



INSTITUT PRIVE POLYVALENT DE
BONAMOOUSSADI

BP : 8527 D1a – Tél : 233-47-00-75

Année scolaire : 2025/2026
Classe : Terminale C&D
Coef : 2
Durée : 2H

EVALUATION FIN TRIMESTRE 1

Prof.: Soreil TCHEPDA

EPREUVE D'INFORMATIQUE

Partie A : ALGORITHME ET LANGAGE C

(10 points)

Le service hospitalier d'un hôpital possède un registre des patients hospitalisés, identifié par un code unique pour chaque patient. Le personnel souhaite vérifier rapidement si un patient est actuellement hospitalisé à partir de son code.

Pour cela, un algorithme de recherche séquentielle est utilisé.

Algorithme fourni

1	Algorithme VerificationPatient	16	Trouvé ← FAUX
2	Variables	17	i ← 1
3	T : Tableau[1..n] de chaînes de caractères	18	TANT QUE (i ≤ N) ET (Trouvé = FAUX) FAIRE
4	N, i : Entier	19	SI (T[i] = X) ALORS
5	X : Chaîne de caractères	20	Trouvé ← VRAI
6	Trouvé : Booléen	21	SINON
7	DEBUT	22	i ← i + 1
8	Ecrire("Entrer le nombre de patients hospitalisés :")	23	FIN SI
9	Lire(N)	24	FIN TANT QUE
10	Pour i ← 1 à N	25	SI (Trouvé = VRAI) ALORS
11	Ecrire("Entrer le code du patient ", i, " :")	26	Afficher("Le patient est hospitalisé position ", i)
12	Lire(T[i])	27	SINON
13	FinPour	28	Afficher("Le patient n'est pas hospitalisé")
14	Ecrire("Entrer le code du patient à rechercher :")	29	FIN SI
15	Lire(X)	30	FIN

Travaux à réaliser :

I. Compréhension de l'algorithme

2,25pts

- Donner l'objectif principal de cet algorithme. 0,5pt
- Citer deux données d'entrée et une donnée de sortie de cet algorithme. 0,75pt
- Donner le rôle des variables i et trouvé. 1pt

II. Analyse de l'algorithme

1,75pt

- Identifier la méthode de recherche utilisée dans cet algorithme. 0,25pt
- Justifier pourquoi la boucle Tant que peut s'arrêter avant d'atteindre la fin du tableau. 0,5pt
- Dire ce qui se passe si le code recherché ne figure pas dans le tableau. 1pt

III. Exécution de l'algorithme.

2pts

On suppose que : N = 4, T = ["PA10", "PA15", "PA20", "PA25"], X = "PA20"

Compléter (sur la feuille de composition), le tableau d'exécution ci-dessous.

Etape	i	T[i]	T[i] = X ?	trouvé	Commentaire

Donner le message affiché.

0,5pt

IV. Traduction en langage C

3,5pts

7. Citer les bibliothèques C à importer pour utiliser les fonctions printf() et scanf(), et les chaînes de caractères. 1pt
8. Réécrire la déclaration des variables N, i et trouvé en langage C. 1 pt
9. Traduire les trois lignes suivantes en langage C : 1,5pt
 - ❖ Lire(X)
 - ❖ TANT QUE ($i \leq N$) ET (Trouvé = FAUX) FAIRE
 - ❖ Afficher("Le patient est hospitalisé position ", i)

Partie B : SYSTEMES D'INFORMATION

(10 points)

Exercice 1 :

(5 points)

Dans une entreprise, le service informatique gère le prêt de matériel (ordinateurs portables, imprimantes, vidéoprojecteurs, ...) aux employés. Les règles de gestion dans cette entreprise sont les suivantes :

- Chaque **employé** est identifié par un **matricule**, possède un **nom**, un **prénom**, et un **service d'affectation**.
- Chaque **matériel** est identifié par un **codeMat**, possède une **désignation**, une **marque**, et un **état** (neuf, en panne, en prêt, ...)
- Chaque **emprunt** est identifié par un **numEmprunt**, et comprend une **dateEmprunt**, et une **dateRetour** prévue
- Un **employé** peut emprunter plusieurs matériels, mais un emprunt concerne un seul matériel.
- Un **matériel** peut être emprunté plusieurs fois à des dates différentes.

Travail demandé :

1. Identifier les entités du MCD et leurs propriétés (avec les identifiants). 2pts
2. Identifier les associations entre ces entités et les cardinalités correspondantes. 1,5pt
3. Construire le MCD. 1,5pt

Exercice 2 :

(5 points)

Un cabinet médical souhaite informatiser la gestion de ses consultations. Les règles de gestions suivantes ont été identifiées.

