affichage des valeurs paires
10:  Pour  $i$  de 1 à 100 faire
11:    Si ( $Tab[i] \bmod 2 = 0$ ) alors // Si le reste de la division par 2 est 0
12:      Écrire( $Tab[i]$ );
13:    FinSi
14:  FinPour
15: Fin
```

2 ÉVALUATION DES COMPÉTENCES (07 PTS)

Contexte

Votre professeur de Français voudrait automatiser le traitement des notes de ses élèves de la classe de PCO. Il souhaiterait produire les statistiques attendues par sa hiérarchie.

Structure de données proposée : var NOTES: tableau[1..50] de réel;

Correction

1. Enregistrer les notes des élèves (comprises entre 0 et 20). (1 pt)

```
1: Var i : entier ;
2:
3: Pour i de 1 à 50 faire
4:   Répéter
5:     Écrire("Entrez la note de l'élève n°", i, " (entre 0 et 20) : ");
6:     Lire(NOTES[i]);
7:   Jusqu'à (NOTES[i] ≥ 0 ET NOTES[i] ≤ 20);
8: FinPour
```

2. En supposant les notes déjà saisies.

2.1. Afficher les notes saisies. (1,5 pt)

```
1: Var i : entier ;
2:
3: Pour i de 1 à 50 faire
4:   Écrire(NOTES[i]);
5: FinPour
```

2.2. Calculer et afficher la somme des notes et la moyenne générale. (1,5 pt)

```
1: Var i : entier ; Somme, Moyenne : réel ;
2:
3: Somme ← 0 ;
4: Pour i de 1 à 50 faire
5:   Somme ← Somme + NOTES[i] ;
6: FinPour
7:
8: Moyenne ← Somme / 50 ;
9:
10: Écrire("La somme des notes est : ", Somme) ;
11: Écrire("La moyenne générale de la classe est : ", Moyenne) ;
```

3. Compter et afficher le nombre de notes ≥ 10 ainsi que le taux de réussite. (1,5 pt)

```
1: Var  $i$ ,  $NbReussite$  : entier ;  $TauxReussite$  : réel ;
2:
3:  $NbReussite \leftarrow 0$  ;
4: Pour  $i$  de 1 à 50 faire
5:   Si ( $NOTES[i] \geq 10$ ) alors
6:      $NbReussite \leftarrow NbReussite + 1$  ;
7:   FinSi
8: FinPour
9:
10:  $TauxReussite \leftarrow (NbReussite \times 100) / 50$  ;
11:
12: Écrire("Nombre d'élèves ayant la moyenne ou plus : ",  $NbReussite$ ) ;
13: Écrire("Taux de réussite : ",  $TauxReussite$ , " %") ;
```

4. Ramener à 10 toutes notes comprises entre 8 et 9.99. (1,5 pt)

Note algorithmique : En informatique, il est plus rigoureux d'écrire < 10 plutôt que ≤ 9.99 pour les nombres réels, afin d'éviter les problèmes de précision flottante.

```
1: Var  $i$  : entier ;
2:
3: Pour  $i$  de 1 à 50 faire
4:   Si ( $NOTES[i] \geq 8$  ET  $NOTES[i] < 10$ ) alors
5:      $NOTES[i] \leftarrow 10$  ;
6:   FinSi
7: FinPour
```

— Fin de la correction —