

Contrôle Terminal – 1^{ère} session

1h30 - Documents non autorisés

Nota : Chaque partie étant indépendante, vous pouvez traiter les exercices dans l'ordre de votre choix.

Partie 1 – divers (3 points)

1. Donner trois exemples de structures de données linéaires ? (1,5 pt)

```
#include <stdio.h>
#include <stdlib.h>

int f(int n){
    switch (n) {
        case 0:
            return 0;
        case 1:
            return 1;
        default:
            return (f(n-1) + f(n-2));
    }
}

int main(){
    int i;
    int *t = malloc(sizeof(int)*10);
    for (int i=0;i<10;i++)
        *(t+i) = f(i);
    for (i=0; i<10; i++)
        printf("%d\t", t[i]);
    printf("\n");
    return 0;
}
```

2. Après avoir compilé, quelles impressions sont effectuées par ce programme sur le terminal ? (1, 5 pts)

Partie 2 – Docs en stock (7 points)

On souhaite créer un programme en C qui gère un ensemble de documents. Un document est caractérisé par les champs code (un entier long), un nom (chaîne de caractères) et une rubrique d'appartenance (trésorerie, communication, etc.)

3. Créer la ou les structures de votre choix qui permettent de gérer un ensemble de documents. Discuter des avantages/inconvénients de votre implémentation (2,5 pts)
4. Créer la fonction `ajouteDocument` qui ajoute un document à l'ensemble des document déjà enregistrés. (1 pt)
5. Créer la fonction `supprimeDocument` qui supprime un document de l'ensemble (1 pt)
6. Créer la fonction `triParNom` qui affiche la liste triée par ordre alphabétique des documents. Comment pourrait-on modifier le stockage des documents afin que le tri soit plus performant ? (2,5 pts)

Partie 3 – des formes ... (5 pts)

1. Définir tout d'abord une structure en C « **Forme** » permettant de représenter une forme qui peut être un **cercle** ou un **carré**. (1 pt)

Ecrire les fonctions suivantes :

1. `Forme creer_forme(<arguments à définir>)` qui permet d'initialiser une cercle ou un carré (2 pts)
2. `float perimetre(Forme f)` qui permet de calculer le périmètre du cercle ou du carré contenu dans la forme (2 pts)

Partie 4 – polynômes ... (5 pts)

Nous considérons ici des polynômes à coefficients entiers.

1. **Création de la structure.** Définir une structure en C permettant de gérer des polynômes de degré r (1 pt)
2. **Création de polynôme.** Ecrire la fonction `Polynome Poly_random(int degre)` qui crée un polynôme aléatoire de degré r (1,5 pt)
3. **Affichage de polynôme.** Ecrire la fonction `void Affiche(Polynome p)` qui affiche un polynôme (1 pt)
4. **Evaluation de polynôme.** Ecrire la fonction `int Eval(Polynome p, int x)` qui évalue le polynôme p au point x (1,5 pt)