

Exercice 1 :

```
Program tri_2_critere;
uses wincrt ;
const n=10;
type tab=array[1..n] of string[20];
var t:tab;
    i,j,pos:integer;
    aux:string;
begin
  writeln('remplir t :');
  for i:=1 to n do
    begin
      write('ch = ');
      readln(t[i]);
    end;

  for i:=1 to n-1 do
    begin
      pos:=i;
      for j:=i+1 to n do
        if (length(t[j])<length(t[pos]))or((length(t[j])=length(t[pos]))and(t[j]<t[pos]))
          then pos:=j ;
      if i<>pos then begin
        aux:=t[i] ;
        t[i]:=t[pos];
        t[pos]:=aux;
      end;
    end;

  for i:=1 to n do
    begin
      writeln(t[i]);
    end;
end.
```

Exercice 2 :

```
Program tri_bulle_ordre_croissant_bidirectionnel;
uses wincrt ;
type tab=array[1..25] of integer;
var t:tab;
    n:integer;
(*****)
Procedure saisir(var n:integer);
begin
  repeat
    writeln('Donner un entier entre 5 et 25');
    readln(n);
  until n in [5..25];
end;
(*****)
Procedure remplir (var t:tab ; n:integer);
var i:integer;
begin
  randomize;
  for i:=1 to n do T[i]:=1+random(100);
end;
(*****)
Procedure trier (var T:tab ;n:integer);
var i,aux,debut,fin:integer;
    permut:boolean;
(*****)
```

```

begin
  debut:=1;fin:=n;
  repeat
    permut:=false;

    for i:=debut to fin-1 do
      if t[i]>t[i+1]
        then begin
          aux:=T[i];
          T[i]:=T[i+1];
          T[i+1]:=aux;
          permut:=true;
        end;
    fin:=fin-1;

    if permut=true
      then for i:=fin downto debut+1 do
        if t[i]<t[i-1]
          then begin
            aux:=T[i];
            T[i]:=T[i-1];
            T[i-1]:=aux;
            permut:=true;
          end;

        debut:=debut+1;
      until (permut=false) or (debut>=fin);
  end;
  (*****
Procedure afficher(T:tab ; n:integer);
  var i:integer;
  begin
    for i:=1 to n do write(T[i], ' ');
  end;
  (***** P.P. *****)
BEGIN
  saisir(n);
  remplir(t,n);
  writeln('Tableau avant le tri :');
  afficher(t,n);
  trier(t,n);
  writeln;
  writeln('Tableau après le tri :');
  afficher(t,n);
END.

```

Exercice 3 :

```

Program tri_2_critere;
uses wincrt;
type tch=array[1..10] of string[20];
    tc=array[1..10] of char;
var n:integer;
    t:tch; c:tc;
(*****
Procedure saisie(var n:integer;var t:tch;var c:tc);
Var i:integer;
begin
  write('N = ');readln(n);
  writeln('remplir les tableaux T et C :');
  for i:=1 to n do
    begin
      write('nom = '); readln(t[i]);
      write('couleur = '); readln(c[i]);

```

```

    end;
end;
(*****)
Procedure tri (n:integer;var t:tch;var c:tc);
Var i:integer;
    permut:boolean;
    aux:string;
    tmp:char;
begin
    repeat
        permut:=false;
        for i:=1 to n-1 do
            if (c[i]>c[i+1])or((c[i]=c[i+1])and(t[i]>t[i+1]))
                then begin
                    aux:=t[i] ;
                    t[i]:=t[i+1];
                    t[i+1]:=aux;
                    tmp:=c[i] ;
                    c[i]:=c[i+1];
                    c[i+1]:=tmp;
                    permut:=true
                end;
            n:=n-1
        until (permut=false) or (n=1);
end;
(*****)
Procedure affiche (n:integer;t:tch;c:tc);
var i:integer;
begin
    for i:=1 to n do
        writeln(t[i], ' ',c[i]);
end;
(*****p.p*****)
Begin
    saisie(n,t,c);
    tri(n,t,c);
    affiche(n,t,c);
End.

```

Exercice 4 :

```

Program Exercice4;
uses wincrt;
type tab1=array [1..20] of integer;
    tab2=array [1..40] of integer;
var v1,v2:tab1;
    v3:tab2;
    n,m,c:integer;
(*****)
Procedure lecture (var taille:integer);
begin
    repeat
        readln(taille);
    until taille in [2..20];
end;
(*****)
Procedure remplir(var t:tab1; taille:integer);
var i:integer;
begin
    for i:= 1 to taille do readln(t[i]);
end;
(*****)
Procedure trier (taille:integer;var t:tab1);
Var i,tmp, min,j:integer;