- Chaque **patient** est identifié par un **numéroPatient**, et on enregistre son **nom**, **prénom**, **sexe**, **date de naissance**, et **adresse**.
- Chaque **médecin** est identifié par un **matricule**, et on conserve son **nom**, **spécialité** et **téléphone**.
- Une consultation est caractérisée par un **numConsultation**, une **date**, un **diagnostic**, et un **montant**.
- Une consultation concerne un seul patient et est effectué par un seul médecin.

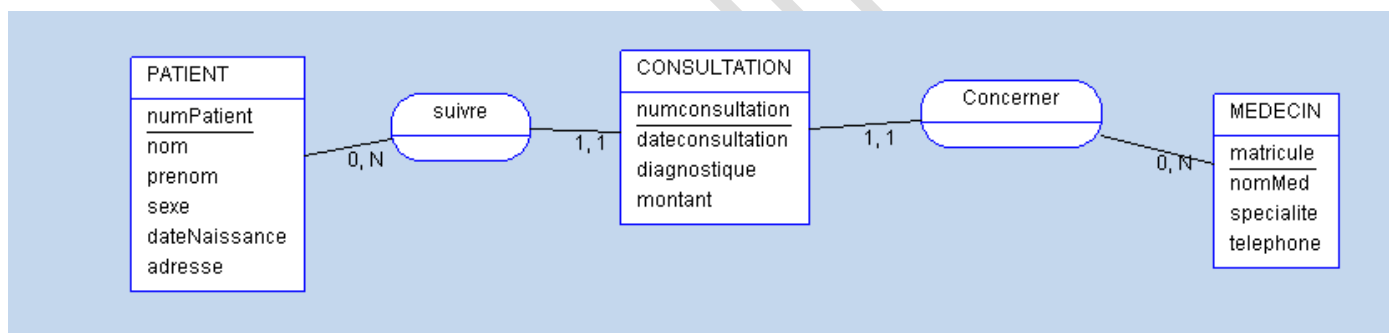
- Un patient peut avoir plusieurs consultations, et un médecin peut suivre plusieurs patients.

En stage dans ce cabinet, votre encadreur vous charge d'implémenter la base de données GES CAB correspondante.

1. Définir un modèle conceptuel de données, identifiant. **1.5pt**
2. Déterminer les entités, les associations portées sur le MCD en annexe (cocher la case correspondante) et préciser l'identifiant de chaque entité. **2pts**

Processus	entité	association	Identifiant (A compléter)
PATIENT			
Suivre			
CONSULTATION			
Concerner			
MEDECIN			

3. Indiquer en justifiant la réponse s'il est possible à partir de ce MCD :
 - a) Qu'une consultation soit réalisée par plusieurs médecins. **0,75pt**
 - b) Qu'un patient ne suive aucune consultation. **0,75pt**



CORRECTION DE L'ÉPREUVE D'INFORMATIQUE

Professeur : Soreil TCHEPDA

Niveau : Terminale

Année Scolaire : 2025–2026

PARTIE A : ALGORITHME ET LANGAGE C (10 points)

I. Compréhension de l'algorithme (2,25 pts)

1. Donner l'objectif principal de cet algorithme. (0,5 pt)

L'objectif est de vérifier si un patient, identifié par un code unique saisi par l'utilisateur, figure dans la liste des patients actuellement hospitalisés (recherche d'un élément dans un tableau).

2. Citer deux données d'entrée et une donnée de sortie de cet algorithme. (0,75 pt)

— **Données d'entrée** : Le nombre de patients N , le tableau des codes patients T , ou le code du patient à rechercher X . (Il suffit d'en citer deux parmi ces trois).

— **Donnée de sortie** : Le message indiquant si le patient est hospitalisé (et sa position) ou non.

3. Donner le rôle des variables i et $trouve$. (1 pt)

— i : C'est un compteur (ou indice) qui permet de parcourir le tableau T élément par élément, de 1 à N .

— $trouve$: C'est un booléen (drapeau) qui sert à mémoriser si le code recherché a été trouvé (VRAI) ou non (FAUX). Il permet aussi d'arrêter la boucle prématurément si l'élément est trouvé.

II. Analyse de l'algorithme (1,75 pts)

4. Identifier la méthode de recherche utilisée dans cet algorithme. (0,25 pt)

Il s'agit de la **recherche séquentielle** (ou recherche linéaire).

5. Justifier pourquoi la boucle Tant que peut s'arrêter avant d'atteindre la fin du tableau. (0,5 pt)

La boucle s'arrête dès que la condition ($Trouve = FAUX$) devient fausse, c'est-à-dire dès que $Trouve$ passe à VRAI. Cela se produit immédiatement lorsque l'élément recherché X est égal à l'élément courant $T[i]$. Ainsi, on ne continue pas à parcourir le reste du tableau inutilement.

