

| COLLEGE PRIVE LAIC MONGO BETI B.P 972 TEL 22 68 62 97/ 33 20 67 23 YAOUNDE |            |              |                |        |             |
|----------------------------------------------------------------------------|------------|--------------|----------------|--------|-------------|
| ANNEE SCOLAIRE                                                             | EVALUATION | EPREUVE      | CLASSE         | DUREE  | COEFFICIENT |
| 2025/2026                                                                  | N° I       | INFORMATIQUE | PREMIERES C, D | 02H    | 02          |
| Nom du Professeur : BIEM KIT MAKONGO EMMANUEL                              |            |              |                | Jour : | Qte         |
| NOMS ET PRENOMS : ----- Classe : -----                                     |            |              |                |        |             |

### PARTIE 1 : EVALUATION DES RESSOURCES (13 Points)

#### Exercice 1 : Rappel sur le Langage C

7.5pts

1) Définir les expressions suivantes :

2pts

a) Bibliothèque :-----

-----

b) Compilateur :-----

-----

2) Citer trois (03) exemples de Bibliothèques utilisés en C : -----1.5pt

3) Donner la signification des symboles suivants utilisés en C dans la fonction **scanf** 2pts

a) & :-----

b) % :-----

4) Donner la syntaxe de la boucle **for** en langage C :

2pts

#### Exercice 2 : Sur Les Algorithmes

5.5pts

1) Citer trois (03) structures Algorithmiques : -----

-----1.5pt

2) Donner une différence entre :

a). Un tableau et un enregistrement-----

-----1pt

b) Une structure de contrôle et une structure de données -----

-----1pt

3) Donner la syntaxe de déclaration d'un tableau en Algorithmique

2pts

## PARTIE 2 : EVALUATION DES COMPETENCES (7points)

On considère l'algorithme ci-dessous :

```
Algorithme Calcul  
Var i,n,p : Entier  
Debut  
Ecrire ("Saisir un Entier")  
Lire (n)  
i ← 2  
p ← 1  
tant que (i ≤ n) faire  
  p ← p * i  
  i ← i + 1  
tant que  
Ecrire ("Le résultat est :", p)  
Fin
```

- 1) Identifier dans cet algorithme :
  - a) La condition d'arrêt de la boucle: ----- 1pt
  - b) Une variable de sortie : ----- 0.5pt
  - c) Une instruction d'initialisation : ----- 0.5pt
  - d) Une instruction d'incrémentatation : ----- 0.5pt
- 2) Quelles seront les valeurs de i et de p à la fin de l'exécution de cet algorithme si on saisit 0 : ----- 1pt
- 3) Que produit cet algorithme si l'utilisateur saisit 4 ? ----- 1pt
- 4) En déduire ce que fait cet Algorithme : ----- 0.5pt
- 5) Réécrire cet Algorithme en remplaçant la boucle **tant que** par la boucle **Répéter** 2pts

*La Perfection N'habite Pas Dans Les Hommes Mais Dans Leurs Intentions*

---

# CORRECTION DE L'ÉPREUVE D'INFORMATIQUE

---

Classe : **Premières C, D**

Année Scolaire : **2025/2026**

Durée : **2h**

Coefficient : **2**

## PARTIE 1 : ÉVALUATION DES RESSOURCES (13 points)

### Exercice 1 : Rappel sur le Langage C (7,5 pts)

#### 1) Définition des expressions (2 pts)

- **Bibliothèque** : C'est un ensemble de fichiers (ou de code) contenant des fonctions, des variables et des types prédéfinis et prêts à l'emploi, permettant d'étendre les fonctionnalités du langage (ex : affichage, manipulation de chaînes, calculs mathématiques).
- **Compilateur** : C'est un programme (logiciel) qui permet de traduire un code source écrit dans un langage de haut niveau (comme le C) en langage machine (code objet ou exécutable) directement compréhensible par le processeur de l'ordinateur.

#### 2) Trois (03) exemples de bibliothèques utilisées en C (1,5 pt)

- `<stdio.h>` (Standard Input Output : pour les entrées/sorties comme `printf`, `scanf`)
- `<stdlib.h>` (Standard Library : pour les fonctions utilitaires comme `malloc`, `exit`)
- `<math.h>` (pour les fonctions mathématiques comme `sqrt`, `pow`)

(D'autres réponses valides : `<string.h>`, `<conio.h>`, `<ctype.h>`, etc.)

#### 3) Signification des symboles utilisés dans la fonction `scanf` (2 pts)

- **& (Esperluette)** : C'est l'opérateur d'adresse. Il permet d'indiquer au compilateur l'adresse mémoire de la variable où la donnée saisie devra être stockée.
- **% (Pourcentage)** : C'est le spécificateur de format. Il indique le type de la donnée à lire (ex : `%d` pour un entier, `%f` pour un réel, `%c` pour un caractère).