```

```

begin
  for i:=1 to taille-1 do
    begin
      min:=i;
      for j:=i+1 to taille do
        if t[j]<t[min] then min:=j;
      if i<>min then begin
        tmp:=t[i];
        t[i]:=t[min];
        t[min]:=tmp;
      end;
    end;
  end;
end;
(*****
Procedure fusionner(v1,v2:tab1; var v3:tab2; n,m:integer; var c:integer);
var i,c1,c2:integer;
begin
  c1:=1; c2:=1; c:=0;
  repeat
    c:=c+1;
    if v1[c1]<v2[c2]
      then begin
        v3[c]:=v1[c1];
        c1:=c1+1;
      end
    else begin
      v3[c]:=v2[c2];
      c2:=c2+1;
    end
  until (c1>n) or (c2>m);
  if c1>n then
    for i:=c2 to m do
      begin
        c:=c+1;
        v3[c]:=v2[i];
      end
    else
      for i:=c1 to n do
        begin
          c:=c+1;
          v3[c]:=v1[i];
        end;
    end;
end;
(*****
Procedure afficher(t:tab2; taille:integer);
var i:integer;
begin
  writeln('Tableau fusion :');
  for i:= 1 to taille do
    write(t[i]:4);
  end;
(*****p*****
begin
  write('Taille V1 : ');lecture(n);
  write('Taille V2 : ');lecture(m);
  writeln('Remplir V1 :');remplir(v1,n);
  writeln('Remplir V2 :');remplir(v2,m);
  trier(n,v1);
  trier(m,v2);
  fusionner(v1,v2,v3,n,m,c);
  afficher(v3,c);
end.

```

Exercice 5 :

```
Program temps_tris;
uses wincrt,windos;
type tab=array[1..1000] of real;
var t,t1,t2:tab;
    n:integer;
    hi1,hi2,mi1,mi2,si1,si2,csi1,csi2,hs1,hs2,
    ms1,ms2,ss1,ss2,css1,css2,ts1,ti1:word;
(***** lecture et duplication *****)
Procedure lecture_duplic(var n:integer;var t,t1,t2:tab);
var i:integer;
begin
  Writeln('Saisir un entier pour la taille des tableaux'); readln(n);
  randomize;
  for i:=1 to n do
    begin
      t[i]:=100*random; { réel aléatoire entre [0,100[ }
      t1[i]:=t[i];
      t2[i]:=t[i];
    end;
end;
(***** TRI SELECTION *****)
Procedure tri1 (n:integer;var t1:tab);
var pm,i:integer;
(*****)
Function posmin(d,f:integer;t:tab):integer;
var i,pmin,j:integer;
begin
  pmin:=d;
  for j:=d+1 to f do
    if t[j] < t[pmin] then pmin:=j;
  posmin:=pmin;
end;
(*****)
Procedure permut (var x,y:real);
var aux:real;
begin
  aux:=x;
  x:=y;
  y:=aux;
end;
(*****)
begin
  for i:=1 to n-1 do
    begin
      pm:=posmin(i,n,t1);
      if pm<>i then permut(t1[pm],t1[i]);
    end;
end;
(***** TRI INSERTION *****)
Procedure tri2 (n:integer;var t2:tab);
Var j,i:integer;
    tmp:real;
(*****)
procedure decaler (var t2:tab;var j:integer;i:integer);
begin
  j:=i;
  WHILE (j>1)and(t2[j-1]>tmp) DO
    Begin
      t2[j]:=t2[j-1];
      j:=j-1;
    End ;
end;
```

```

(*****)
Begin
  for i:=2 to n do
    if t2[i]<t2[i-1]
      then Begin
        tmp:=t2[i];
        Decaler(t2,j,i);
        t2[j]:=tmp;
      End ;
  End;
(***** Affichage *****)
Procedure affiche(n:integer;t:tab);
var i:integer;
begin
  for i:=1 to n do write(t[i]:2:2, ' ');
end;
(***** Programme principal *****)
BEGIN
  lecture_duplic(n,t,t1,t2);
  gettime(hs1,ms1,ss1,css1);
  tri1(n,t1);
  gettime(hs2,ms2,ss2,css2);
  ts1:=(hs2-hs1)*3600*100+(ms2-ms1)*60*100+(ss2-ss1)*100+css2-css1;
  gettime(hi1,mi1,si1,csi1);
  tri2(n,t2);
  gettime(hi2,mi2,si2,csi2);
  ti1:=(hi2-hi1)*3600*100+(mi2-mi1)*60*100+(si2-si1)*100+csi2-csi1;
  affiche(n,t1);
  readln;
  affiche(n,t2);
  readln;
  writeln('tri selection  : ',ts1, ' Centième de seconde');
  writeln('tri insertion  : ',ti1, ' Centième de seconde');
END.

