Corrigés

Exercices
En
Turbo Pascal

Prof : FENNI Salah

Lycée Ibn Rochd – La Chebba

LES STRUCTURES SIMPLES

Exercice 4

```
[ V ] Readln (A) ;
[ F ] Readln (45) ;
[ F ] Readln (Test) ;
[ V ] Writeln ('A = ', A) ;
[ V ] Writeln (5 mod 7 div 2) ;

[ F ] Readln ('A') ;
[ F ] Readln ('A =', A);
[ F ] Readln (Jour) ;
[ F ] Writeln (A:6:2) ;
[ F ] Writeln (Nom[2]) ;

[ F ] Readln (A+B) ;
[ F ] Readln (Nom) ;
[ F ] Writeln (Jour) ;
[ V ] Writeln (45) ;
[ V ] Writeln (A+B, Nom, Test) ;
```

Exercice 5

```
0) Début  Sortie_Inverse
1) Ecrire ("A = "), Lire (A)
2) Ecrire ("B = "), Lire (B)
3) Ecrire ("C = "), Lire(C)
4) Ecrire (C, "    ", B, "    ", A)
5) Fin Sortie_Inverse
```

Exercice 6

```
0) Début  Cylindre
1) Ecrire ("Donner le rayon : "), Lire(R)
2) Ecrire ('Donner la hauteur : "), Lire (H)
2) V ← PI*R*R*H
3) Ecrire ("Volume = ", V)
4) Fin Cylindre
```

Exercice 7

```
0) Début  Surface_rectangle
1) Ecrire ("Largeur = "), Lire (la)
2) Ecrire ("Longueur = "), Lire (lo)
3) S← la * lo
4) Ecrire ("La surface du rectangle dont la longueur mesure ", lo, " m et la largeur mesure ", la," m, a une surface égale à ", s," mètres carrés.")
5) Fin Surface_rectangle
```

Exercice 8

```
0) Début  Piscine
1) Ecrire ("Donner les dimensions de la piscine"), Lire (LO, LA, PR)
2) V ← LO*LA*PR
3) EAU ← V*1000
4) Ecrire ("Le volume = ", V)
5) Ecrire ("Quantité d'eau = ", EAU, " litres")
6) Fin Piscine
```

Exercice 9

```
0) Début  Aire_Trapeze
1) Ecrire ("Donner les dimensions du trapèze "), Lire (H, B1, B2)
2) S← H*(B1 + B2)
3) Ecrire ("La surface = ", S)
4) Fin Aire_trapeze
```

Exercice 10

```
0) Début  Permut
1) Lire (A, B)
2) AUX ← A
3) A ← B
4) B ← AUX
5) Ecrire ("La nouvelle valeur de A est : ", A)
6) Ecrire ("La nouvelle valeur de B est : ", B)
7) Fin Permut
```

Exercice 11

```
0) Début  Permut_Circulaire
1) Lire (A, B, C)
2) AUX ← A
3) A ← C
4) C ← B
5) B ← AUX
6) Ecrire (A, "    ", B, "    ", C)
7) Fin Permut_Circulaire
```

Exercice 12

```
0) Début  Permut
1) Lire (X, Y)
2) X ← X+Y
3) Y ← X-Y
4) X ← X-Y
5) Ecrire (X, "    ", Y)
6) Fin Permut
```

Exercice 13

```
0) Début  Division
1) Ecrire ("A = "), Lire (A)
2) Ecrire ("B = "), Lire (B)
3) Q ← A div B
4) R ← A mod B
5) Ecrire ("Le quotient est ", q, " et le reste est ", r)
6) Fin Division
```

Exercice 14

```
0) Début  Temperature
1) Ecrire ("Donner une température en °C : "), Lire (D)
2) F ← 9/5 * D + 32
3) Ecrire (D, " °C = ", F, " Fahrenheit")
4) Fin Temperature
```

Exercice 15

```
0) Début  Mile_marin
1) Ecrire ("Donner le nombre de Km : "), Lire (km)
2) Mm ← km/1.852
3) Ecrire (km," km = ", mm," miles marins")
4) Fin mile_marin
```

Exercice 16

```
0) Début  Conversion
1) Ecrire ("Nombres de bits = "), Lire (bit)
2) Octet ← bit/8
3) Kilo ← octet/1024
4) Mega ← kilo/1024
5) Giga ← mega/1024
6) Ecrire (octet, kilo, mega, giga)
7) Fin conversion
```

Exercice 17

```
0) Début   Temps
1) Ecrire ("Donner une durée en secondes "), Lire (T)
2) H ← T div 3600
3) M ← (T div 60) mod 60
4) S ← T mod 60
5) Ecrire (H, " : ", M, " : ", S)
6) Fin Temps
```

Exercice 19

```
0) Début  Interet_Simple
1) Ecrire ("Donner la somme initiale : "), Lire (SOM)
2) Ecrire ("Donner le taux d'intérêt : "), Lire (TAUX)
3) INTERET ← (SOM * TAUX/100) * 5
4) VA ← SOM + INTERET
5) Ecrire ("Après 5 ans la somme sera = ", VA)
6) Fin Interet_Simple
```

Exercice 20

```
0) Début  Consommation
1) Ecrire ("Nombre de Km parcouru : "), Lire(Y)
2) Ecrire ("Nombre de litres d'essence : "), Lire (L)
3) TAUX ← (L*100)/Y
4) Ecrire ("Taux de consommation est =  ", TAUX, "  %")
5) Fin Consommation
```

Exercice 21

```
0) Début  Résistance
1) Ecrire ("Donner les trois résistances : "), Lire (R1, R2, R3)
2) Rser ← R1 + R2 + R3
3) Rpar ← 1/ (1/R1 + 1/R2 + 1/R3)
4) Ecrire ("Résistance résultante sérielle : ", Rser)
5) Ecrire ("Résistance résultante parallèle : ", Rpar)
6) Fin Résistance
```

Exercice 18

```
0) Début  Futur
1) Ecrire ("Donner un verbe du 1er groupe : "), Lire (verbe)
2) Ecrire ("Je ", verbe, "ai")
3) Ecrire ("Tu ", verbe, "as")
4) Ecrire ("Il ou elle ", verbe, "a")
5) Ecrire ("Nous ", verbe, "ons")
6) Ecrire ("Vous ", verbe, "ez")
7) Ecrire ("Ils ou elles ", verbe, "ont")
8) Fin Futur
```

Exercice 22

```
0) Début  NBR_3
1) Ecrire ("Saisir un entier formé de 3 chiffres non nuls : "), Lire (N)
2) c ← n  div 100
3) d ← n  mod  100  div  10
4) u ← n  mod 10
5) r1 ← c*100+u*10+d
6) r2 ← u*100+d*10+c
7) r3 ← u*100+c*10+d
8) r4 ← d*100+c*10+u
9) r5 ← d*100+u*10+c
10) Ecrire (N, r1, r2, r3, r4, r5)
11) Fin NBR_3
```

Exercice 23

```
0) Début  Sup_Inf
1) Ecrire ("A = "), Lire (A)
2) Ecrire ("B = "), Lire (B)
3) SUP ← (A + B + abs (A - B))  div  2
4) INF ← (A + B - abs (A - B))  div  2
5) Ecrire ("Valeur sup =  ", SUP, "   Valeur inf =  ", INF)
6) Fin Sup_Inf
```

Exercice 24

```
0) Début IMAGE
1) Ecrire ("Entrer la largeur de l'image: "), Lire(l)
2) Ecrire ("Entrer la hauteur de l'image: "), Lire (h)
3) Ecrire ("Entrer la résolution de l'image: "), Lire(r)
4) Ecrire ("Entrer le codage de l'image: "), Lire(c)
5) n ← l*r*h*r
6) p ← (n*c) / (1024*1024)
8) Ecrire (n,p)
9) Fin IMAGE
```

Exercice 25

```
Program   Date;
Uses   Wincrt, Windos;
Var     A, M, J : Word;
Begin
    Write ('Entrez L''année : ');Readln (A);
    Write ('Entrez Le Mois : '); Readln (M);
    Write ('Entrez Le Jour : '); Readln (J);
    Setdate (A,M,J);
End.
```

Exercice 26

```
Program Distance ;
Uses Wincrt ;
Var
   Xa, Ya, Xb, Yb :  Integer ;
   Dist :  Real ;
Begin
   Writeln ('Entrez Les Coordonnées Du Point A : Xa,Ya ');
   Readln (Xa,Ya) ;
   Writeln ('Entrez Les Coordonnées Du Point B : Xb,Yb ');
   Readln (Xb, Yb);
   Dist := Sqrt (Sqr(Xa-Xb)+Sqr(Ya-Yb));
   Writeln ('La Distance Entre A Et B Est ', Dist :2 :2);
End.
```

Exercice 27

```
Program Prix_TTC ;
Uses Wincrt ;
Var
   Pnet, Tva :  Integer;
   Pttc :  Real;
Begin
   Writeln('Entrez Le Prix Net De L''article : ');
   Readln(Pnet);
   Writeln('Entrez Le Taux De La Tva (En %) : ' );
   Readln(Tva);
   Pttc := Pnet+Pnet*Tva/100;
   Writeln('Le Prix Ttc Est ', Pttc :2 :2);
End.
```

Exercice 27

```
Program Prix_NET ;
Uses Wincrt ;
Var
   Tva :  Integer;
   Pttc,Pnet :  Real;
Begin
   Writeln('Entrez Le Prix Ttc De L''article : ');
   Readln(Pttc);
   Writeln('Entrez Le Taux De La Tva (En %) : ' );
   Readln(Tva);
   Pnet := Pttc*100/(100+Tva);
   Writeln('Le Prix Net Est ', Pnet :2 :2);
End.
```

LES STRUCTURES CONDITIONNELLES

Exercice 1

```
Program   Min2 ;
Uses   Wincrt ;
Var     a, b, min : Integer ;
Begin
  Writeln ('Saisir deux entiers : ') ;
  Readln (a, b) ;
  IF  a<b   Then  min := a
            Else   min := b ;
  Writeln ('La plus petite valeur est : ', min) ;
End.
```

Exercice 2

```
Program   Max3 ;
Uses    Wincrt ;
Var      a, b, c, maxi : Integer ;
Begin
  Writeln ('Saisir trois entiers : ') ;
  Readln (a, b, c) ;
  maxi := a ;
  IF  b>maxi   Then  maxi := b ;
  IF  c>maxi   Then  maxi := c ;
  Writeln ('La plus grande valeur est : ', maxi) ;
End.
```

Exercice 3

```
Program   Racine ;
Uses   Wincrt ;
Var     x : Real ;
Begin
   Write ('Saisir un réel  ') ; Readln (x) ;
   IF  x >= 0
     Then  Writeln ('Racine carrée ', x,' = ', sqrt(x))
     Else   Writeln ('Donnée incorrecte') ;
End.
```

Exercice 4

```
Program    Aire_Triangle ;
Uses     Wincrt ;
Var       a, b, c, S, P : Real;
 Begin
     Writeln ('Donner 3 réels positifs :'); Readln (a, b, c);
     IF (a+b=c) Or (a+c=b) Or (b+c=a)
         Then  Writeln ('Il ne s''agit pas d''un triangle')
         Else   Begin
                  P := (a+b+c)/2;
                  S := sqrt (P*(P-a)*(P-b)*(P-c));
                End;
     Writeln ('Aire de triangle = ', S:4:2);
End.
```

