

LYCEE BILINGUE DE YAOUNDE

Evaluation	Epreuve	Département	Durée	Session	Coef	Classe
Eval N°2	Informatique	INFORMATIQUE	2heures	Nov 2024	2	1 ^{ère} C

Exercice 1 :

(3pts)

1. Définir : structure de données, structure algorithmique 1pt
2. Citez 02 exemples de structure de données 1pt
3. Citez 02 exemples de structure algorithmique 1pt

Exercice 2 :

(5pts)

- 1) Ecris un algorithme qui demande à l'utilisateur la largeur et la longueur d'un champ rectangulaire et qui affiche le périmètre et la surface 2pts
Remarque : périmètre = (largeur+longueur)*2 et surface=largeur*longueur
- 2) CANAL+ souhaite informatiser la gestion de ses abonnés. pour cela, il sollicite votre aide. Sachant qu'un abonné est caractérisé par son nom, prénom, sexe, sa localité, numéro du decodeur.
 - Définissez la structure qui permet de représenter un abonné. 1pt
 - Déclaré un objet noté AB de cette structure 1pt
 - Initialisez cet objet avec les données suivantes : (nom : Kuete, Prénom : Alex, sexe : masculin, localité : Bastos, numéro : 968507) 1pt

Exercice 3 :

(4pts)

- (1). Algorithme moyenne_age;
- (2). Var i ,age,som : entier ;
- (3). Debut
- (4). i←0 ; age←0 ;
- (5). Repeter
- (6). Ecrire('entrer l'âge du personne ');
- (7). Lire('age') ;
- (8). i←i+1 ;
- (9). Som←som+ age ;
- (10). Ecrire('Encore une autre personne (O/N) ?')
- (11). Lire(rep) ;
- (12). Jusqu'à (rep='N')
- (13). Ecrire('la moyenne d'âges des personnes est : ',som/i) ;
- (14). fin

- a) Combien d' instruction compte cet algorithme ? 1pt
- b) Quelle est le type structure algorithmique utilisée dans cet algorithme ? Justifiez. 1 pt
- c) Quels sont les rôles des instructions de la ligne (4) ? 1pt
- d) Quelles sont les instructions de saisie utilisées dans cet algorithme ? 1pt

Exercice 4 : (2pts)

Ecrire un algorithme qui initialise un tableau avec 10 nombres entiers saisi par utilisateur et les affiche ensuite à l'écran.

Exercice 5 : Programmation langage C 6pts

- 1- Définir : Bibliothèque, Paramètre 2pts
- 2- Donner le rôle des symboles suivant en programmation C
 % et & 2pts
- 3- Traduire l'algorithme de l'exercice 4 en langage C 2pts

Correction de l'évaluation n°2 d'Informatique

Lycée Bilingue de Yaoundé – Session de novembre 2024

Classe : 1^{ère} Durée : 2 heures Coefficient : 2

Exercice 1

(3 pts)

1. Définitions (1 pt)

- **Structure de données** : manière d'organiser et de regrouper des données en mémoire pour pouvoir les stocker et les manipuler facilement.
- **Structure algorithmique** : manière d'organiser les instructions d'un algorithme. Elle détermine l'ordre dans lequel les instructions sont exécutées.

2. Deux exemples de structures de données (1 pt)

Le tableau et l'enregistrement (structure). On peut aussi citer la liste, la pile ou la file.

3. Deux exemples de structures algorithmiques (1 pt)

- la structure conditionnelle (Si ... Alors ... Sinon);
- la structure itérative (Pour, Tant que, Répéter ... Jusqu'à).

La structure séquentielle est aussi acceptée.