6. Dire ce qui se passe si le code recherché ne figure pas dans le tableau. (1 pt)

Si le code n'est pas dans le tableau, la condition $T[i] = X$ ne sera jamais vraie. La variable $Trouve$ restera donc à FAUX tout au long de l'exécution. La boucle continuera jusqu'à ce que i dépasse N (condition $i \leq N$ devient fausse). À la fin, le test SI ($Trouve = VRAI$) échouera, et le programme affichera le message : « Le patient n'est pas hospitalisé ».

III. Exécution de l'algorithme (2 pts)

Hypothèses : $N = 4$, $T = ["PA10", "PA15", "PA20", "PA25"]$, $X = "PA20"$

Tableau d'exécution complété :

Etape	i	$T[i]$	$T[i]=X ?$	$trouve$	Commentaire
Initialisation	1	-	-	FAUX	Début de la recherche
Itération 1	1	PA10	FAUX	FAUX	$PA10 \neq PA20$, on incrémente i
Itération 2	2	PA15	FAUX	FAUX	$PA15 \neq PA20$, on incrémente i
Itération 3	3	PA20	VRAI	VRAI	$PA20 = PA20$, on arrête la boucle

Note : La boucle s'arrête ici car `trouve` est devenu VRAI.

Message affiché :

Le patient est hospitalise position 3

IV. Traduction en langage C (3,5 pts)

7. Citer les bibliothèques C à importer. (1 pt)

- Pour `printf()` et `scanf()` : `<stdio.h>`
- Pour les chaînes de caractères (comparaison `strcmp`, etc.) : `<string.h>`

8. Réécrire la déclaration des variables `N`, `i` et `trouve` en langage C. (1 pt)

```
int N, i;  
int trouve; // Ou bool trouve si <stdbool.h> est inclus, sinon int (0 ou 1)
```

Note : En C standard sans C99/bool, on utilise souvent `int` pour les booléens.

9. Traduire les trois lignes suivantes en langage C. (1,5 pt)

- Lire(`X`) (en supposant que `X` est un tableau de char `char X[20]`) :

```
scanf("%s", X);
```

- TANT QUE (`i <= N`) ET (`Trouve = FAUX`) FAIRE :

```
while (i <= N && trouve == 0) // ou !trouve  
{
```

- Afficher("Le patient est hospitalise position", `i`) :

```
printf("Le patient est hospitalise position %d", i);
```

PARTIE B : SYSTÈMES D'INFORMATION (10 points)

Exercice 1 : Gestion du prêt de matériel (5 points)

1. Identifier les entités du MCD et leurs propriétés (avec les identifiants). (2 pts)

- **EMPLOYE**
 - Identifiant : `matricule`
 - Propriétés : `nom`, `prenom`, `service_affectation`
- **MATERIEL**
 - Identifiant : `codeMat`
 - Propriétés : `designation`, `marque`, `etat`
- **EMPRUNT** (Cette entité est nécessaire car c'est une association many-to-many avec des attributs propres, ou une entité faible selon la modélisation choisie. Ici, l'énoncé dit « Chaque emprunt est identifié par un `numEmprunt` », c'est donc une entité à part entière ou une association portée).
 - Identifiant : `numEmprunt`
 - Propriétés : `dateEmprunt`, `dateRetour_prevue`

2. Identifier les associations entre ces entités et les cardinalités correspondantes. (1,5 pt)

- Association entre **EMPLOYE** et **EMPRUNT** :
 - Un employé peut faire plusieurs emprunts.

- Un emprunt concerne un seul employé.
- Cardinalités : EMPLOYE (1,N) — EMPRUNT (1,1) [Note : Si on considère l'emprunt comme une association directe entre Employé et Matériel, voir ci-dessous].

Analyse plus fine basée sur l'énoncé :

L'énoncé lie directement Employé, Matériel et Emprunt.

Règle : « Un employé peut emprunter plusieurs matériels, mais un emprunt concerne un seul matériel. » et « Un matériel peut être emprunté plusieurs fois ».

Il est plus classique de modéliser cela avec une association **EMPRUNTER** entre EMPLOYE et MATERIEL, où l'association elle-même porte les attributs de l'emprunt (date, etc.) et possède son propre identifiant si nécessaire, ou bien une entité centrale EMPRUNT.

Cependant, regardons les règles strictes :

- Employé (1,N) vers Emprunt.
- Matériel (1,N) vers Emprunt.
- Emprunt (1,1) vers Employé.
- Emprunt (1,1) vers Matériel.

