SUP Formations Certification — Service Universitaire de Pédagogie — CTU
UNIVERSITÉ DE FRANCHE-COMTÉ

Licence 2 – Année 2021-2022

Programmation Avancée

Durée : 3 heures – **Session** 1 du Mai 2022

Ce sujet comporte : 3 page(s)

Matériel autorisé : Aucun document

Exercice 1 : Cours (5 points)

Question 1 : Répondez aux questions suivantes :

1. Quel caractère permet de déterminer la fin d'une chaîne de caractères ?
2. Quelle est l'utilité d'une classe abstraite ?
3. Qu'est-ce qu'une méthode virtuelle pure ?
4. Qu'est-ce que le polymorphisme ?
5. Qu'est-ce que les tests en boîte noire ?
6. Qu'est-ce que la récursivité ?
7. En C, quel impact aura le mot clé static devant le nom d'une méthode ?
8. En C, quelles sont les méthodes pour allouer et libérer de la mémoire ?
9. À quoi servent les espaces de noms ?
10. À quoi sert la couverture de code ?
11. Qu'est-ce qu'un pointeur ?

Exercice 2 : Top Model (2 points)

Modélisez en UML le besoin suivant :
— classe abstraite joueur ayant une méthode jouer.
— classe humain héritant de joueur.
— classe ordinateur héritant de joueur.

Exercice 3 : Qu'est-ce ?(3 points)

Question 1 : Expliquez ce que fait le programme ci-dessous.

Question 2 : Qu'est-ce qui sera affiché dans la console après exécution ?

```
1  #include <iostream>
2
3  int mystere(char a, char* b)
4  {
5     int rc = -1;
6
7     if (NULL != b)
8     {
9        rc = 1;
10    }
11
12    while (*b && rc)
13    {
14       if (*b == a)
15       {
16          *b = '.';
17       }
18       b++;
19    }
20
21    return rc;
22 }
23
24 int main()
25 {
26    char texte[] = "je suis un texte.\0";
27
28    std::cout << texte << std::endl;
29    int rc = mystere('e', texte);
30    std::cout << rc << std::endl;
31    std::cout << texte << std::endl;
32
33    return 0;
34 }
```

Exercice 4 : Parcours (10 points)

On s'intéresse ici au stockage de données sous forme d'arbre. Un arbre binaire est un arbre où chaque noeud peut avoir au maximum deux branches. Pour différencier les branches, on les nomme souvent gauche et droite (left, right). La figure 1 présente un arbre binaire contenant sept valeurs.

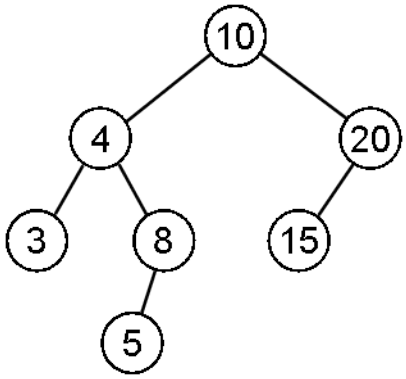

FIGURE 1 – Arbre binaire

Un noeud est représenté par une valeur appelée clé et possède donc au maximum deux sous-arbres. On peut ainsi représenter un noeud par la structure suivante :

```
1 typedef struct _node_
2 {
3    uint32_t key;
4    struct sNode *left;
5    struct sNode *right;
6 } sNode ;
```

Un noeud sans parent est appelé racine, une noeud sans sous-abre est appelée feuille, les valeurs sont ajoutées ordonnées de la façon suivante :

- chaque noeud du sous-arbre gauche a une clé inférieure à celle du noeud considéré.
- chaque noeud du sous-arbre droit possède une clé supérieure à celle-ci.
- l'arbre ne peut contenir deux noeuds de clé identique.

Question 1 : Proposez une méthode permettant d'ajouter une valeur dans un arbre.

Question 2 : Proposez une méthode indiquant si une valeur est présente dans l'arbre.

Question 3 : Proposez une méthode permettant d'afficher toutes les valeurs d'un arbre.

Suggestion de prototypes :

```
1    void add(sNode **tree, uint32_t key);
2    bool has(sNode **tree, uint32_t key);
3    void print(sNode **tree);
```