Exercice 2

(5 pts)

1) Périmètre et surface du champ (2 pts)

```
Algorithme champ_rectangulaire;  
Var largeur, longueur, perimetre, surface : réel;  
Debut  
    Ecrire('Entrer la largeur : ');  
    Lire(largeur);  
    Ecrire('Entrer la longueur : ');  
    Lire(longueur);  
    perimetre ← (largeur + longueur) * 2;  
    surface ← largeur * longueur;  
    Ecrire('Le périmètre est : ', perimetre);  
    Ecrire('La surface est : ', surface);  
Fin
```

2) CANAL+ Structure qui représente un abonné (1 pt)

```
Type Abonne = Structure  
    nom : chaîne;  
    prenom : chaîne;  
    sexe : chaîne;  
    localite : chaîne;  
    numero : entier;  
FinStructure
```

Déclaration de l'objet AB (1 pt)

```
Var AB : Abonne;
```

Initialisation de AB (1 pt)

```
AB.nom ← 'Kuate';  
AB.prenom ← 'Alex';  
AB.sexe ← 'masculin';  
AB.localite ← 'Bastos';  
AB.numero ← 968507;
```

Exercice 3

(4 pts)

a) **Nombre d'instructions (1 pt)** L'algorithme compte **14 lignes**. Parmi elles, les lignes 1, 2, 3, 5, 12 et 14 sont des en-têtes, des déclarations ou des mots-clés. Il y a **9 instructions proprement dites** : 2 affectations à la ligne 4, puis les lignes 6, 7, 8, 9, 10, 11 et 13. Le corrigé officiel attend probablement « 14 », mais on peut préciser les 9 instructions exécutables.

b) **Structure algorithmique utilisée (1 pt)** C'est la **structure itérative (répétitive)** Répéter ... Jusqu'à. Les instructions des lignes 6 à 11 sont répétées jusqu'à ce que la condition $rep = 'N'$ soit vraie. La condition est testée à la fin, donc le bloc s'exécute au moins une fois.

c) **Rôle des instructions de la ligne (4) (1 pt)** Ce sont des **affectations d'initialisation** : $i \leftarrow 0$ initialise le compteur de personnes à 0, et $age \leftarrow 0$ initialise la variable `age` à 0.

d) **Instructions de saisie (1 pt)** Lire('age') et Lire(rep) (lignes 7 et 11).

Remarque : la variable som n'est pas initialisée à 0 avant la boucle, ce qui est une erreur de l'algorithme.

Exercice 4

(2 pts)

```
Algorithme remplir_tableau;  
Var T : tableau[1..10] de entier;  
    i : entier;  
Debut  
    Pour i de 1 à 10 faire  
        Ecrire('Entrer le nombre ', i, ' : ');  
        Lire(T[i]);  
    FinPour  
    Ecrire('Les éléments du tableau sont : ');  
    Pour i de 1 à 10 faire  
        Ecrire(T[i]);  
    FinPour  
Fin
```

1) Définitions (2 pts)

- **Bibliothèque** : fichier contenant des fonctions déjà écrites, que l'on inclut dans un programme avec `#include` (exemple : `stdio.h` pour `printf` et `scanf`).
- **Paramètre** : variable ou valeur transmise à une fonction pour lui fournir les données dont elle a besoin.

2) Rôle des symboles (2 pts)

- `%` : c'est l'opérateur **modulo**, qui donne le reste de la division entière (exemple : `7 % 2 = 1`). Dans `printf` et `scanf`, il introduit aussi un **format** (`%d`, `%f`, `%c`).
- `&` : c'est l'**opérateur d'adresse**. Il donne l'adresse mémoire d'une variable (exemple : `scanf("%d", &x);`).

3) Traduction de l'exercice 4 en C (2 pts)

```
#include <stdio.h>

int main()
{
    int T[10];
    int i;

    for (i = 0; i < 10; i++)
    {
        printf("Entrer le nombre %d : ", i + 1);
        scanf("%d", &T[i]);
    }

    printf("Les elements du tableau sont :\n");
    for (i = 0; i < 10; i++)
    {
        printf("%d\n", T[i]);
    }

    return 0;
}
```