Exercice 6

```
Program   Parite ;
Uses    Wincrt ;
Var      N : Integer ;
Begin
   Writeln ('Donner un entier ') ; Readln (n) ;
   IF  n  mod 2 = 0
       Then  Writeln (n,' est pair')
       Else   Writeln (n,' est impair') ;
End.
```

Exercice 5

```
Program   Abs_diff;
Uses   Wincrt;
Var     a, b, z : Integer;
Begin
   Write ('Donner deux entiers : '); Readln (a, b);
   IF  (a-b) < 0   Then  z:=b-a
                   Else   z:=a-b;
   Writeln ('valeur absolue de a-b = ', z);
End.
```

Exercice 7

```
Program   Chez_la_fourmi;
Uses     Wincrt;
label    1,2 ;
Var       na, nb, err : Integer;
         a, b, z : Char;
Begin
  1: Writeln ('Nombre de doigts montrés par le joueur A');
  a:=readkey;
  Writeln ('Nombre de doigts montrés par le joueur B');
  b:=readkey;
  VAL (a, na, err);
  VAL (b, nb, err);
  IF (na+nb) mod 2 =0
     Then Writeln ('Le joueur A gagne.')
      Else Writeln ('Le joueur B gagne.');
  Writeln ('Voulez vous jouer encore ? (o/n)'); Readln (z);
  IF z='n' Then goto 2 Else goto 1 ;
2:End.
```

Exercice 9

```
Program   Sup_Inf;
Uses    Wincrt ;
Var      a, b : Integer ;
         sie : String ;
Begin
  Writeln ('Saisir deux entiers : ') ;
  Readln (a, b) ;
  IF  a>b
      Then  sie := ' est supérieur à '
      Else  IF a<b
            Then   sie := ' est inférieur à '
            Else    sie := ' est égal à ' ;
  Writeln (a, sie, b) ;
End.
```

Exercice 8

```
Program    Invite ;
Uses    Wincrt ;
Var     titre, foulen, term1, term2 : String ;
Begin
   Write ('Titre = ') ; Readln (titre) ;
   Write ('Votre nom = ') ; Readln (foulen) ;
   IF   titre = 'Mr'
        Then  Begin
                 term1 := 'e' ;
                 term2 := '' ;
               End
         Else IF (titre = 'Mlle') Or (titre='Mme')
              Then Begin
                      term1 := 'a' ;
                      term2 := 'e' ;
                    End ;
    Writeln (titre, ' ', foulen, ', soyez l', term1,
                                          ' bienvenu', term2) ;
End.
```

Exercice 10

```
Program   Ordre ;
Uses    Wincrt ;
Var     e1, e2, petit, grand : Integer ;
Begin
  Writeln ('Saisir deux entiers : ') ; Readln (e1, e2) ;
  petit := e1 ;
  grand := e2 ;
  IF  e1>e2    Then  Begin
                      petit := e2 ;
                      grand := e1 ;
                   End;
  Writeln (petit, grand:3) ;
End.
```

Exercice 12

```
Program   Equa_1d ;
Uses    Wincrt ;
Var      a, b : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  IF  a <> 0
     Then  Writeln ('x = ', -b/a)
      Else  IF  b = 0
                Then  Writeln ('IR')
                 Else  Writeln ('{}') ;
End.
```

Exercice 11

```
Program   Tri;
Uses     Wincrt;
Var      a, b, c, aux: Integer;
Begin
   Write ('a = ') ; Readln (a) ;
   Write ('b = ') ; Readln (b) ;
   Write ('c = ') ; Readln (c) ;
   IF  a>b  Then   begin
                     aux:=a;
                     a:=b;
                     b:=aux;
                   end;
   IF  b>c  Then   begin
                     aux:=b;
                     b:=c;
                     c:=aux;
                   end;
   IF  a>b  Then   begin
                     aux:=a;
                     a:=b;
                     b:=aux;
                   end;
   Writeln (a, b:4, c:4);
End.
```

Exercice 13

```
Program    Equa_2d ;
Uses     Wincrt ;
Var      a, b, c, delta : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  Write ('c = ') ; Readln (c) ;
  IF  a = 0          {équation 1er degré}
     Then  IF  b = 0
          Then  IF  c = 0
                  Then  Writeln ('IR')
                  Else   Writeln ('{}')
          Else  Writeln ('x = ', -c/b)
     Else  delta := sqr (b) - 4*a*c ;
  IF  delta = 0       {solution réelle double}
     Then Writeln ('x1=x2= ', -b/ (2*a))
      Else  IF  delta > 0        {deux solutions réelles}
        Then  Begin
              Writeln ('x1= ', (-b-sqrt (delta))/ (2*a)) ;
              Writeln ('x2= ', (-b+sqrt (delta))/ (2*a)) ;
             End
        Else  Writeln ('Deux solutions complexes') ;
 End.
```

Exercice 14

```
Program   Inequation ;
Uses   Wincrt ;
Var      a, b : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  IF  a>0
    Then  Writeln ('x < ', -b/a)
    Else  IF  a<0
           Then Writeln ('x > ', -b/a)
           Else  IF  b<0
                   Then  Writeln ('IR')
                   Else   Writeln ('Impossible') ;
End.
```

Exercice 15

```
Program   Touche ;
Uses   Wincrt;
Var     c : Char ;
        nature : String;
Begin
   Writeln ('Taper sur une touche'); Readln (c);
   Case  c  of
    'a'..'z','A'..'Z' : IF  UPCASE(c) in ['A','E','I','U','O','Y']
                          Then nature := 'Voyelle'
                           Else  nature := 'Consonne';
       '0'..'9'   : nature := 'Chiffre';
        Else  nature := 'Symbole';
   End;
   Writeln (nature);
End.
```

Exercice 16

```
Program   Calculette ;
Uses     Wincrt ;
Var      a, b : Real ;
         op : Char ;
Begin
  Readln (a) ; Readln (op) ; Readln (b) ;
  Case   op   of
      '+' : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a+b:3:2 ) ;
      '-'  : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a-b:3:2) ;
      '*'  : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a*b:3:2) ;
       '/' : IF  b = 0
            Then  Writeln ('impossible')
             Else  Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a/b:3:2);
      Else  Writeln ('Opérateur incorrect');
  End ;
End.
```

Exercice 17

```
Program   Bulletin ;
Uses   Wincrt ;
Var    moy : Real ;      me, dec : String ;
Begin
  Write ('Donner la moyenne annuelle : ') ; Readln (moy) ;
  IF  moy < 10
     Then   Begin
               dec := 'Redouble' ;
               me := '' ;
            End
     Else   Begin
              dec := 'Admis' ;
              IF  moy <  12
                 Then   me := 'Passable'
                 Else  IF  moy <  14
                    Then  me := 'Assez bien'
                    Else IF  moy <  16
                       Then  me := 'Bien'
                       Else  IF  moy <  18
                          Then  me := 'Très bien'
                          Else   me := 'Excellent' ;
            End ;
    Writeln ('Moyenne = ', moy,' Décision = ', dec,
               ' Mention = ', me) ;
End.
```

Exercice 18

```
Program   nbr_jours_mois;
Uses   wincrt;
Var    nbj, mm, an : integer;

Begin
    Write ('N° du mois : ') ; readln (mm) ;

    nbj :=31;

    if mm in [4,6,9,11]
          then  nbj :=30
          else begin
                  write ('Année : '); readln (an);
                  nbj := 28;
                  if (an mod 4=0)and((an mod 100<>0)or(an mod 400=0))
                            then  nbj := 29;
                 end;

     writeln ('Le nombre de jours du mois saisi est : ', nbj);
End.
```

Exercice 20

```
Program   Anciennete ;
Uses     Wincrt ;
Var      ji, mi, ai, jf, mf, af, jj, mm, aa : Integer ;
Begin
     Write ('Donner la date initiale : ') ;
     Readln (ji, mi, ai) ;
     Write ('Donner la date finale : ') ;
     Readln (jf, mf, af) ;
     IF  ji > jf
          Then  Begin
                     jf := jf + 30 ;
                     mf := mf - 1 ;
                  End ;
     IF  mi > mf
          Then   Begin
                       mf := mf + 12 ;
                       af := af - 1 ;
                   End ;
     jj := jf - ji ;
     mm := mf - mi ;
     aa := af - ai ;
     Writeln (aa, ' années ', mm, ' mois ', jj, ' jours') ;
End.
```

Exercice 19

```
Program   Date ;
Uses    Wincrt ;
Var      mm, err : Integer ;
          date, jj, aa, mois : String [10] ;
Begin
  Writeln ('Saisir une date sous la forme jj/mm/aaaa') ;
  Readln (date) ;
  jj := COPY (date, 1, 2) ;
  aa := COPY (date, 7, 4) ;
  VAL (COPY (date, 4, 2), mm, err) ;
  Case    mm   of
       1 : mois := 'Janvier' ;
       2 : mois := 'Février' ;
       3 : mois := 'Mars' ;
       4 : mois := 'Avril' ;
       5 : mois := 'Mai' ;
       6 : mois := 'Juin' ;
       7 : mois := 'Juillet' ;
       8 : mois := 'Août' ;
       9 : mois := 'Septembre' ;
       10 : mois := 'Octobre' ;
       11 : mois := 'Novembre' ;
       12 : mois := 'Décembre' ;
       End ;
  Writeln (jj, mois:2, aa:2) ;
End.
```

Exercice 21

```
Program  Lendemain;
Uses   wincrt ;
Var    j,m,a,ms,js,as:integer;
        biss,dj:boolean;
begin
   writeln ('donner la date d''aujourd''hui : ');
   readln (j,m,a);
   if  ( a mod 100)=0
         then  biss:=(a mod 400)=0
         else  biss:=(a mod 4)=0;
   case   m   of
         1,3,7,8,10,12 : dj:=(j=31);
         4,6,9,11 : dj:=(j=30);
         2 : if  biss  then dj:=(j=29)
                    else  dj:=(j=28);
   end;
if  dj  then  begin
                  js:=1;
                  if  m=12 then begin
                                         ms:=1;
                                         as:=a+1;
                                      end
                              else begin
                                        ms:=m+1;
                                        as:=a;
                                     end;
              end
      else begin
                js:=j+1;
                ms:=m;
                as:=a;
             end;
writeln ('le jour suivant est :',js,'/',ms,'/',as);
end.
```

Exercice 22

```
Program   Toute_Lettre;
Uses    Wincrt;
Const
    Chiffres : Array[0..19] of String =('','un','deux','trois','quatre','cinq','six','sept','huit','neuf','dix',
                                        'onze','douze','treize','quatorze','quinze','seize','dix-sept','dix-huit','dix-neuf');
    Dizaines : Array[2..9] of String = ('vingt','trente','quarante','cinquante','soixante','','quatre-vingt','');
Var     n : Integer;
        result : String;
Begin
  Writeln ('Donner un entier entre 0 et 99');  Readln (n);
  Case   n  of
     0..19 : result := Chiffres [n];
     20..69,80..89: IF ((n mod 10 = 1) and (n<>81))
                     Then    result := Dizaines [n div 10] + ' et ' + Chiffres [n mod 10]
                     Else    result := Dizaines [n div 10] + ' ' + Chiffres [n mod 10];
     70..79,90..99: IF (n = 71)
                    Then    result := Dizaines [n div 10 -1] + ' et ' + Chiffres [n mod 10 + 10]
                    Else    result := Dizaines [n div 10 -1] + ' ' + Chiffres [n mod 10 + 10];
  End;

 IF n=0   Then  Writeln ('zéro')
          Else  Writeln (result);
End.
```

