

**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];  
    double note, coef;  
}matiere;
```

```
typedef struct cel{  
matiere donnee;  
struct cel *next;  
}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` :

**Solution:** list

`l->tete` :

(b) `(*l).tete->next` :

**Solution:** cellule \* ou pointeur sur cellule

`l->tete->donnee.nom` :

**Solution:** chaine ou char[30]

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

**Solution:**

```
list ajoutTete(list L, matiere m){  
    cellule *p;  
    p=(cellule*) malloc (sizeof*p);  
    p->donnee=m;  
    p->next=L.tete;  
    L.tete=p;
```

```
                L.taille++;  
return (L);  
}
```

3. (10 points) Écrire une fonction récursive SomNotPos() qui prend une liste chaînée de matières (list) calcule et retourne la **somme pondérée** des notes  $\geq 10$  de la liste des matières.

**Solution:**

```
double SomNotPos(list l){  
    cellule *p;  
  
    if(l.taille==0) { return 0;}  
    else{  
        p=l.tete;  
        l.tete=l.tete->next;  
        l.taille--;  
        if(p->donnee.note>10)  
            return (p->donnee.note*p->donnee.coef+ SomNotPos(l));  
        else  
            return (                SomNotPos(l));  
    }  
}
```