

\*\*\*\*Exo1\*\*\*\*

```
1)
{{{
    Oui.
    Les instructions l2,l3 sont des instructions simples
    et se terminent.
    L'instruction l4 est une boucle. Une boucle se termine
    à deux conditions
        - La condition d'arrêt est vérifiée au bout d'un certain
          nombre d'itération
        - Chaque instruction dans le corps de la boucle se termine
          à chaque tour de boucle.
    Montrons que la boucle l5 se termine :
        la condition d'arrêt est  $j \geq n$  (for(..;j<n;..))
        or la valeur de n ne change pas après chaque tour
        de boucle et j augmente de 1 à chaque tour. Donc
        forcément au bout d'un moment  $j < n$  ne sera plus vérifié
        Le corps de la boucle ne contient qu'une instruction simple
        donc elle se termine.
    Montrons que la boucle l4 se termine.
        Même argument que pour l5 mais avec  $i < n$ ,
        ,l'instruction du corps de cette boucle est la boucle
        l5 qui, on vient de le montrer,se termine.

    Points importants (pour avoir des points)
        j (ou i) augmente à chaque tour et n est invariant
        et excepté les deux boucles il y
        uniquement des instructions simples
}}
```

2) la fonction renvoie nxn.  
(On veut une valeur de retour en fonction de l'entrée)

Justification (non demandée) :  
r est renvoyé et r vaut zero au départ.  
la seule instruction qui modifie r est r++ qui est situé  
dans une double boucle qui font chacune n itérations donc  
r++ sera exécutée nxn fois.  
donc en fin de fonction r vaut nxn.

3)  
Complexité en temps :  $\Theta(n^2)$  où n est la valeur de l'entrée  
Complexité en espace :  $\Theta(1)$  (la place occupée ne varie pas  
en fonction de n)

\*\*\*\*Exo2\*\*\*\*

$(\log n)^{345} \ll \text{racine}(n) \log n \ll n, \text{racine}(n)\text{racine}(n), (n^2+n)/(n+1)$   
 $\ll n^{45} \ll n^{(0,001)n}$   
 $n, \text{racine}(n)\text{racine}(n)$  et  $(n^2+n)/(n+1)$  sont dans la même classe  
de domination

\*\*\*\*Exo3\*\*\*\*

```
int index(cellule_t* liste,int val)
{
    int i;
    for(i=0;liste!=NULL;i++,liste=liste->suiv)
    {
        if(liste->val==val)
            return i;
    }
    return -1;
}
```

Exemple d'utilisation :

```

    cellule_t* maListe=NULL;
    // ajout d'élément dans la liste
    index(maListe,3);
***Exo4*****

void shuffle(cellule_t** listeD, cellule_t** listeS)
{
    cellule_t* temp;
    if(*listeS==NULL)
        return;
    if(*listeD==NULL)
    {
        *listeD=*listeS;
        *listeS=NULL;
        return;
    }
    temp=*listeS;
    *listeS=temp->suiv;
    temp->suiv=(*listeD)->suiv;
    (*listeD)->suiv=temp;
    shuffle(&temp->suiv, listeS);
}

```

Exemple d'utilisation :

```

    cellule_t* maListeA=NULL;
    // ajout d'élément dans la liste maListeA
    cellule_t* maListeB=NULL;
    // ajout d'élément dans la liste maListeB

```

```

    shuffle(&maListeA, &maListeB);
***Exo5*****
int renduMonnaie(int n, int* tab, int R)
{
    int a, s;
    if(R==0)
        return 0;
    if(n<1 || R<0)
        return -1;
    s=renduMonnaie(n-1, tab, R);
    a=renduMonnaie(n-1, tab, R-tab[n-1]);
    if(a==-1)
        return s;
    if(s==-1)
        return a+1;
    return s<a+1?s:a+1;
}

```

Exemple d'utilisation :

```

    int T[10]={1,4,5,1,5};
    int r=renduMonnaie(5, T, 7);

```