```

Exercice 6 :

```

Program tri_rang;
uses wincrt,ecran;
type tab=array[1..25] of integer;
var t,r,s:tab;
    n:integer;
(*****)
Procedure saisir(var n:integer);
begin
  repeat
    writeln('Donner un entier entre 5 et 25');
    readln(n);
  until n in [5..25];
end;
(*****)
Procedure remplir (n:integer ; var T:tab);
var i:integer;
begin
  randomize;
  for i:=1 to n do T[i]:=random(101); {valeur aléatoire entre [0..100]}
end;
(*****)
Procedure RANG (N:Integer ; T:Tab;Var R,S:Tab);
Var i,j : integer;
Begin
  For l:=1 To N Do S[l]:=1;
  For l:=1 To N-1 Do
    For j:=l+1 To N Do

```

```

    If T[I]>T[J]
      Then S[I]:=S[I]+1
      Else S[J]:=S[J]+1;
    For I:=1 To N Do R[S[I]]:=I;
End;
(*****
Procedure afficher(n:integer ; t:tab);
var i:integer;
begin
  for i:=1 to n do write(T[i]:4);
end;
(***** P.P. *****)
Begin
  Saisir(N);
  Remplir(N,T);
  Writeln('Tableau T :');
  Afficher(N,T);
  Rang(N,T,R,S);
  Writeln;
  Writeln('Tableau Trié Suivant L"indice :');
  Afficher(N,R);
End.

```

Fenni Salah

Exercice 3:

```

Program Exercice3;
uses wincrt;
type tab=array[1..100] of char;
var t:tab;
    max_suite:string;
    n:integer;
(*****)
Procedure saisie(var n :integer ; var t:tab);
Var i :integer ;
Begin
  Repeat
    Write('N = ');
    Readln(n);
  Until n in [10..100];
  for i:=1 to n do
    repeat
      readln(t[i]);
    until t[i]<>";
end;
(*****)
Function Long_suite(n:integer;t:tab) :string ;
Var  max, suite:string;
    i:integer;
begin
  max:=t[1];
  suite:=t[1];
  for i:=2 to n do
    if t[i]=t[i-1]
      then suite:=suite+t[i]
      else begin
        if length(suite)>length(max) then max:=suite;
        suite:=t[i];
        end;
  long_suite:=max;
end;
(*****p.p*****)
Begin
  Saisie(n,t);
  Max_suite:=long_suite(n,t);
  writeln (max_suite[1], ' ', length(max_suite));
End.

```

Exercice 4:

```

Program Exercice4 ;
Uses Wincrt;
type Tab = Array [1..50] of Integer;
Var  T,R : Tab;
    n,j : Integer;
(*****)
Procedure Saisie (Var T : Tab ; Var n : Integer);
Var i : Integer;
Begin
  Repeat
    Writeln ('Donner un entier'); Readln (n);
  Until n in [2..50];
  writeln('Saisir les éléments du tableau');
  FOR i:=1 To n Do
    Repeat
      Readln (T[i])
    Until T[i] >= 0;
End;

```

```

(*****)
Procedure compactage (n:integer;T:Tab;var R:tab;var j:integer);
var i,k,m:integer;
Begin
  i:=0; j:=0;
  repeat
    i:=i+1; j:=j+1;
    if t[i]<>0
      then r[j]:=t[i]
      else begin
        m:=0; k:=i;
        repeat
          m:=m+1;
          k:=k+1;
        until (t[k]<>0) or (k>n);
        R[j]:=m;
        i:=k-1;
        end;
    until i=n;
End;
(*****)
Procedure affiche (n:integer ; t:tab);
var i:integer;
begin
  for i:=1 to n do write(t[i], ' ');
  writeln;
end;
(*****p.p*****)
Begin
  Saisie (T, N);
  compactage (n,t,r,j);
  writeln('Le tableau initial :');
  affiche(n,t);
  writeln('Après compactage :');
  affiche(j,r);
End.

```

Exercice 5:

```

Program Exercice5 ;
Uses Wincrt;
type Tab = Array [1..50] of Integer;
Var  T : Tab;
    N, V : Integer;
(*****)
Procedure Saisie (Var T : Tab ; Var n, v : Integer);
Var i : Integer;
Begin
  Repeat
    Writeln ('Donner un entier'); Readln (n);
  Until n in [2..49];
  Writeln ('Saisir les éléments de T en ordre croissant');
  Readln (T[1]);
  FOR i:=2 To n Do
    Repeat
      Readln (T[i])
    Until T[i] >= T[i-1];
    Writeln ('Donner la valeur à insérer'); Readln (v);
End;
(*****)
Procedure insertion (v,n:integer ; var T:Tab) ;
var i,p:integer;
(*****)

```



```

Function recherche_pos (v,n:Integer ; T:Tab):Integer;
Var d, g, m: Integer;
Begin
  g:=1 ; d:=n ;
  Repeat
    m:=(g+d) div 2 ;
    IF V>T[m]
      Then g:=m+1
      Else d:=m-1;
  Until g>d;
  recherche_pos:=g;
End;
(*****)
Begin
  p:=recherche_pos(v,n,t);
  For i:=n+1 downto p Do T[i]:=T[i-1];

```

```

  T[p]:=v;
End;
(*****)
Procedure affiche(n:integer ; t:tab);
var i:integer;
begin
  writeln('Le nouveau tableau :');
  for i:=1 to n+1 do
    write(t[i], ' ');
end;
(*****p.p*****)
Begin
  Saisie (T, N, V);
  Insertion (v,n,t);
  Affiche(n,t);
End.

```

Fenni Salah