Exercice 23

```
Program   jour_semaine;
Uses    wincrt;
Var    day, month, year, dayyear, daymonth, weekday, cm:integer;
       jj:string;
Begin
  writeln ('Donner le jour'); readln (day);
  writeln ('Donner le mois'); readln (month);
  writeln ('Donner l''année'); readln (year);
  dayyear:=(year-1)*365 + ((year-1) div 4);
  daymonth:=0;
  for  cm:=1  to  (month-1)  do
    case   cm   of
     1, 3, 5, 7, 8, 10, 12 : daymonth:=daymonth+31;
            4, 6, 9, 11 : daymonth:=daymonth+30;
                   2 : if (year mod 4=0) and ((year mod 100<>0) or (year mod 400 =0))
                          then  daymonth:=daymonth+29
                          else   daymonth:=daymonth+28;
     end;
  weekday:=(dayyear+daymonth+day) mod 7;
  case  weekday  of
     0:jj:='Dimanche';
     1:jj:='Lundi';
     2:jj:='Mardi';
     3:jj:='Mercredi';
     4:jj:='Jeudi';
     5:jj:='Vendredi';
     6:jj:='Samedi';
  end;
  writeln ('Le jour correspondant est ', jj);
End.
```

Exercice 24

```
Program Signe_produit;
Uses Wincrt;
Var
  A,B:  Integer;
Begin
  Writeln('Introduisez deux nombres entiers :'); Readln(A,B);
  If (A=0) Or (B=0)
     Then Writeln ('Le produit ',A,' * ',B,' est nul')
     Else  If ((A>0) And (B>0)) Or ((A<0) And (B<0))
        Then Writeln ('Le signe du produit ',A,' * ',B,' est positif')
        Else  Writeln ('Le signe du produit ',A,' * ',B,' est négatif') ;
End.
```

Exercice 25

```
Program Signe_somme;
Uses Wincrt;
Var
  A,B:  Integer;
Begin
  Writeln('Introduisez deux nombres entiers :');
  Readln(A,B);
  If (A=0) And (B=0)
    Then Writeln ('La somme ',A,' + ',B,' est zéro')
  Else If ((A>0) And (B>0))
        Or ((A<0) And (B>0)And(Abs(a)<Abs(b)))
        Or ((A>0) And (B<0)And (Abs(a)>Abs(b)))
    Then Writeln ('Le signe de la somme ',A,' + ',B,' est positif')
    Else Writeln ('Le signe de la somme ',A,' + ',B,' est négatif');

End.
```

Exercice 26

A=10 et B=5	**premier choix**
	troisième choix
A=5 et B=5	**troisième choix**
A=5 et B=10	**quatrième choix**
A=10 et B=10	**quatrième choix**
A=20 et B=10	**premier choix**
	quatrième choix
A=20 et B=20	**deuxième choix**
	quatrième choix

Exercice 27

"premier choix" apparaît pour $(A>B)$ et $(A>10)$

"deuxième choix" apparaît pour $(10 \geq A>B)$

"troisième choix" apparaît pour $(10 \geq A>B \geq 10)$ et $(A=B)$
10>10 impossible
A>B et A=B impossible => "troisième choix" n'apparaît jamais

"quatrième choix" apparaît pour $(10 \geq A>B \geq 10)$ et $(A \neq B)$
10>10 impossible => "quatrième choix" n'apparaît jamais

LES STRUCTURES ITERATIVES

Exercice 1

```
Program  Alphabet;
Uses  Wincrt ;
Var    c : Char ;
Begin
   FOR  c:= 'A'  To  'Z'  Do  Write (c:2);
   Writeln ;
   FOR  c:= 'Z'  Downto  'A'  Do  Write (c:2);
End.
```

Exercice 2

```
Program  Table3;
Uses   Wincrt ;
Const   n = 10 ;
Var      i : Integer ;
Begin
   FOR  i:= 1  To  n  Do
        Writeln ('3*',i,' = ',3*i);
End.
```

Exercice 3

```
Program  Somme_Produit ;
Uses   Wincrt ;
Var    s, i : Integer ; p : Real ;
Begin
   S := 0 ; P := 1 ;
   FOR  i:= 1  To  20  Do
        Begin
           s := s + i ;
           p := p * i ;
        End ;
   Writeln ('Somme = ', s);
   Writeln ('Produit = ', p:2:2);
End.
```

Exercice 4

```
Program  Suite ;
Uses   Wincrt ;
Var    som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  FOR  i := 1 To  100  Do
       Begin
          som := som + u ;
          u := u + 3 ;
       End ;
  Writeln (som) ;
End.
```

Exercice 4

```
Program  Suite ;
Uses  Wincrt ;
Var    som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  i := 1 ;
  Repeat
      som := som + u ;
      u := u + 3 ;
      i := i+1 ;
  Until  (i>100) ;
  Writeln (som) ;
End.
```

Exercice 4

```
Program  Suite ;
Uses  Wincrt ;
Var    som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  i := 1 ;
  While  (i<=100)   Do
        Begin
           som := som + u ;
           u := u+3 ;
           i := i+1 ;
        End ;
  Writeln (som) ;
End.
```

Exercice 5

```
Program  Pythagore ;
Uses     Wincrt ;
Const    n = 9 ;
Var        i, j : Byte ;
Begin
    FOR  i:=1 To  n   Do
        Begin
           FOR  j:=1  To  n  Do  Write (i * j : 4) ;
           Writeln ;
        End ;
End.
```

Exercice 6

```
Program    Pyramide;
uses  wincrt;
const   N=4;
var      i, j : integer;
begin
  for  i:=0  to  n  do
     begin
        for  j:=i+1  to  n  do   write (' ');
        for  j:=-i  to  i  do   write ('*');
        writeln;
     end;
end.
```

Exercice 7

```
Program   Moy_Notes;
Uses    Wincrt;
Var      i, n : Integer;
          note, s : Real;
Begin
  Write ('Combien de notes : '); Readln (n);
  s:=0;
  FOR i:=1 To n Do
     Begin
       Write ('Note ', i, ' : ');
       Readln (note);
       s := s+note;
     End;
  Writeln ('Moyenne de ces ', n, ' notes : ', s/n:2:2);
End.
```

Exercice 8

```
Program  Factoriel ;
Uses     Wincrt ;
Var       i, n : Byte ;
           fact : Real ;
Begin
  Repeat
     Writeln ('Saisir un entier');
     Readln (n) ;
  Until  n  IN  [0..255] ;

  fact := 1 ;
  FOR   i := 2  To  n  Do  fact := fact * i ;

  Writeln (n, ' ! = ',  fact) ;
End.
```

Exercice 10

```
Program  Diviseurs;
Uses    Wincrt;
Var      n, m, r : Integer;
Begin
  Writeln ('Donner un entier');
  Readln (n);
  m:=n;
  Repeat
    r:= m mod 10;
    m:= m div 10;
    IF  (n mod r = 0)  Then Write (r, ' ');
  Until m=0;
End.
```

Exercice 11

```
Program  Som_15;
Uses    Wincrt;
Var      i, j, k : Integer;
Begin
  FOR  i:=1  TO  9   DO
      FOR  j:=1   TO   9   DO
          FOR   k:=1   TO   9   DO
                 IF  (i+j+k=15)
                      Then  Begin
                              Writeln (i, '  ', j, '  ', k);
                              Readln ;
                            End;
End.
```

Exercice 9

```
Program   Jeu ;
Uses      Wincrt ;
Label     1, 2 ;
Var       np, nc, essai : Integer ;
          z : Char ;
Begin
   1: Clrscr ;
   Randomize ;
   nc := Random (100) +1 ;
   essai := 0 ;
   Repeat
      essai := essai+1 ;
      Write ('Essai numéro ', essai, 'Votre nombre : ':20);
      Readln (np) ;
      IF  np > nc  Then  Writeln ('C''est grand')
          Else  IF  np < nc  Then  Writeln ('C''est petit')
          Else  Writeln ('Bravo vous avez gagné !!') ;
   Until   (np = nc)  Or  (essai = 7) ;

   IF np<>nc
       Then  Writeln ('Perdu, le nombre cherché est : ', nc);

   Writeln ('Voulez vous jouer encore ? (o/n)');
   Readln (z);
   IF z='n' Then goto 2 Else goto 1 ;
2:End.
```

Exercice 12

```
Program   Histogramme;
Uses      Wincrt;
 Var       a, b, c, max, i : Integer;
Begin
  Writeln ('Entrer trois entiers compris entre 0 et 15');
  Readln (a, b, c);
  max:=a;
  IF  b>max  Then   max:=b;
  IF  c>max  Then   max:=c;
  FOR   i:= max   Downto   1   Do
          Begin
             IF  i>a    Then Write (' ')
                        Else Write ('A');
             IF  i>b    Then Write (' ':4)
                        Else Write ('B':4);
             IF  i>c    Then Writeln (' ':4)
                        Else Writeln ('C':4);
          End;
End.
```

```
Program histogramme;
Uses wincrt;
Var   n1, n2, n3:integer;
Procedure lecture (var n:integer);
Begin
   writeln ('Entrer trois entiers compris entre 0 et 15');
   repeat   readln(n)   until  n  in [0..15] ;
End;
Procedure affiche (n,c:integer; ca:char);
Var  l,i:integer ;
Begin
  l:=21; {numéro de ligne}
  for i:=1 to n do
    begin
      gotoxy(c,l);
      writeln(ca);
      l:=l-1;
    end;
End;
Begin
  lecture(n1); lecture(n2); lecture(n3);
  affiche(n1,10,'A'); affiche(n2,14,'B'); affiche(n3,18,'C');
End.
```

Exercice 13

```
Program   Som_Chiffres;
Uses    Wincrt;
Var     n, som, r : Integer;
Begin
  Writeln ('Donner un entier'); Readln (n);
  som:=0;
  Repeat
     r:= n mod 10;
     som:=som+r;
     n:= n div 10;
  Until n=0;
  Writeln ('La somme de chiffres est : ', som);
End.
```

Exercice 14

```
Program   Nbr_Cube;
Uses    Wincrt;
Var      k, c, d, u : Integer;
Begin
  FOR   k:=100   To   999  Do
        Begin
           c:=k div 100 ;
           d:=(k div 10) mod 10;
           u:=k mod 10 ;
           IF  (u*u*u+d*d*d+c*c*c) = K
                     Then  Writeln (k, ' est un nombre cubique');
        End;
End.
```

Exercice 15

```
Program   Somme ;
Uses   Wincrt;
Var     n, i : Integer ;
         s1, s2, s3 : Real ;
Begin
   Repeat
      Write ('Saisir un entier impair : ');  Readln (n);
   Until   odd (n);
   s1:=0; s2:=0; s3:=0;
   FOR  i:=1 To  n  Do
        IF  odd (i)   Then   s2 := s2 + 1/i
                      Else   s3 := s3 - 1/i ;
    s1:= s2 - s3 ;
    Writeln (s1:8:2,  s2:8:2,  s3:8:2);
End.
```

Exercice 16

```
Program   Syracuse ;
Uses   Wincrt;
Var     n, i, s : Integer ;

Begin
      Writeln ('Saisir un entier > 0 ');    Readln (n);
      S:=n;
      FOR i:=1  To  50  Do
         Begin
              Write (S, ' ');
              IF   S mod 2 = 0
                     Then  S:= S div 2
                     Else   S:= 3*S+1;
         End;
End.
```

