

Réponses à l'examen de structure de données avancées

Partie 1 : Notions et concepts de base (4 pts)

1. Définir les termes suivants : fonction, paramètre

- **Fonction** : Un bloc de code réutilisable qui effectue une tâche spécifique.
- **Paramètre** : Une variable passée à une fonction pour lui fournir des données d'entrée.

2. Quel est l'avantage d'un tableau ? d'un enregistrement ?

- **Tableau** : Permet de stocker et d'accéder facilement à plusieurs éléments de même type.
- **Enregistrement** : Permet de regrouper des données de types différents sous une même structure.

3. Lorsqu'on fait référence à la complexité d'un algorithme, quels sont les critères ?

Les principaux critères de complexité d'un algorithme sont :

- Le temps d'exécution (complexité temporelle)
- L'espace mémoire utilisé (complexité spatiale)

4. Quelle est la différence entre une variable globale et une variable locale ?

- **Variable globale** : Accessible dans tout le programme.
- **Variable locale** : Accessible uniquement dans le bloc de code où elle est définie.

Partie 2 : Synthèse et compréhension du cours (6 pts)

1. A quoi sert un sous-algorithme ?

Un sous-algorithme (ou sous-fonction) sert à décomposer un problème complexe en sous-problèmes plus simples et réutilisables.

2. Citez les types de sous-algorithmes qui existent et les différencier.

Les principaux types de sous-algorithmes sont :

- Sous-algorithme avec paramètres
- Sous-algorithme sans paramètre
- Sous-algorithme récursif

3. Quelle est la différence entre les deux modes de passage en paramètres dans les sous-algorithmes ? (0,75 * 2 pts)

- **Passage par valeur** : La valeur du paramètre est copiée dans le sous-algorithme, les modifications n'affectent pas la variable d'origine.
- **Passage par référence** : L'adresse de la variable est passée au sous-algorithme, les modifications affectent la variable d'origine.

4. Différence entre un paramètre par valeur et un paramètre par variable (0,5 * 2 pts)

- **Paramètre par valeur** : La valeur du paramètre est copiée dans le sous-algorithme.
- **Paramètre par variable** : L'adresse de la variable est passée au sous-algorithme.

5. Quelle est la différence entre un enregistrement et un tableau d'enregistrement ? (1 pt)

- **Enregistrement** : Une structure regroupant des données de types différents.

- **Tableau d'enregistrement** : Un tableau contenant des enregistrements, permettant de stocker et d'accéder à plusieurs structures de même type.

Partie 3 : Étude de cas fondamental (4 pts)

1. **Écrire une fonction qui calcule la factorielle d'un nombre entier positif et la traduire en programme.**

```
// Fonction factorielle en pseudo-code
fonction factorielle(n):
  si n = 0 alors
    retourner 1
  sinon
    retourner n * factorielle(n-1)
  fin si
fin fonction

// Programme en Java
public int factorielle(int n) {
  if (n == 0) {
    return 1;
  } else {
    return n * factorielle(n-1);
  }
}
```

2. **Écrire un algorithme qui prend en entrée trois entiers et qui les trie par ordre croissant, puis le traduire en programme.**

```
// Algorithme de tri en pseudo-code
algorithme tri_croissant(a, b, c):
  si a > b alors
    échanger a et b
  fin si
  si b > c alors
    échanger b et c
  fin si
  si a > b alors
    échanger a et b
  fin si
  retourner a, b, c
fin algorithme

// Programme en Java
public int[] tri_croissant(int a, int b, int c) {
  if (a > b) {
    int temp = a;
    a = b;
    b = temp;
  }
  if (b > c) {
    int temp = b;
    b = c;
    c = temp;
  }
  if (a > b) {
    int temp = a;
    a = b;
    b = temp;
  }
  return new int[]{a, b, c};
}
```

Partie 4 : Étude de cas Complexe (6 pts)

1. **Créer la structure de Fournisseur (1 pt)**

```
// Structure Fournisseur en pseudo-code
structure Fournisseur:
  nom : chaîne de caractères
  prénomF : chaîne de caractères
  âge : entier
fin structure

// Structure Fournisseur en Java
public class Fournisseur {
  public String nom;
  public String prénomF;
  public int âge;
}
```

2. Écrire la fonction Différence_age (1.25 pt)

```
// Fonction Différence_age en pseudo-code
fonction Différence_age(age1, age2):
  retourner |age1 - age2|
fin fonction

// Fonction Différence_age en Java
public int Différence_age(int age1, int age2) {
  return Math.abs(age1 - age2);
}
```

3. Écrire un algorithme qui permet de saisir l'âge du Fournisseur1 et du Fournisseur 2 et affiche la différence d'âge entre les deux (utiliser la fonction Différence_age) (2.5 pts)

```
// Algorithme en pseudo-code
algorithme afficher_différence_age(Fournisseur1, Fournisseur2):
  lire âge_Fournisseur1, âge_Fournisseur2
  différence_âge = Différence_age(âge_Fournisseur1, âge_Fournisseur2)
  afficher "La différence d'âge entre les deux fournisseurs est de", différence_âge, "ans"
fin algorithme

// Algorithme en Java
public void afficher_différence_age(Fournisseur Fournisseur1, Fournisseur Fournisseur2) {
  int âge_Fournisseur1 = Fournisseur1.âge;
  int âge_Fournisseur2 = Fournisseur2.âge;
  int différence_âge = Différence_age(âge_Fournisseur1, âge_Fournisseur2);
  System.out.println("La différence d'âge entre les deux fournisseurs est de " + différence_âge + " ans");
}
```

4. Qu'affiche à l'écran l'algorithme suivant (1.25 pt)

```
// Algorithme Passage par_adresse en pseudo-code
Algorithme Passage par_adresse
Variables n, p : entiers
// Déclaration de la procédure échange
Procédure échange(Var a : entier, Var b : entier)
Variable e : entier
Début
Ecrire("Début échange ", a, " ", b)
a <- b
b <- e
Ecrire("Fin échange ", a, " ", b)
FinProc
// Programme principal
Début
n <- 10
p <- 20
Ecrire("Avant l'appel ", n, " ", p)
échange n, p
Ecrire ("Fin échange ", n, " ", p)
Fin
```

Cet algorithme affichera à l'écran :

```
Avant l'appel 10 20  
Début échange 10 20  
Fin échange 20, b  
Fin échange 20 10
```