



Partiel d'Algorithmique

Barème: exercice 1 (4 points), exercice 2 (4 points), exercice 3 (4 points).

Exercice 1 : Soit la fonction suivante :

```
int Mystere(int *t, int n, int v)
{
    if(n < 0)
        return v;
    else
    {
        if(*t > v)
            v = *t;

        return Mystere(++t, --n, v);
    }
}
```

En déduire ce que fait la fonction « Mystere » pour l'appel :

```
int var = Mystere(array, N-1, *array);
```

Exercice 2 : Soit un tableau T contenant des nombres réels qui représentent les moyennes générales de N étudiants.

1. Donnez une fonction itérative permettant de vérifier si le taux de réussite dépasse la moitié.
2. Proposez une fonction récursive pour cette vérification.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|------|---|-------|----|----|---|-----|---|
| 9 | 7 | 10.5 | 4 | 11.75 | 15 | 10 | 3 | 9.5 | 8 |

 Return 0

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|------|----|-------|----|----|---|-----|---|
| 19 | 7 | 10.5 | 14 | 11.75 | 15 | 10 | 3 | 9.5 | 8 |

 Return 1

Exercice 3 : Étant donné un tableau T contenant N entiers triés en ordre croissant. Écrire une fonction récursive qui permet d'ajouter une nouvelle valeur dans T.

Corrigé

Exercice 1 :

Mystere est une fonction récursive permettant de retourner le plus grand élément d'un tableau d'entiers.

Exercice 2 :

1- Version itérative :

```
int verif_iter(float T[],int N)
{
    int i,ct=0 ;
    for(i=0 ;i<N;i++)
        {
            if(T[i]>=10) ct++;
            if(ct >N/2) return 1;
        }
    return 0 ;
}
```

2- Version récursive :

```
int verif_recu(float T[],int N,int i,int ct)
{
    if(i==N) return 0;
    if(T[i] >=10)ct++ ;
    if(ct >N/2) return 1;
    return verif_recu(T,N,++i, ct)
}
```

L'appel récursif : pour un tableau T de taille N

```
int v= verif_recu( T, N,0,0) ;
```

Exercise 3 :

```
void Classer(int T[ ] , int v, int N)
{
    if (N<0)    T[N+1]=v ;

    else

        if ( T[N] < v )    T[N+1] = v ;

        else Classer( T,v,N-1) ;

}
```