Exercice 17

```
Program   Probabilite;
Uses     Wincrt ;
Const    n = 12; essai = 100;
Var      d1, d2, d3, cumul, i : Byte;
Begin
   Randomize;
   cumul :=0;
   FOR   i:=1 To  essai   Do
        Begin
           d1 := 1 + Random (6);
           d2 := 1 + Random (6);
           d3 := 1 + Random (6);
           IF (d1+d2+d3 = n)  Then    cumul := cumul +1;
         End;
  Writeln ('Probabilité est : ', cumul/essai :5:2);
End.
```

Exercice 18

```
Program   Puissance_n ;
Uses     Wincrt ;
Var      n, k : Integer ;
         y, x : Real ;
Begin
    Write ('Saisir un nombre réel : ') ;  Readln (x) ;
    Write ('Saisir la puissance n : ') ;   Readln (n) ;
    y := 1 ;
    FOR  k := 1 To  abs (n)  Do  y := y * x ;
    IF  n<0  Then  y := 1/y ;
    Writeln (x:5:2, ' puissance '  , n , ' = ' , y:5:2) ;
End.
```

Exercice 19

```
Program   Multiplication_Addition ;
Uses      Wincrt ;
Var       x, y, s, aux, i : Integer ;
Begin
   Writeln ('Donner deux entiers') ; Readln (x, y) ;
   Write (x, ' * ', y, ' = ');
   IF  abs(y)>abs(x)  Then   Begin
                                aux := x ;
                                x := y ;
                                y := aux ;
                            End ;
    IF  y<0   Then   Begin
                          y:= -y ;
                          x:= -x ;
                     End ;
    s := 0 ;
    FOR   i:=1  To  y  Do   s := s+x ;

    Writeln (s) ;
End.
```

Exercice 20

```
Program    Suite ;
Uses      Wincrt ;
Var       i, n : Integer ;   s, invfact : Real ;
Begin
   Writeln ('Donner un entier');  Readln (n) ;
   s := 1 ;
   invfact := 1 ;
   FOR  i := 1  To  n  Do
       Begin
          invfact := invfact/i ;
          s := s + invfact ;
       End ;
   Writeln (s:5:2) ;
End.
```

Exercice 21

```
program    produits;
uses  wincrt;
var  a, b, c, d:integer;
begin
    for a:=1 to 9 do
       for c:=a to 9 do
          for b:=c downto a do
           for d:=c downto a do
             if ((10*a+b)*(10*c+d) = (10*b+a)*(10*d+c))
                        and (a<>b) and (b<>c)
         then   Writeln (a, b, ' * ', c, d, ' = ', b, a, ' * ', d, c) ;
end.
```

Exercice 22

```
PROGRAM    PI_WALLIS;
USES   WINCRT;
VAR      P, r, diff : Real;
          i : LONGINT;
BEGIN
      P:=1; i:=0;
      REPEAT
           i:=i+2;
           r:=i/(i-1)*i/(i+1);
           diff:= (P*r) - P;
           P:=P*r;
      UNTIL  abs (diff) < 1e-8;
      WRITELN ('Par la formule de Wallis Pi = ',  2*P:2:7);
END.
```

Exercice 23

```
Program   PGCD_Euclide;
Uses     Wincrt;
Var      a, b, r : Integer;
Begin
    Repeat
         Writeln ('Saisir deux entiers > 0'); Readln (a, b);
    Until   (a>0) and (b>0) ;
   While   b<>0   Do
       Begin
         r := a mod b;   a := b;   b := r;
       End;
   Writeln ('PGCD =  ',  a);
End.
```

Exercice 24

```
Program   PGCD_Diff;
Uses    Wincrt;
Var     a, b : Integer;
Begin
     Repeat
          Writeln ('Saisir deux entiers  >0 '); Readln (a, b);
     Until   (a>0) and (b>0) ;
    While   a<>b   Do
         IF  a>b   Then     a:= a-b
                       Else     b:= b-a ;

    Writeln ('PGCD =  ', a) ;
End.
```

Exercice 25

```
Program   PPCM;
Uses     Wincrt ;
Var       pcm, m, n, aux : Integer;
Begin
    Repeat
         Writeln ('Saisir deux entiers > 0'); Readln (m, n) ;
    Until  (m>0) and (n>0);
    IF  m < n    Then   Begin
                                aux:= m;
                                m := n;
                                n := aux;
                            End;
    pcm := m;
    While   (pcm mod n <> 0)   Do   pcm := pcm + m;
    Writeln ('PPCM =  ', pcm);
End.
```

Exercice 26

```
Program    Fibonacci ;
uses      wincrt ;
var       k, f0, f1, f2 : integer ;
begin
      f0 := 1 ; f1 := 1 ;
      write (f0, '  ', f1, '  ') ;

      for   k := 2  to   19   do
          begin
                f2 := f1+f0 ;
                f0 := f1 ;
                f1 := f2 ;
                write (f2, '  ') ;
           end ;
end.
```

Exercice 27

```
Program   Nbre_Premiers ;
uses     wincrt ;
var       nb, i : integer ;
begin
     for   nb := 2  to  400  do
         begin
             i := 2 ;
             while  (nb mod i <> 0) and (i <= nb div 2)  do   i:= i+1 ;
             if  (i > nb div 2)  then  write (nb:4) ;
          end ;
end.
```

Exercice 28

```
Program   Parfait ;
uses     wincrt;
var       nb, d, som, a, b : integer;
begin
     repeat
          Readln(a,b);
     until  (1<a) and (a<b);
     for   nb:=a   to  b   do
           begin
               som:=0;
               for  d:=1  to  (nb div 2)  do
                       if  (nb mod d = 0)   then  som:=som+d ;
               if  nb=som   then  writeln (nb, ' est parfait');
           end;
end.
```

Exercice 32

```
Program   Exponentiel;
uses wincrt;
var    x, s, epsilon, p, f:real;
        i:integer;
begin
     write ('epsilon = '); readln (epsilon);
     write ('x = '); readln (x);
      s:=1; i:=1; p:=1; f:=1;
      repeat
            p:=p*x;
            f:=f*i;
            s:=s+p/f;
            i:=i+1;
      until   abs(p/f) <= epsilon;
      writeln ('expn = ', s:2:10);
end.
```

Exercice 29

```
Program   Amis;
uses     wincrt;
var       m, n, sdn, sdm : integer;
(************************************)
function  diviseurs (x : integer) : integer;
var    sdx, i : integer;
begin
   sdx:=1;
   for  i:=2  to  (x div 2) do
        if (x mod i) = 0  then sdx := sdx+i;
   diviseurs:=sdx;
end;
(************************************)
begin
   for  m:=1  to  1000  do
      for  n:=1  to  1000  do
         begin
           sdn := diviseurs (n);
           sdm := diviseurs (m);
           if (sdm=n) and (sdn=m)
                then   writeln (n, ' et ', m, ' sont amis') ;
         end;
end.
```

Exercice 30

```
Program   Facteur_Premier ;
uses      wincrt ;
type      tab = array [1..100] of  integer ;
var        fp : tab ;
            n, i, f : integer ;
begin
   repeat
      writeln ('donner un entier entre 2 et 1000') ;  readln (n) ;
   until  (n>=2) and (n<=1000) ;
   write (n, ' = ');
   i := 2 ; f:=0;
   repeat
      if  (n mod i = 0)
           then  begin
                        n:= n div i;
                        f:=f+1;
                        fp[f] := i;
                      end
            else   i:=i+1;
   until (n=1);
   write (fp[1]);
   for   i:=2  to   f   do   write (' * ', fp[i]);
end.
```

Exercice 31

```
Program calcul_sinus;
uses   wincrt;
var    x:real;
(************************************)
function  sinus (x:real) : real;
var   som, term1, term2, i:real;
begin
    som:=x;
    term2:=x;
    i:=1;
    repeat
         i:=i+2;
         term1:=term2;
         term2:=term2 * -sqr(x) / (i*(i-1));
         som:=som+term2;
    until  abs (term2-term1) <= 0.0001;
    sinus:=som;
end;
(*****************P.P*****************)
begin
  repeat
       write ('donner un réel x ');
       readln(x);
  until  (- Pi <= x) and (x <= Pi);
  write ('sin(', x:1:2,') = ', sinus(x):10:10);
end.
```

Exercice 33

```
Program   Somme;
Uses   Wincrt;
Var     i, n : Integer; s : Real;
(***************************************)
Procedure saisie (Var m : Integer);
Begin
   Repeat
      Writeln ('Donner un entier positif');
      Readln (m);
   Until  m>0;
End;
(****************************************)
Function  puissance (x : Integer):Longint;
Var    j : Integer ;  p : Longint;
Begin
   p:=1;
   FOR  j:=1 To  x  Do   p:=p*x;
   puissance:=p;
End;
(****************************************)
Begin
   saisie (n);
   s:=0;
   FOR  i:=1  To  n  Do  s:= s + (2*i-1) / puissance (i);
   Writeln ('la somme = ', s:2:3);
End.
```

Exercice 34

```
Program   Combinaison ;
Uses    Wincrt ;
Var      cnp : Real ;
          n, p : Integer ;
(****************************************)
Function    Fact (x : Integer) : LongInt ;
Var     f : LongInt ;    i : Integer ;
Begin
   f := 1 ;
   FOR   i := 2  To  x  Do  f := f * i ;
   fact := f ;
End ;
(**************************************)
Begin
   Repeat
     Writeln (‘Donner deux entiers : ’);
      ReadLn (p, n);
   Until  (0<p)  and  (p<n) ;
   cnp := fact (n) / (fact (p) * fact (n-p)) ;
   Writeln ('Combinaison =  ', cnp :4:2) ;
End.
```

Exercice 34

```
Program   Combinaison ;
Uses    Wincrt ;
Var      cnp : Real ; n, p, i : Integer ;
          Nf, pf, npf : Longint ;
Begin
   Repeat
       Write (‘p = '); Readln (p);
       Write (‘n = '); Readln (n);
   Until  (0<p)  and  (p<n) ;
   Nf :=1;
   Pf :=1;
   Npf :=1;
   FOR  i:=2  To  n  Do
          Begin
            Nf := nf*i ;
            IF  i<=p    Then  pf := pf*i;
            IF  i<=n-p  Then  npf := npf*i;
           End;
    Cnp := nf / (pf*npf) ;
   Writeln ('Combinaison =  ', cnp :4:2) ;
End.
```

Exercice 35

```
Program     Ordered ;
Uses      Wincrt;
Var       n : Integer;    i, mn, mx, p : Longint;
(**************************************)
Procedure    min_max (m : Integer; Var min, max:Longint);
Var    i : Integer;
Begin
  min:=0; max:=0;
  FOR  i:=1  To  m  Do
         Begin
            min:=10*min + i;
            max:=10*max + (9-m+i);
         End;
End;
(************************************)
Function   ordre (m:Longint) : Boolean;
Var      c1, c2 : Integer;   valide : Boolean;
Begin
  Repeat
     c1:= m MOD 10;
     m:= m DIV 10;
     c2:= m MOD 10;
     valide:= (c1>c2);
  Until  Not (valide) Or (m<10) ;
      ordre:=valide;
End;
(*********************************************)
Procedure saisie (Var m : Integer);
Begin
   Repeat
      Write ('Donner un entier n compris entre 2 et 8 : ');
      Readln (m);
   Until m in [2..8];
End;
(*********************************************)
Begin
   saisie (n);
   min_max (n, mn, mx);
   p:=0;
   FOR  i:=mn  To  mx  DO
           IF  ordre (i)  Then
                           Begin
                               p:=p+1;
                               Writeln (p:10, ' - ', i); Readln ;
                       End;
End.
```