#### 4) Syntaxe de la boucle `for` en langage C (2 pts)

```
1 for (initialisation; condition; incrementation)
2 {
3     // Instructions a repeter
4 }
```

### Exercice 2 : Sur Les Algorithmes (5,5 pts)

#### 1) Citer trois (03) structures algorithmiques (1,5 pt)

- La structure **Séquentielle** (ou linéaire)
- La structure **Conditionnelle** (ou alternative : *Si...Alors...Sinon*)
- La structure **Itérative** (ou répétitive / bornée : *Pour, Tant que, Répéter*)

#### 2) Donner une différence entre (2 pts)

a) **Un tableau et une variable :**

Une **variable** est un espace mémoire nommé qui ne peut contenir qu'**une seule valeur** à la fois. Un **tableau** est une structure de données qui permet de stocker **plusieurs valeurs** (de même type) sous un seul nom, chaque valeur étant identifiée par un indice.

b) **Une structure de contrôle et une structure de données :**

Une **structure de contrôle** dicte le flux et l'ordre d'exécution des instructions dans l'algorithme (ex : les boucles, les conditions). Une **structure de données** définit la manière dont les informations sont organisées, représentées et stockées en mémoire (ex : tableau, enregistrement, liste).

**3) Syntaxe de déclaration d'un tableau en Algorithmique (2 pts)**

```
1 Var Nom_du_tableau : Tableau [borne_inf .. borne_sup] de Type_des_elements
```

(Exemple : *Var T : Tableau [1..10] de Entier*)

## PARTIE 2 : ÉVALUATION DES COMPÉTENCES (7 points)

### Rappel de l'algorithme :

```

1 Algorithme Calcul
2 Var i, n, p : Entier
3 Debut
4   Ecrire("Saisir un Entier")
5   Lire(n)
6   i <- 2
7   p <- 1
8   tant que (i <= n) faire
9     p <- p * i
10    i <- i + 1
11  fin tant que
12  Ecrire("Le resultat est: ", p)
13 Fin

```

#### 1) Identifier dans cet algorithme (2,5 pts)

- Condition d'arrêt de la boucle :  $i > n$**   
(La condition de poursuite est  $i \leq n$ , donc on s'arrête quand  $i > n$ )
- Une variable de sortie :  $p$**   
(C'est elle qui est affichée à la fin)
- Une instruction d'initialisation :  $i \leftarrow 2$**   
(ou  $p \leftarrow 1$ )
- Une instruction d'incrément :  $i \leftarrow i + 1$**

#### 2) Valeurs de $i$ et $p$ si on saisit 0 (1 pt)

Si  $n = 0$ , la condition de la boucle  $i \leq n$  (soit  $2 \leq 0$ ) est **fausse** dès le départ. La boucle n'est donc jamais exécutée.

| Variable | Valeur finale |
|----------|---------------|
| $i$      | <b>2</b>      |
| $p$      | <b>1</b>      |

#### 3) Ce que produit l'algorithme si l'utilisateur saisit 4 (1 pt)

- **Tour 1** :  $i=2$ ,  $p = 1 \times 2 = 2$ , puis  $i$  devient 3.
- **Tour 2** :  $i=3$ ,  $p = 2 \times 3 = 6$ , puis  $i$  devient 4.
- **Tour 3** :  $i=4$ ,  $p = 6 \times 4 = 24$ , puis  $i$  devient 5.
- **Arrêt** :  $5 \leq 4$  est faux.

⇒ **Résultat affiché** : Le résultat est: 24

#### 4) Ce que fait cet algorithme (0,5 pt)

Cet algorithme calcule et affiche la **factorielle** de l'entier  $n$  saisi par l'utilisateur (pour  $n \geq 1$ ).

$$n! = 1 \times 2 \times 3 \times \cdots \times n$$

#### 5) Réécrire cet algorithme en remplaçant la boucle « Tant que » par la boucle « Répéter » (2 pts)

```

1 Algorithme Calcul
2 Var i, n, p : Entier
3 Debut
4   Ecrire("Saisir un Entier")
5   Lire(n)

```

```
6      i <- 2
7      p <- 1
8
9      // Remplacement de la boucle
10     Repeter
11         p <- p * i
12         i <- i + 1
13     Jusqu'a (i > n)
14
15     Ecrire("Le resultat est: ", p)
16 Fin
```

[colback=yellow !10 !white, colframe=orange !70 !black, title=Note pédagogique] Bien que la structure soit correcte, il est bon de noter que si l'utilisateur saisit 0 ou 1, la boucle « Répéter » s'exécutera au moins une fois contrairement au « Tant que », ce qui donnerait un résultat erroné pour la factorielle de 0. Cependant, dans le cadre strict de la consigne « remplacer la boucle », la traduction structurelle ci-dessus est celle attendue par le correcteur.