

Durée : 30 minutes

Algorithmique et programmation C II

Nom et prénom :Groupe : A B

Soit les déclarations des trois structures suivantes :

```
typedef struct {  
    char nom[30];  
    float note, coef;  
}matiere;
```

```
typedef struct cel{  
matiere info;  
struct cel *suiv;  
}cellule;
```

```
typedef struct {  
cellule *tete;  
int taille;  
}list;
```

1. (5 points) Soit la déclaration de la variable liste suivante :

```
list l;
```

Quel est **le type** de chacune de ces expressions :

- (a) `l.taille` :

Solution: entier ou int

- (b) `l.tete` :

Solution: pointeur sur cellule ou cellule*

- (c) `l.(*tete).info.note` :

Solution: réel ou float

- (d) `l.tete->info.nom` :

Solution: chaîne ou char[30]

2. (5 points) Écrire une **fonction ajoutFin()** qui prend en paramètre une liste (list) et une matière et ajoute une cellule en fin de la liste en question et retourne la liste modifiée.

Solution:

```
list ajoutFin(list L, matiere m){ //0,25pt  
    cellule *p, *r; //0,25pt  
    p=(cellule*) malloc(sizeof*p); //0,5pt  
    p->info=m; //0,25pt
```

```

p->suiv=NULL;          //0,25 pt
    if(l.tete==NULL) L.tete=p;    //0,5 pt
    else{
        r=l.tete;          //0,5 pt
        while(r->suiv!=NULL)    //0,5 pt
            r=r->suiv;          //0,5 pt
        r->suiv=p;          //0,5 pt
    }
    L.taille++;          //0,5 pt
return (L);          //0,5 pt
}

```

3. (10 points) Écrire une **fonction récursive** en langage C occ (l : list, n : réel) : entier qui retourne le nombre d'occurrence de la note n dans une liste l.

Solution:

```

int occ(list l, float n) //1pt
{
    cellule *p ;          //0,5 pt
    if(l.tete==NULL) //1Pt    return 0; //1pt
    p= l.tete; //1pt
        l.tete=l.tete->suiv; //1pt
        l.taille--; //1pt
    if(p->info.note==n) //1pt
        return(1+occ(l,n)); //1pt
    else //0.5 Pt
        return(occ(l,n)); //1pt
}
}

```