Exercice 36

```
Program    Nbre_Impairs;
Uses       Wincrt;
Var        i, n : Integer;
Begin
    n:=0;
    FOR  i:=1  To  99  Do
        IF  ODD (i)  and  (i mod 7 <>0)
               Then  Begin
                             n:=n+1;
                             IF   n mod 5 <>0
                                      Then  Write (i:4)
                                      Else    Writeln (i:4);
                         End;
End.
```

Exercice 37

```
program  sommes_entiers;
uses wincrt;
var   n, i, j, k, s:integer;
begin
   writeln ('entrer la valeur de N :'); readln (n);
   for   i:=1  to  n div 2  do
       begin
         S := i;  j:= i;
         Repeat
              J := j +1;
              S :=  S + j;
         until  s >= n;
        if  S=N  then
             begin
               write (n,' = ',i);
               for k:=i+1 to j do  write(' + ',k);
               writeln;
             end;
        end;
end.
```

Exercice 38

```
Program  carre_parfait;
uses   wincrt;
var     n,k:integer;
Begin
    for n:=1 to 9999 do
        begin
            k:=0;
            repeat
                k:=k+1;
            until (k*k)>=n;
            if  sqr(k)=n
                   then writeln(n);
       end;
End.
```

Exercice 39

```
Program  Reine ;
Uses   Wincrt;
Var     x, y, i, j : Byte ;
Begin
  Write ('Les coordonnées de la dame: X = ') ; Readln (X);
  Write ('                                   Y = ') ; Readln (Y);
  FOR  i:=1  To  8  Do
    Begin
          FOR  j:=1  To  8  Do
          IF (i=x) and (j=y)
              Then  Write (' R ')
               Else  IF (i=x) Or (j=y) Or (abs(x-i)=abs(y-j))
                           Then    Write (' * ')
                            Else     Write ('   ');
          Writeln ;
    End;
End.
```

Exercice 40

```
Program  ppcm_pgcd;
uses wincrt;
var a, b : integer;
(************************************)
Procedure  saisie (var a,b:integer);
begin
    repeat
      writeln('Donner deux entiers >0');
      readln (a, b);
    until (a>0) and (b>0);
end;
(***********************************)
Procedure  affiche(a,b:integer);
var k:integer;
begin
  k := 0;
  repeat
       k := k+1
  until (a*k) mod b = 0;
  writeln ('ppcm de ', a, ' et ', b, ' = ', a*k);
  writeln ('pgcd de ', a, ' et ', b, ' = ', b div k);
end;
(***************** P.P ****************)
BEGIN
   saisie(a,b);
   affiche(a,b);
END.
```

Exercice 41

```
Program   produit_ab;
uses  wincrt;
var  a, b,ppcm,pgcd : integer;
(**********************************)
Procedure  pgcd_ppcm(a,b:integer;var pgcd,ppcm:integer);
var   k:integer;
begin
  k := 0;
  repeat
     k := k+1
  until (a*k) mod b = 0;
  ppcm:=a*k;
  pgcd:=b div k;
end;
(***************** P.P ****************)
begin
  writeln('Donner a et b : ');
  readln (a, b);
  pgcd_ppcm(a,b,pgcd,ppcm);
  writeln(a,' * ',b,' = ',pgcd*ppcm);
end.
```

Exercice 42

```
Program  Exercice_42;
uses wincrt;
var  a,b,s:real;
       n,i,signe:integer;
(**************************************)
Procedure  saisie(var a,b:real;var n:integer);
begin
   writeln('donner a et b');  readln(a,b);
   repeat
      writeln('donner n ');
      readln(n);
   until (n>=3) and odd(n);
end;
(**********************************)
Function  power ( x:real;n:integer):real;
var   k:integer;
         p:real;
begin
   p:=1;
   for k:= 1 to n do p:=p*x;
   power:=p;
end;
(***************** P.P ****************)
begin
   saisie(a,b,n);
   s:=0;
   signe:=1;
   for i:=0 to n-1 do
       begin
         s:=s+signe*power(b,i)*power(a,n-i-1);
         signe:=-signe;
       end;
   writeln((a+b)*s:2:2);
end.
```

Exercice 43

```
Program    divisible_11;
uses     wincrt;
var      x:integer;
(*************************************)
Procedure  saisie (var x:integer);
begin
   repeat
      write('Donner un entier ');
      readln(x);
   until x>0;
end;
(************************************)
Function  divs_11(x:integer):boolean;
var   signe,som:integer;
begin
  signe:=1;
  som:=0;
  repeat
       som:=som+signe*(x mod 10);
       x:=x div 10;
       signe:=-signe;
  until x=0;
  divs_11:= som mod 11 = 0;
end;
(*************** P.P **************)
begin
  saisie(x);
  if  divs_11(x)
        then  writeln('divisible par 11')
        else  writeln('non divisible par 11');
end.
```

Exercice 44

```
Program  somme ;
Uses    WinCrt ;
Var      n,p,signe:Integer;
           som:real;
(*************************************)
Function  Comb(p,n:integer):real;
     (***************************)
     Function   Fact (x:integer):LongInt;
     var     f:LongInt; i:integer;
     begin
       f:=1;
       for i:=2 to x do  f:=f*i;
       fact:=f;
     end;
     (************************)
begin
   comb:=fact(n)/(fact(p)*fact(n-p));
end;
(***************** P.P ****************)
Begin
   Writeln ('Donner n : ');
   ReadLn (n);
   som:=1;
   signe:=-1;
   for p:=1 to 2*n do
      begin
        som:=som+signe*sqr(comb(p,2*n));
        signe:=-signe;
      end;

    Writeln ('somme = ',som:2:2) ;
End.
```

Exercice 45

```
Program  divis_7_13;
uses  wincrt;
var    n,nb:integer;
(**************************************)
Function  div_7 (n : integer ) : boolean ;
begin
    while  (n>99)  do   n := (n div 10) - 2 * (n mod 10) ;
    div_7:=(n mod 7 = 0);
end ;
(************************************)
Function  div_13 (n : integer ) : boolean ;
begin
    while (n>99) do   n := (n div 10) + 4 * (n mod 10) ;
    div_13:=(n mod 13 =0);
end ;
(***************** P.P ****************)
BEGIN
    writeln('les nombres divisibles par 7 :');
    nb:=0;n:=0;
    repeat
         if  div_7(n)  then begin
                                write(n:5);
                                 nb:=nb+1;
                             end;
          n:=n+1;
   until   nb=100;
   writeln;
   writeln('les nombres divisibles par 13 :');
   nb:=0; n:=0;
   repeat
          if  div_13(n)  then  begin
                                   write(n:5);
                                   nb:=nb+1;
                                 end;
           n:=n+1;
   until   nb=100;
END.
```

Exercice 46

```
Program  Nbre_kaprekar;
uses  wincrt;
var   k: integer;
(**********************************)
Function kaprekar(m : longint): boolean;
var   l,n1,n2,err :integer;
      ch,ch1,ch2 : string;
begin
   str(sqr(m),ch);
   l := length(ch);
   ch1 := copy(ch, 1, l div 2);
   ch2 := copy(ch, (l div 2)+1, l);
   val(ch1,n1,err);
   val(ch2,n2,err);

   kaprekar := (m=n1+n2);
end;
(*************** P.P ***************)
Begin
 For  k:=1  to  1000  do
         if  kaprekar(k)  then writeln(k);
End.
```

Exercice 48

```
Program  frac_egypt;
uses     wincrt;
var      i,n,d:longint;
begin
  write('n=');readln(n);
  write('d=');readln(d);
  repeat
       i:=(d div n)+1;
       write(1,'/',i,' + ');
       n:=i*n-d;
       d:=i*d;
  until d mod n =0;
  write(1,'/',d div n);
end.
```

Exercice 47

```
Program  Premier_circulaire;
uses wincrt;
var  p,q,n:integer;
(************************************)
Procedure  saisie(var p,q:integer);
begin
  repeat
    write('p=');readln(p);
    write('q=');readln(q);
  until (10<p)and(p<q)and(q<20000);
end;
(*************************************)
Function  circulaire(n:integer):boolean;
Var  err,i:integer;
     ok:boolean;
     ch:string;

function premier(n:integer):boolean;
var  i,d:integer;
begin
  d:=2;
  for i:=2  to (n div 2) do
        if (n mod i)=0  then d:=d+1;
  premier := (d=2);
end;

begin
      ok:=premier(n);
      if ok
         then begin
                str(n,ch);i:=0;
                repeat
                  i:=i+1;
                  ch:=ch[length(ch)]+copy(ch,1,length(ch)-1);
                  val(ch,n,err);
                  ok:=premier(n);
                until (i=length(ch)-1) or (not ok);
              end;
     circulaire:=ok;
end;
(*****************P.P******************)
BEGIN
  saisie(p,q);
  for n:=p to q do
      if  circulaire(n)   then write(n,' ');
END.
```

Exercice 49

```
Program super_premier;
uses   wincrt;
var     n,i:longint;
(*****************************************)
Procedure saisie(var n:longint);
begin
  repeat
     write('n = ');
     readln(n);
  until (40000<n) and (n<100000);
end;
(************************************)
Function premier(x:longint):boolean;
var i,d:longint;
begin
  d:=2;
  for i:=2 to (x div 2) do
       if x mod i =0 then d:=d+1;
  premier:= d=2;
end;
(*****************************************)
Function super(x:longint):boolean;
var test:boolean;
begin
  test:=false;
  repeat
    x:=x div 10;
    if x<>1 then test:=premier(x);
  until (test=false) or (x<10);
  super:=test;
end;
(*****************P.P**********************)
begin
   saisie(n);
   for i:=2 to n do
           if premier(i)
               then if super(i)
                           then writeln(i,' super premier')
                           else writeln(i);
end.
```

Exercice 51

```
Program  jeux_allumette;
uses  wincrt;
var    i,j,s:integer;
begin
  randomize;
  j:= 10+random(20);
  writeln('Jeu avec ',j,' allumettes');
  while j > 0 do
    begin
      randomize;
      if j > 3 then i := 1+random (3)
            else if j=3 then i:=1+random(2)
                        else i:=1;
         j:=j-i;
         writeln('Je prend ',i,' allumette(s). Il en reste ',j);
        if j=0 then writeln('Bravo vous avez gagné !')
              else    repeat
                        writeln('Donnez votre jeu: 1 ou 2 ou 3');
                       readln(s);
                     until s in[1..3];
        j := j - s;
        if  j = 0 then writeln('Vous avez perdu !');
     end;
end.
```

