

Contrôle continu d'Informatique III

Exercice 1 : 5 points

On code les listes d'entiers au moyen d'une liste des cellules d'entiers de la manière suivante :

```
typedef struct cellule_entier{
    int donnee ;
    struct cellule_entier *suivant ;
}CelluleEntier ;
typedef Entier *Liste;
```

Programmez en langage C une fonction récursive prenant comme argument une liste d'entiers et renvoyant cette liste triée. (On se servira des algorithmes vus en cours)

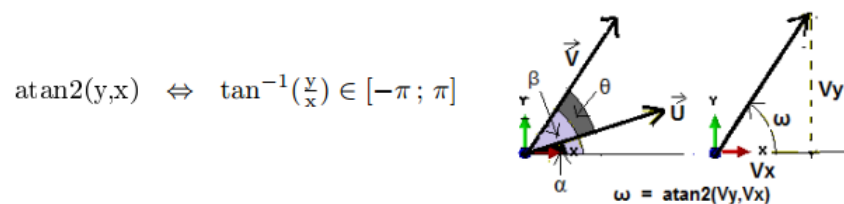
Exercice 2 : 10 points

On code les vecteurs au moyen d'un tableau deux réels de la manière suivante :

```
typedef float Vecteur[2]; //V[0] représente l'abscisse
```

Implémentez les méthodes suivantes : U et V sont de type Vecteur.

- *SontColineaires*(U,V) : fonction qui retourne la colinéarité de U et V
- *ProduitScalaire*(U,V) : fonction qui retourne le produit scalaire de U par V
- *SontOrthogonaux*(U,V) : fonction qui retourne l'orthogonalité de U et V
- *Norme*(U) : fonction qui retourne la norme de U.
- *Cosinus*(U,V) : fonction qui retourne le cosinus de la mesure de l'angle $(\widehat{U,V})$
- *MesureAngle*(U,V) : fonction qui retourne la mesure en degré de l'angle $(\widehat{U,V})$
- *Sinus*(U,V) : fonction qui retourne le sinus de la mesure de l'angle $(\widehat{U,V})$



Exercice 3 : 5 points

On code les polynômes au moyen d'une liste des monômes de la manière suivante :

```
typedef struct monome{
    float coeff;
    int degre;
    struct monome *suivant;
}Monome ;
typedef Monome *Polynome;
```

Implémentez les méthodes suivantes : P est de type Polynome

- *DeriveePoly*(P) : procédure qui dérive P sur place.
- *PrimitivePoly*(P) : procédure qui intègre P sur place.