Donc les associations sont :

- Effectuer** (entre EMPLOYE et EMPRUNT) : Un employé effectue 0,N emprunts. Un emprunt est effectué par 1,1 employé.
- Concerner** (entre MATERIEL et EMPRUNT) : Un matériel concerne 0,N emprunts. Un emprunt concerne 1,1 matériel.

3. Construire le MCD. (1,5 pt)

[Schéma décrit textuellement]

- Entité **EMPLOYE** (matricule, nom, prenom, service)
- Entité **MATERIEL** (codeMat, designation, marque, etat)
- Entité **EMPRUNT** (numEmprunt, dateEmprunt, dateRetour)

Liaison **EMPLOYE** —(1,N)— **EMPRUNT** —(1,1)— **MATERIEL**

Note : Souvent, dans les MCD simples, on met une association « Emprunter » entre Employé et Matériel avec les cardinalités (0,N) et (0,N), et on ajoute les attributs de l'association. Mais ici, l'existence d'un identifiant *numEmprunt* suggère fortement l'entité **EMPRUNT**.

Représentation standard MCD avec entité d'association :

EMPLOYE — Effectuer — EMPRUNT — Concerner — MATERIEL

- Cardinalités :
 - Employé → Effectuer : 1,N (Un employé fait plusieurs emprunts)
 - Effectuer → Employé : 1,1 (Un emprunt appartient à un seul employé)
 - Matériel → Concerner : 1,N (Un matériel est concerné par plusieurs emprunts historiques)
 - Concerner → Matériel : 1,1 (Un emprunt spécifique concerne un seul matériel)

Exercice 2 : Cabinet médical (5 points)

1. Définir un modèle conceptuel de données, identifiant. (1,5 pt)

Les entités et associations sont déjà partiellement données dans le tableau de la question 2. Le MCD doit relier PATIENT, MEDECIN et CONSULTATION.

2. Déterminer les entités, les associations portées sur le MCD en annexe et préciser l'identifiant de chaque entité. (2 pts)

Processus / Élément	Type	Identifiant (À compléter)
PATIENT	Entité	numPatient
Suivre	Association	(Pas d'identifiant propre, lien logique)
CONSULTATION	Entité (ou Association portée)	numConsultation
Concerner	Association	(Pas d'identifiant propre)
MEDECIN	Entité	matricule

Justification des types :

- **PATIENT**, **MEDECIN**, **CONSULTATION** sont des entités car elles ont des identifiants propres et des propriétés décrites.
- **Suivre** et **Concerner** sont des associations (verbes) reliant les entités.

Complétion du tableau :

- Ligne 1 : Entité | Identifiant : **numPatient**
- Ligne 2 : Association
- Ligne 3 : Entité | Identifiant : **numConsultation**
- Ligne 4 : Association
- Ligne 5 : Entité | Identifiant : **matricule**

3. Indiquer en justifiant la réponse s'il est possible à partir de ce MCD : (1,5 pt)

Analyse du schéma en bas de page :

Le schéma en bas indique :

- PATIENT (1,1) — Concerner — (0,N) CONSULTATION
- CONSULTATION (1,1) — Suivre — (0,N) MEDECIN ? *Non, regardons bien les chiffres.*
 - Côté Patient : 1,1 (Une consultation concerne 1 seul patient).
 - Côté Consultation (vers Patient) : 0,N (Un patient peut avoir 0 à N consultations).
 - Côté Médecin : 1,1 (Une consultation est effectuée par 1 seul médecin).
 - Côté Consultation (vers Médecin) : 0,N (Un médecin peut suivre 0 à N patients/consultations).

(a) Qu'une consultation soit réalisée par plusieurs médecins. (0,75 pt)

Réponse : Non.

Justification : La cardinalité du côté de l'entité MEDECIN par rapport à l'association « Suivre » (ou liant à Consultation) est 1,1 (ou selon l'autre sens, la consultation est liée à 1 seul médecin). L'énoncé précise également : « Une consultation [...] est effectuée par un seul médecin ». Dans le MCD, si la liaison entre CONSULTATION et MEDECIN a la cardinalité 1,1 du côté médecin pour une consultation donnée, alors une consultation ne peut pas être liée à plusieurs médecins.

(b) Qu'un patient ne suive aucune consultation. (0,75 pt)

Réponse : Oui.

Justification : La cardinalité minimale du côté de l'entité CONSULTATION par rapport au PATIENT est 0 (noté 0,N dans le schéma près de Consultation ou impliquant que le patient participe 0 à N fois). Cela signifie qu'un patient peut exister dans la base sans avoir encore eu de consultation enregistrée.