Exercice 50

```
Program  Conversion_base2_base10 ;
Uses  wincrt;
Var  ch_bin:string;
(***********************************************)
procedure saisir(var ch_bin:string);
var binaire:boolean;
   i:byte;
begin
repeat
   writeln('Donner un nombre binaire');
   readln(ch_bin);
   i:=0;
   repeat
     i:=i+1;
     binaire :=ch_bin[i] in ['0','1'];
   until (not binaire) or (i=length(ch_bin));
  until (binaire=true);
end;
(******* Conversion de la base2 vers base10 ********)
Function Conv_b2_b10(ch_bin:string):longint;
var  i:byte; dec,puiss:longint;
begin
  dec:=0;puiss:=1;
  for  i:=length(ch_bin) downto 1 do
      begin
       if ch_bin[i]='1' then  dec:=dec+puiss;
       puiss:=puiss*2;
      end;
  conv_b2_b10:=dec;
end;
(******************* P.P ************************)
begin
 saisir(ch_bin);
 writeln('(',ch_bin,')2',' =(',conv_b2_b10(ch_bin),')10');
end.
```

Exercice 52

```
Program  auto_nombre;
uses wincrt;
var n:integer;
(***************************************)
function verif(n:integer):boolean;
var  y:integer;
    test:boolean;
(*************************)
function somchif(x:integer):integer;
var sc:integer;
begin
  sc:=0;
  repeat
     sc:=sc+ x mod 10;
     x:=x div 10;
  until x=0;
  somchif:=sc;
end;
(**************************)
begin
  y:=n;
  repeat
     y:=y-1;
     test:=n=y+somchif(y);
  until (test) or (y<=n div 2);
  verif:=test;
end;
(**************************************)
begin
  for n:=1 to 999 do
   if verif(n)=false
         then writeln (n,' est auto nombre');
end.
```

Exercice 53

```
Program  vampire;
uses wincrt;
var n:integer;
(***************************************)
function verif(n:integer):boolean;
var c,d,u,p1,p2,p3,p4,p5,p6:integer;
begin
  c:=n div 100;
  d:=n div 10 mod 10;
  u:=n mod 10;
  p1:=c*(d*10+u);
  p2:=c*(u*10+d);
  p3:=d*(c*10+u);
  p4:=d*(u*10+c);
  p5:=u*(d*10+c);
  p6:=u*(c*10+d);
  verif:=(n=p1)or(n=p2)or(n=p3)or(n=p4)or(n=p5)or(n=p6);
end;
(**************************************)
begin
  for n:=100 to 999 do
       if verif(n)
          then write(n,' ');

end.
```

Exercice 53

```
Program  PGCD_fact_prem;
Uses Wincrt ;
Type    Tab = Array [1..100] of  integer ;
Var     fa,fb : Tab ;
        a,b,na,nb : integer ;

(**********************************************)
procedure saisie (var a,b:integer);
Begin
  Repeat
    write('a = '); Readln(a) ;
    write('b = '); readln(b);
  Until  (10<=a) and (a<=b) and (b<=10000) ;
end;
(**********************************************)
procedure factprem(n:integer;var fp:tab;var f:integer);
var  i : integer ;
Begin
  i := 2 ;f:=0;
  repeat
    if n mod i = 0
       then begin
              n:= n div i;
              f:=f+1;
              fp[f] := i;
            end
       else i:=i+1;
  until (n=1);
end;
(**********************************************)
function pgcd(fa,fb:tab;na,nb:integer) : longint;
var   t:array[1..100] of integer;
      i,j,k:integer;
      f:longint;
begin
  i:=1;j:=1;k:=0;
  repeat
    if fa[i]=fb[j]
      then  begin
              k:=k+1;
              t[k]:=fa[i];
              i:=i+1;
              j:=j+1;
            end
      else if fa[i]>fb[j]  then j:=j+1
                           else  i:=i+1;
  until (i>na) or (j>nb);
  f:=1;
  for i:=1 to k do  f:=f*t[i];

  pgcd:=f;
end;
(************************P.P*******************)
BEGIN
  saisie(a,b);
  factprem(a,fa,na);
  factprem(b,fb,nb);
  writeln('PGCD(',a,',',b,')',' = ',pgcd(fa,fb,na,nb));
End.
```

Exercice 54

```
Program  PPCM_fact_prem;
Uses Wincrt ;
Type    Tab = Array [1..100] of  integer ;
Var     fa,fb : Tab ;
        a,b,na,nb : integer ;

(**********************************************)
procedure saisie (var a,b:integer);
(**********************************************)
procedure factprem(n:integer;var fp:tab;var f:integer);
(**********************************************)
function ppcm (fa,fb:tab;na,nb:integer) : longint;
var  t:array[1..200] of integer;
     i,j,k,m:integer;
     f:longint;
begin
  i:=1;j:=1;k:=0;
  repeat
    k:=k+1;
    if fa[i]=fb[j]
      then  begin
                t[k]:=fa[i];
                i:=i+1;
                j:=j+1;
              end
      else if fa[i]<fb[j] then begin
                                t[k]:=fa[i];
                                i:=i+1;
                              end

                          else  begin
                                  t[k]:=fb[j];
                                  j:=j+1;
                                end;
  until (i>na) or (j>nb);
  if i>na
    then  for m:=j to nb do
               begin
                  k:=k+1;
                  t[k]:=fb[m];
               end
    else  for m:=i to na do
               begin
                 k:=k+1;
                 t[k]:=fa[m];
               end;
  f:=1;
  for i:=1 to k do  f:=f*t[i];

  ppcm:=f;
end;
(*****************P.P***************************)
BEGIN
  saisie(a,b);
  factprem(a,fa,na);
  factprem(b,fb,nb);
  writeln('PPCM(',a,',',b,')',' = ',ppcm(fa,fb,na,nb));
End.
```

Exercice 57

```
Program  Range;
uses wincrt;
Var   F,G : Integer;
(**************************************)
Procedure Affiche (F,G:Integer);
Var     N,X,Y,I,l:Integer;
Begin
     n:=0;X:=30;Y:=10; l:=(1+f) div g;
     for  i:=1 to g do
        begin
            repeat
                n:=n+1;
                Gotoxy(X,Y);
                Writeln(n);
                Inc(Y);
```

Exercice 58

```
Program Perles;
Uses  Wincrt;
Var    X,Y : Word;
(*************************************************)
Function Pgcd(A,B : Word) : Word;
  Var R : Word;
  Begin
     While B<>0 Do
         Begin
            R:=A mod B;
            A:=B;
            B:=R;
         End;
     Pgcd:=A;
  End ;
```

```
            until (y=I+10) or (n=f);
            Inc(X,10);Y:=10;
        end;
End;
(*********************************************)
BEGIN
   Repeat
       Write('Donner un entier F : '); Readln(F);
       Write('Donner un entier G : '); Readln(G);
   Until (F>G) And (G>0);
   Affiche (F,G);
End.
```

```
(**************************************************)
BEGIN
    Repeat
        Write ('Donner Le nombre de perles blanches (X) : ');
        Readln (X);
    Until (X>0);
    Repeat
        Write ('Donner Le nombre de perles noires (Y) : ');
        Readln (Y);
    Until (Y>0);
    Write ('Le nombre maximum de répétitions est : ',Pgcd (X,Y));
END.
```

Exercice 59

```
Program Nombres;
Uses   Wincrt;
Var   N,I : Longint;
(**********************************************)
Procedure Lecture (Var N : Longint);
  Begin
      Repeat
          Write('Donner un entier : ');  Readln(N);
      Until (N>1);
  End;
(***********************************************)
Function Palindrome (Nb : Longint) : Boolean;
  Var      Ch : String;
  Begin
      Str(Nb,Ch);
      While (Length(Ch)>1) And (Ch[1]=Ch[Length(Ch)]) Do
        Ch := Copy(Ch,2,Length(Ch)-2);
      Palindrome := Length(ch)<=1;
  End;
(*********************************************)
Procedure Affiche (Nb : Longint);
  Var      J,K,Somme : Longint;
           Ch,Ch1: String;
  Begin
      J := 0;
      Repeat
          J:=J+1;
          Somme:=0;
          K:=J;
          Ch:='';
          While (Somme<Nb) Do
              Begin
                  Somme:= Somme + Sqr(K);
                  Str(K,Ch1);
                  Ch:= Ch + Ch1 + '^2 + ';
                  K:= K + 1;
              End;
      Until (Somme=Nb) Or (J>Sqrt(Nb));
      Delete(Ch,Length(Ch)-2,3);
      If Somme=Nb Then Writeln(Nb ,' = ',Ch);
  End;
(********************P.P**********************)
BEGIN
   Lecture(N);
   For  I := 1 To  N  Do
      If Palindrome(I) Then Affiche(I);
END.
```

Exercice 60

```
Program   Smith;
Uses  Wincrt;
Var     Nb,M,N:Integer;
(*********************************)
Procedure Saisie(Var M,N:Integer);
Begin
  Repeat
     Writeln('Saisir M Et N');
     Readln(M,N);
  Until (10<M) And(M<N) And(N<100);
End;
(**********************************)
Function Som_Chif(Nb:Integer):Integer;
Var   S:Integer;
Begin
  S:=0;
  Repeat
     S:=S + (Nb Mod 10);
     Nb:=Nb Div 10;
  Until Nb=0;
  Som_Chif:=S;
End;
(***********************************)
Function Som_Fact(Nb:Integer):Integer;
Var  D,S:Integer;
Begin
  D:=2;
  S:=0;
  Repeat
     If Nb Mod D =0
       Then  Begin
                Nb:=Nb Div D;
                S:=S+Som_Chif(D);
              End
       Else D:=D+1;
  Until Nb=1;
  Som_Fact:=S;
End;
(*****************P.P*******************)
Begin
  Saisie(M,N);
  For  Nb:=M  To  N  Do
    If Som_Chif(Nb)=Som_Fact(Nb)
         Then Writeln(Nb);
End.
```

Exercice 61

```
Program  Harshad_Zuckerman ;
Uses  Wincrt ;
Var   M,N,I : Integer ;
        Ch:String;
(*******************************************)
Procedure  Saisir(Var N,M : Integer ) ;
Begin
   Repeat
      Write('N = ') ;
      Readln(N) ;
   Until N>=100 ;
   Repeat
      Write('M = ') ;
      Readln(M) ;
   Until N<M ;
End ;
(*********************************************)
Function  Harzuc ( Nb : Integer ) : Boolean ;
Var  I,Som,V,Err,Prod : Integer ;
    Ch : String ;
Begin
   Str(NB,Ch) ;
   Som:=0 ;Prod:=1;
   For  I:=1  To  Length(Ch)  Do
        Begin
            Val(Ch[I],V,Err) ;
            Som:=Som+V ;
             Prod:=Prod*V;
        End ;
   Harzuc:= (Nb Mod Som  = 0) And (Nb Mod Prod =0)
End ;
(************* P.P.***************************)
BEGIN
   Saisir(N,M) ;
   For I:=N To M Do
      Begin
          Str(I,Ch) ;
          If Pos('0',Ch)=0
                 Then If Harzuc(I)
                               Then Writeln(I);
      End;
END.
```

Exercice 62

```
Program   STEINHAUS;
Uses   Wincrt;
Var     N,D,U,Err,K:Integer;
          Ch,C:String;
Begin
            Randomize;
             N:=10+Random(90);
             Str(N,Ch);
             K:=0;
             Repeat
                 K:=K+1;
                 Val(Ch[K],D,Err);
                 Val(Ch[K+1],U,Err);
                 Str(D*U,C);
                 Ch:=Ch+C;
             Until Length(Ch)=100;
             Writeln(Ch);
End.
```

Exercice 63

```
Program    Pidovan ;
Uses     Wincrt ;
Var      K, P0, P1, P2, Pn : Integer ;
Begin
       P0:=1;
       P1:=1 ;
       P2:=1;

      Write (P0, '  ', P1, '  ',P2) ;

      For   K:=3  To   19   Do
          Begin
              Pn := P0+P1 ;
              P0 := P1 ;
              P1 := P2 ;
               P2 := Pn;
              Write ('  ',Pn) ;
         End ;
End.
```

