



Mercredi 17 Janvier 2018

Devoir semestriel (S3)

Module : **Informatique**

Durée : **2h00**

Exercice 1 (6 pts)

1. Ecrire une fonction nommée **dupliquer** permettant de dupliquer les éléments pairs d'une liste chaînée d'entiers, et de garder les éléments impairs inchangés. Par exemple la liste $\langle 3, 5, 10, 7, 8 \rangle$ sera modifiée en $\langle 3, 5, 10, 10, 7, 8, 8 \rangle$. Cette fonction doit avoir comme arguments la liste à dupliquer et devra retourner la nouvelle liste obtenue.

2. Une liste chaînée A est dite **includ** dans une liste chaînée B si les éléments de A appartiennent à B dans le même ordre et successivement. Ecrire une fonction nommée **includ** qui permet de vérifier si une liste A d'entiers est incluse dans une liste B. Cette fonction doit avoir comme arguments les deux listes et devra retourner 1 si A est incluse dans B et 0 sinon.

Exemples :

- La liste A contenant les valeurs $\langle 2, 8, 4, 12 \rangle$ est incluse dans la liste B contenant les valeurs $\langle 3, 2, 8, 4, 12, 5 \rangle$.
- La liste A contenant les valeurs $\langle 2, 8, 4, 12 \rangle$ n'est pas incluse dans la liste B contenant les valeurs $\langle 3, 2, 8, 6 \rangle$.

3. On se donne deux files F et Q et une pile P. La file F contient une suite d'entiers, Q et P sont initialement vides. Ecrire une fonction nommée **transférer** qui met dans Q les entiers impairs de F et laisse dans F les entiers pairs. Notons que les entiers doivent être dans un ordre inversé que celui du début pour les deux files F et Q. Par exemple, si la file F contient les éléments $[12, 13, 6, 15, 7, 11, 28]$ (**12 est le premier élément de F**), la file F sera $[28, 6, 12]$ et la file Q sera $[11, 7, 15, 13]$. (Utiliser les fonctions de manipulation des piles et des files sans donner leurs codes correspondants).

Exercice 2 (6 pts)

Soit la liste des valeurs suivantes :

26 20 32 38 53 10 29 34 23 6 15 72

- Construire l'arbre binaire de recherche (ABR) correspondant à cette liste.
- Donner l'ordre des nœuds visités selon le parcours préfixé de l'arbre.
- Ajouter successivement à l'ABR obtenu les valeurs **17**, **27** et **33**. Donner le nouveau ABR obtenu.
- A partir de l'ABR obtenu (question 3), supprimer successivement les valeurs **32**, **23** et **20**. Donner le nouveau ABR obtenu.
- Ecrire une fonction récursive **afficher_feuilles** qui permet d'afficher les feuilles d'un arbre binaire.
- Ecrire une fonction récursive **hauteur** qui permet de calculer la hauteur d'un arbre binaire, cette fonction doit avoir comme argument la racine de l'arbre et devra retourner la hauteur obtenue.
- Un arbre binaire est dit **AVL** si pour chaque nœud de l'arbre, la différence entre la hauteur du sous arbre gauche et la hauteur du sous arbre droit est d'au plus un. Ecrire une fonction récursive **AVL** qui permet de vérifier si un arbre binaire est AVL en faisant appel à la fonction **hauteur**, cette fonction doit avoir comme argument la racine de l'arbre et devra retourner 0 ou 1.

Exercice 3 (3 pts)

Soit les deux programmes suivants:

P1.c

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

struct element{
    int donnee;
    element * suivant;
};

element * operation(element * L1, element * L2, int n){
    element * tmp;
    int i;

    tmp = L1;
    for(i = 1; tmp!=NULL && i < n; i++){
        tmp = tmp->suivant;
    }
    if(tmp == NULL)
        return L2;
    tmp->suivant = L2;
    return L1;
}
```

P2.c

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

struct pile{
    int donnee;
    pile * precedent;
};

pile * mystere(pile *P){
    pile * A,* B;
    int x, y, z;

    A = NULL ; B = NULL ;
    x = depiler(&P);
    while(est_vide(P)==0){
        y = depiler(&P);
        if(y > x && y % 2 != 0)
            A = empiler(A, y);
        if(y <= x && y % 2 == 0)
            B = empiler(B, y);
    }
    while(est_vide(B) == 0){
        z = depiler(&B);
        P = empiler(P, z);
    }
    P = empiler(P, x);
    while(est_vide(A) == 0){
        x = depiler(&A);
        P = empiler(P, x);
    }
    return P;
}
```

1. Donner le résultat d'exécution pour les listes L1 d'exécution de la fonction **operation** pour la liste L.
2. Donnez la pile obtenue après exécution la fonction **mystere** pour la pile P = [8, 9, 11, 15, 12, 6, 10] 10 est le sommet de la pile P. (On suppose qu'on fait appel aux fonctions de manipulation des piles).

Exercice 4 (4 pts)

Soit un graphe représenté par la matrice d'adjacence suivante :

		0	1	2	3	4	5	6
0	S		11	13				
1	A			1	3			
2	B					6		
3	C						5	
4	D						8	12
5	E							2
6	F							

1. Donner le graphe correspondant à cette représentation.
2. Quel est l'ordre des sommets visités selon un parcours en profondeur du graphe ?
3. Donner le plus court chemin ainsi que sa longueur depuis le sommet S jusqu'au sommet F. Préciser le nom de l'algorithme utilisé.
4. Que se passe-t-il si la valeur de l'arc (B, D) devient -6 ?