

Exercice N°1 : (4 pts)

Résultats affichés après l'exécution des deux programmes tester:

Programme tester (passage par référence) :	Programme tester (passage par valeur) :
2 6 3	2 6 3
6 3 2 (1pt)	2 3 2 (1pt)
3 2 6 (1pt)	2 2 2 (1pt)

Exercice N°2 : (6 pts)

Programme qui permet de créer une liste **L** à partir de **n** valeurs entières lues et de l'inverser :

```
program LLC; (0.5pt)
type ptListe=^tListe;
      tListe=record
          v:integer;
          suiv:ptListe;
      end;
var tete:ptListe; n:integer;
```

```
procedure construire(var vtete:ptListe; vn:integer); (2pts)
var courant, nouv:ptListe; i:integer;
begin if vn<1 then vtete:=nil
      else begin new(vtete); write('Valeur 1:'); readln(vtete^.v);
              vtete^.suiv:=nil; courant:=tete;
              for i:= 2 to vn do
                  begin new(nouv); write('Valeur ',i,':'); readln(nouv^.v); nouv^.suiv:=nil;
                          courant^.suiv:=nouv; courant:=nouv;
                  end;
              end;
end;
```

```
procedure inverser(var vtete:ptListe); (2.5pts)
var t, p, l: ptListe;
begin
    t:= Nil; p:=vtete; l:=vtete;
    while l<>Nil do begin p:=l; l:=l^.suiv; p^.suiv:=t; t:=p end;
    vtete:=t;
end;
```

BEGIN

```
(* Construire une LLC L à partir de n données lues*) (1pt)
write('Combien d"éléments voulez-vous saisir? '); readln(n);
construire(tete,n);
(* Inverser la LLC L *)
inverser(tete) (* Pour ceux qui n'ont pas utilisé des procédures la création est sur 2.5pts et
l'inversion est sur 3pts *)
```

END.

Exercice N° 3 : (5 pts)

Programme qui permet de remplir une tableau **A** de 10 elements entiers en utilisant une procédure **remplir_tab** et calculer le nombre d'apparition de chaque élément **A** en utilisant la fonction entière **occur**.

```
program tableau; (0.5pt)  
type TA = array [1..10] of integer;  
var A: TA;
```

```
procedure remplir_tab(var tab: TA); (1pt)  
var i: integer;  
begin For i:= 1 to 10 do  
    begin write('Donner la ',i,'eme valeur: ');  
          readln(tab[i]);  
    end;  
end;
```

```
function occur(tab: TA; val: integer): integer; (2,5pt)  
var i, j, cmpt: integer;  
begin cmpt:= 0;  
    for i:= 1 to 10 do  
        if (tab[i]= val) then cmpt:=cmpt+1;  
        occur := cmpt;  
    end;  
  
procedure occur_N_opt(tab:TA);  
var i: integer;  
begin for i:= 1 to 10 do  
    writeln ('Le nombre d"occurrence de: ', tab[i], ' est: ', occur(tab, tab[i]));  
end;
```

```
BEGIN (1pt)  
remplir_tab(A);  
occur_N_opt(A);  
END.
```

Exercice N°4 : (5 pts)

- 1) Ecrire une fonction Itérative qui prend comme paramètre un entier naturel n et calcule le terme u_n de la suite.

```
fonction suite_iterative( n :entier) :entier ;  
var u, i : entier ;  
debut  
  u ← 2 ;  
  i ← 0 ;  
  Tantque (i<n) faire u←3*u-1;  
                i ←i+1 ;  
                fftantque ;  
  suite ←u ;  
fin ;
```

(2pts)

- 2) Ecrire une autre fonction récursive qui prend comme paramètre un entier naturel n et calcule le terme u_n de la suite.

```
fonction suite_recursive(n :entier) : entier ;  
debut  
  si (n = 0) alors suite_recursive ← 2 ;  
            sinon suite_recursive ← 3 * suite(n-1) -1 ;  
  finsi ;  
fin ;
```

(3pts)