Exercice 64

```
Program Nb_Riche;
Uses  Wincrt;
Var    N:Integer;
(*********************************************)
Function  Nbfact(N:Integer):Integer;
Var    I,K:Integer;
Begin
      K:=0;
      I:=2;
      Repeat
          If  N Mod I  = 0
                 Then  Begin
                             K:= K + 1;
                             N := N Div I;
                          End
                 Else  Begin
                             I := I + 1;
                             K:=0;
                         End;
       Until (N=1) Or (K>=2);
       Nbfact:=K;
End;
(***************P.P*************************)
Begin
     For N:=1 To 1000 Do
       If   Nbfact(N)>=2
             Then  Writeln (N);
End.
```

Exercice 65

```
Program  Harshad_Morphique ;
Uses  Wincrt ;
Var     I : Longint ;
(*********************************************)
Function  Hm ( Nb : Longint ) : Boolean ;
Var    I,Som,V,Err : Integer ;
         Ch : String ;
Begin
   Str(Nb,Ch) ;
   Som:=0 ;
   For  I:=1  To  Length(Ch)  Do
        Begin
           Val(Ch[I],V,Err) ;
           Som:=Som+V ;
        End ;
   Hm:= (Nb Mod Som  = 0)And(Nb Mod 100  = Som);
End ;
(************* P.P.***************************)
Begin
    For   I:=100  To  99999   Do
          If   Hm(I)  Then   Writeln(I);
End.
```

Exercice 66

```
Program narcissiques;
Uses   Wincrt;
Var    L,Err,Nb,N,Som,J:Integer;
       Ch:String;
(**********************************************)
Function Puissance(N,L:Integer):Integer;
Var   P,I:Integer;
Begin
    P:=1;
    For  I:=1  To  L   Do   P:=N*P;
    Puissance:=P;
End;
(****************P.P***********************)
Begin
  For  Nb:=1   To   10000   Do
      Begin
        Str(Nb,Ch);
        L:=Length(Ch);
        Som:=0;
        For  J:=1  To  L   Do
          Begin
            Val(Ch[J],N,Err);
            Som:=Som+Puissance(N,L);
         End;
       If Nb=Som
          Then   Writeln(Nb,' Est Un Nombre Narcissique ');
     End;
End.
```

Exercice 67

```
Program  Nbr_Freres;
Uses Wincrt;
Var N1,N2:Integer;
(**********************************************)
   Procedure Saisie(Var N:Integer);
   Begin
       Readln(N)
   End;
(**********************************************)
   Function Frere(N1,N2:Integer):Boolean;
   Var  Ch1,Ch2:String;
          V:Boolean;
          I:Integer;
   Begin
       Str(N1,Ch1);
       Str(N2,Ch2);
       I:=1;
       Repeat
            V:=Pos(Ch1[I],Ch2)<>0
            I:=I+1;
       Until (I>Length(Ch1)) Or (V= False);
       Frere:=V;
   End;
(******************P.P**********************)
Begin
  Write('N1= ');
  Saisie(N1);
  Write('N2= ');
  Saisie(N2);
  If Frere(N1,N2) And Frere(N2,N1)
      Then  Writeln(N1,' Et ',N2,' Sont Frères')
      Else   Writeln(N1,' Et ',N2,' Ne Sont Pas Frères');
End.
```

Exercice 69

```
Program  Unitairement_Parfait;
Uses  Wincrt;
Var    N:Longint;
(*********************************************)
Function Sdu(Nb:Longint):Longint;
  Var   D,Som:Longint;
  Function Pgcd(A,B:Longint):Longint;
    Begin
        While A<>B Do
             If A>B Then  A:= A-B
                    Else  B:= B-A ;
        Pgcd:=A;
   End;
Begin
   Som:=0;
   For  D:=1  To  (Nb Div 2) Do
        If  (Nb Mod D =0) And (Pgcd(D,Nb Div D)=1)
               Then Som:=Som+D;
   Sdu:=Som;
End;
(*******************P.P********************)
Begin
 For  N:=1  To  100000   Do
    If  Sdu(N)=N
        Then Writeln(N);
End.
```

LES TABLEAUX

Exercice 1

```
Program   Som_Produit_MoyArith ;
Uses    Wincrt ;
Type    Tab = Array [1..10] of  Integer ;
Var      T : Tab ;  n, i, st : Integer ;  mt, pt : Real ;
Begin
    Repeat
        Write ('N = ');
        Readln (n) ;
    Until    (n>5) and (n<=10) ;
    FOR   i := 1 To   n  Do
         Repeat
             Write ('T', i, ' = ');
             Readln (T[i]) ;
        Until  (1<= T[i]) and (T[i] <= 20) ;
    st := 0 ;
    pt := 1 ;
   FOR  i := 1 To  n   Do
       Begin
           St := st + T[i] ;
           Pt := pt * T[i] ;
        End ;
   Writeln ('Somme = ', st);
   Writeln ('Produit = ', trunc (pt));
   Writeln ('Moyenne arithmétique = ', st/n:2:2);
End.
```

Exercice 2

```
Program     Affich_sans_redondance;
Uses     Wincrt;
Var      T : Array [1..20] of Char;
          n, i, j : Integer;
Begin
    Repeat
        Writeln ('Donner un entier'); Readln (n);
    Until  n  in  [3..20] ;
    FOR  i:=1  To  n  Do
           Repeat
               Writeln ('Saisir la case d''ordre ', i);
               Readln (T[i]);
            Until  upcase (T[i]) in ['A'..'Z'] ;
     FOR  i:=1  To  20  Do Write (T[i], ' ');
     Writeln ;
     Writeln ;
     Write (T[1], ' ');
     FOR  i:=2  To  20  Do
            Begin
                j:=i;
                While (j>2) and (T[i]<>T[j-1])  Do   j:=j-1;
                IF T[i] <>T[j-1]  Then   Write (T[i], ' ');
           End;
End.
```

Exercice 3

```
Program  Freq_Lettre ;
Uses      Wincrt ;
Const     n=35 ;
Var        LET : Array [1..n]  of Char ;
            FE : Array ['A'..'Z'] of  Byte ;
            i : Byte ; j : Char;
Begin
   Randomize ;
   FOR   i := 1 To  n  Do
        Begin
         LET[i] := CHR (65+ Random (26)) ;
         Write (LET[i] :2) ;
        End;
   FOR   j := 'A'  To  'Z'  Do   FE[j] := 0 ;
   FOR   i := 1  To   n  Do
         FE[LET[i]] := FE[LET[i]] + 1;

   Writeln ; Writeln;
   FOR   j := 'A' To  'Z'  Do  Write (j:2);
   Writeln ;
   FOR   j := 'A' To  'Z'  Do  Write (FE[j]:2) ;
End.
```

Exercice 4

```
Program   Conversion_base10_base2;
Uses     Wincrt;
Var      rest : Array [1..50] of 0..1;
           n, i, j : Integer;
Begin
   Repeat
      Writeln ('Donner un entier positif');   Readln (n);
   Until  (n > 0);
   i:=0;
   Repeat
       i:=i+1;
       rest[i]:=n mod 2;
       n:=n div 2;
   Until n=0;
   FOR  j:=i  Downto  1  Do    Write (rest[j]);
End.
============ Solution 2 ==================
Program   Conversion_base10_base2;
Uses     Wincrt;
var     n:integer;
(******************************************)
procedure saisir(var n:integer);
begin
   Repeat
      Writeln ('Donner un entier positif');  Readln (n);
   Until  (n > 0);
end;
(*****************************************)
function dec_bin (n:integer):string;
var    chb,chr:string;  r:0..1;
begin
   chb:='';
   Repeat
       r:=n mod 2;
       str(r, chr);
       insert (chr, chb, 1);
       n:=n div 2;
   Until n=0;
   dec_bin:=chb;
end;
(*******************PP********************)
Begin
  Saisir (n);
  writeln ('(',n,')10 = (',dec_bin(n),')2');
End.
```

Exercice 5

```
Program  Conversion_b1_b2;
Uses  Wincrt;
Var    nb, reste : Array [1..50] of 0..15;
       b1, b2, n, i, j, err, nb10 : Integer;
       nch : String;
Begin
   Repeat
     Write ('Base b1 = '); Readln (b1);
     Write ('Base b2 = '); Readln (b2);
   Until  (b1 in [2..16]) and (b2 in [2..16]);

   Writeln ('Donner le nombre à convertir'); Readln (nch);
   n:=Length (nch);
   FOR  i:=1  To  n  Do
      IF  ORD (nch[i]) <65
         Then   VAL (nch[i], nb[i], err)
         Else    nb[i] :=(ORD (nch[i]) - ORD ('A') + 10) ;
   {conversion de la base b1 au décimal}
   nb10 :=0;
   FOR  i:=1  To  n-1  Do  nb10:=(nb10+nb[i])*b1;
   nb10:=nb10+nb[n];

   {conversion de nb10 du décimal à la base b2}
   i:=0;
   Repeat
      i:=i+1;
      reste[i]:=nb10 mod b2;
      nb10:=nb10 div b2;
   Until nb10=0;

   {affichage du résultat}
   FOR  j:=i  Downto  1  Do
      IF  reste[j] < 10
          Then  Write (reste[j])
          Else   Write (CHR (reste[j]-10 + ORD ('A')));
End.
```

Exercice 6

```
Program   Eclater_tab ;
Uses     Wincrt ;
Type     Tab = Array [1..50] of Integer ;
Var      T, TN, TP : Tab ;
         n, i, j, k : Integer ;
Begin
    Repeat
       Writeln ('Saisir un entier') ;
       Readln (n) ;
    Until  (n>=10) and (n<=50) ;
    Writeln (‘Saisir les ', n, ' éléments  de T') ;
    FOR   i:=1  To   n  Do  Readln (T[i]) ;
    j := 0 ; k := 0 ;
    FOR  i := 1 To  n   Do
       IF T[i] < 0   Then   Begin
                               j := j+1 ;
                              TN[j] := T[i] ;
                            End
                      Else   Begin
                               k := k+1 ;
                               TP[k] := T[i] ;
                            End ;
    FOR   i := 1 To   j  Do  Write (TN[i]:4) ;
    Writeln ;
    FOR   i := 1 To   k  Do  Write (TP[i]:4) ;
End.
```

Exercice 7

```
Program   Inverser_tab ;
Uses   Wincrt ;
Type   Tab = Array [1..50] of Integer ;
Var     T : Tab ;
        i, n, aux : Integer ;
Begin
    Repeat
      Writeln ('Saisir un entier') ;
      Readln (n) ;
    Until   n  in [10..50] ;
   Writeln (‘Saisir les ', n, ' éléments  de T') ;
   FOR   i := 1  To   n   Do   Readln (T[i]) ;
   FOR  i := 1 To  (n div 2)  Do
      Begin
         aux := T[i] ;
         T[i] := T[n-i+1] ;
         T[n-i+1] := aux ;
      End ;
    Writeln ; Writeln ;
    Writeln ('Tableau inversé :') ;
    FOR  i := 1 To  n  Do  Write(T[i]:4) ;
End.
```

Exercice 8

```
Program    Regrouper_tab ;
Uses   Wincrt ;
Var     T : Array [1..50] of Integer ;
        i, j, k, n, tmp : Integer ;
Begin
  Repeat
     Write ('N = '); Readln (n) ;
  Until   (n>=10) and (n<=50) ;
  Randomize ;
  FOR  i := 1 To  n    Do
        Begin
            T[i] := -20+Random (41) ;
             Write (T[i]:4) ;
        End;
  k:=0 ;
  FOR  i := 1 To  n  Do
     IF  (T[i] mod 2) = 0
         Then  Begin
                  k := k+1 ;
                  IF  i <> k
                    then begin
                            tmp := T[i] ;
                            FOR  j:=i  Downto   k+1  Do  T[j]:=T[j-1] ;
                            T[k] := tmp ;
                          end;
                End ;
   Writeln ; Writeln ;
   FOR   i := 1 To   n  Do  Write (T[i]:4) ;
End.
```

Exercice 9

```
Program   Min_Max_tab ;
Uses  Wincrt;
Var    T : Array [1..50] of Integer;
       i, min, max, n : Integer;
Begin
    Repeat  Readln (n)  Until  (n>=10) and (n<=50);
    FOR  i:=1  TO  N  DO  Readln (T[i]);
    min:=T[1];
    max:=T[1];
    FOR i:=2  TO  n   DO
       Begin
          IF  T[i]<min  Then  min:=T[i];
          IF  T[i]>max  Then  max:=T[i];
       End;
    Writeln ('Valeur maximale = ', max);
    Writeln ('Valeur minimale = ', min);
End.
```

Exercice 11

```
Program   Ranger_tab ;
Uses   Wincrt ;
Var     T : Array [1..20] of Integer ;
        i,k, n, tmp : Integer ;
Begin
   Repeat
        Write ('N = ');
        Readln (n) ;
   Until (n>=5) and (n<=20) ;

   FOR  i:=1  To  n   Do   readln (T[i]);

   k:=0 ;
   FOR  i:=1  To  n   Do
      IF  (T[i] >= 0)
           Then  Begin
                   k := k+1 ;
                   tmp := T[k] ;
                   T[k]:=T[i] ;
                   T[i] := tmp ;
                 End ;

   FOR  i:=1  To  n   Do Write (T[i]:4) ;
End.
```

Exercice 10

```
Program   Symetri_tab ;
Uses    Wincrt ;
Type    Tab = Array [1..50] of Integer ;
Var     T : Tab ;
        i, j, n : Integer ;
Begin
    Repeat
      Writeln ('Saisir un entier') ;
      Readln (n) ;
    Until   (n>1) and (n mod 2 =0) ;
   Writeln ('Saisir ', n div 2, ' éléments  de T') ;
   FOR   i := 1  To   (n div 2)   Do
        Begin
            Readln (T[2*i-1]) ;
            T[2*i]:= T[2*i-1];
       End;
   FOR  i := 1 To  (n div 2)-1  Do
      Begin
         FOR j:=i+1  To  n-i  Do  T[j] := T[j+1] ;
         T[n-i+1] := T[i] ;
      End ;
    Writeln ('Tableau symétrique :') ;
    FOR  i := 1 To  n  Do  Write (T[i]:4) ;
End.
```

Exercice 12

```
Program     Recherche_Dichotomique_tab ;
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  [10..50];

  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 chercher'); Readln (v);
End;
(************************)
Function  Recherche (v, n : Integer ; T : Tab) : Integer;
Var   d, g, m, pos : Integer;
Begin
   g:=1 ; d:=n ; pos:=0;
   Repeat
     m:=(g+d) div 2 ;
     IF  V=T[m]   Then   pos:=m
                  Else   IF   V>T[m]
                              Then    g:=m+1
                               Else    d:=m-1;
   Until   (pos=m) Or (g>d);
   recherche:= pos ;
End;
(************************)
Begin
   Saisie (T, N, V);
   IF  Recherche (V, N, T) = 0
        Then Writeln (V, ' ne figure pas dans le tableau')
         Else Writeln (V, ' se trouve à la position ',
                                            recherche (V, N, T));
End.
```

Exercice 12

```
Program    Recherche_sequentielle_tab ;
Uses      Wincrt ;
Var       T : Array [1..50] of Integer ;
           i, v, n : Integer ;
Begin
    Repeat
        Write ('N = ');  Readln (n);
    Until  (10<=n) and (n<=50);
    Writeln ('Saisir les ', n, ' éléments de T ');
    FOR  i := 1 To  n   Do   Readln (T[i]) ;
     Writeln ('Donner la valeur à chercher') ; Readln (v) ;
    i := 0 ;
    Repeat
           i :=i+1 ;
    Until  (v=T[i])  Or (i=n) ;
    IF v=T[i]
        Then   Writeln (v, ' se trouve à la position ', i)
        Else   Writeln (v, ' ne figure pas dans le tableau');
End.
```

Exercice 13

```
Program   Regrouper_Tab ;
Uses    Wincrt ;
Var     T : Array [1..20] of Integer ;
        i, j, k, n, tmp : Integer ;
Begin
   Repeat
       Write ('N = ') ; Readln (n) ;
   Until   (n>=2) and (n<=20) ;
   Writeln ('Saisir les éléments de T') ;
   FOR  i:=1  To  n  Do  Readln (T[i]) ;
   FOR  i:=1  To  n-1 Do
       FOR  j:=i+1 To  n  Do
           IF  (T[j] = T[i])  Then  T[j]:=0 ;
   k:=0 ;
   FOR  i:=1 To  n  Do
       IF T[i]<>0 Then Begin
                          k:=k+1 ;
                          IF  T[i] <> T[k] Then  Begin
                                                   tmp := T[k] ;
                                                   T[k] := T[i] ;
                                                   T[i] := tmp ;
                                                 End ;
                       End ;
   FOR  i:=1 To  n  Do  Write(T[i]:3) ;
End.
```

Exercice 14

```
Program    Frequence ;
Uses       Wincrt ;
Const      n=20 ;
Var        T : Array [1..n]  of  1..6 ;
           F : Array [1..6] of  0..20 ;
           i : 1..20 ;
Begin
  Randomize ;
  FOR   i := 1 To  n  Do
         Begin
            T[i] := 1+ Random (6) ;
            Write (T[i] : 2) ;
         End ;

  FOR   i:=1  To  6  Do   F[i] := 0 ;

  FOR   i:=1  To  n  Do
          F[T[i]] := F[T[i]] + 1 ;

  Writeln ;
  FOR   i := 1 To  6  Do   Write (F[i] : 4) ;
End.
```

Exercice 15

```
Program   Moy_Rang;
Uses      Wincrt;
Const     n=30;
Var       A, R : Array [1..n] of Real;
          j, i : Integer;
Begin
   FOR  i:=1  To  n  Do
          Repeat
               Write ('Note élève ', i, ' : ');
                Readln (A[i]);
          Until (A[i]>=0) and (A[i]<=20);

    FOR  i:=1  To  n  Do
          Begin
             R[i]:=1;
             FOR  j:=1 To  n  Do
                  IF A[i]<A[j] Then R[i]:=R[i]+1;
          End;

   Writeln ('Moyens':25, 'Rangs':8);
   FOR   i:=1  To  n  Do
        Writeln (A[i]:25:2, trunc (R[i]):5);
End.
```

Exercice 16

```
Program   Insert_Tab;
Uses      Wincrt ;
Const     n_max = 100;
Var       T : Array [1..n_max]  of  Char;
          c : Char;
          i, k, n : Integer;
Begin
  Repeat
    Writeln ('Donner un entier '); Readln (n);
  Until   (n>=1) and (n<n_max);

  Writeln (‘Saisir les éléments de T’) ;
  FOR  i:=1  To  n  Do   Readln (T[i]);

  Writeln ('Donner le caractère à insérer');
  Readln (c);
  Repeat
     Writeln ('Donner la position d''insertion');
     Readln (k)
  Until   k    in   [1..n];

  {décalage des éléments vers droite}
  FOR  i:= n   Downto  k  Do  T[i+1] := T[i];
  T[k]:=c;
  FOR   i:=1  To  n+1  Do   Write(T[i]:4);
End.
```

Exercice 17

```
Program  Triangle_Pascal;
Uses    Wincrt;
Type    Tab = Array [1..15] of Integer;
Var     T : Tab; N : Integer;
Procedure  init (n : Integer ; Var T : Tab);
Var   i : Integer;
Begin
   T[1]:=1;
   FOR  i:=2  To  n  Do  T[i]:=0;
End;
Procedure  triangle (n : Integer ; Var T : Tab);
Var   i, j : Integer;
Begin
  Writeln (T[1]);
  FOR   i:=2  To   n  Do
        Begin
          FOR  j:=i  Downto  2  Do
                 Begin
                    T[j]:=T[j]+T[j-1];
                    Write (T[j], '  ');
                 End;
          Writeln (T[1]);
        End;
End;
(*****************************************)
Begin
  Repeat
    Writeln ('Donner La Taille Du Triangle : '); Readln (N);
  Until  N  In  [2..15];
  Init (N, T);
  Triangle (N, T);
End.
```

Exercice 17

```
Program  Triangle_Pascal;
Uses    Wincrt;
Type    matrice = Array [1..15, 1..15] of Integer;
Var     T : matrice;  N : Integer;
Procedure triangle (n : Integer ; Var T:matrice);
Var   l, c : Integer;
Begin
   T[1,1]:=1;
   FOR  l:=2  To  n  Do
          Begin
             T[l,1]:=1;
              FOR  c:=2  To  l-1  Do
                         T[l,c]:=T[l-1,c]+T[l-1,c-1];
             T[l,l]:=1;
          End;
End;
Procedure  Afficher (n : Integer ; T:matrice);
Var    l, c : Integer;
Begin
  FOR l:=1 To n Do
     Begin
       FOR c:=1 To l Do  Write (T[l,c], '  ');
       Writeln ;
     End;
End;
(********************************************)
Begin
  Repeat
    Writeln ('Donner La Taille Du Triangle : '); Readln (N);
  Until  N  In  [2..15];
  Triangle (N, T);
  Afficher (N, T);
End.
```

Exercice 18

```
Program  transpose_matrice;
Uses  Wincrt;
Const  Nmax=10;
Type   Mat=Array[1..Nmax,1..Nmax] Of Integer;
Var     M:Mat;
         N:Integer;
Procedure  Saisie (Var N:Integer);
Begin
     Repeat
          Writeln('Donner N :');
          Readln(N);
     Until N In [1..Nmax];
End;
Procedure  remplir (Var M:Mat; n:integer);
var   i, j:integer;
begin
     For i:=1 to n do
          For j:=1 to n do
                begin
                    Writeln('Donner M[',i,',',j,']');
                    readln(M[i,j]);
                 end;
end;
Procedure  Transpose (Var M:Mat;n:integer);
var a ux,i,j:integer;
begin
     For i:=1 to n do
          For j:=1 to i-1 do
               begin
                  aux:=M[i,j];
                  M[i,j]:=M[j,i];
                  M[j,i]:=aux;
               end;
 end;
Procedure  Affiche ( M:Mat; n:integer);
var    i, j:integer;
begin
    For i:=1 to n do
        begin
             For j:=1 to n do   Write( M[i,j],'  ');
             writeln;
      end;
end;
 {Programme Principal}
 Begin
      Saisie (N);
      Remplir (M, N);
      Transpose (M, N);
       Affiche (M, N);
End.
```

Exercice 20

```
Program    TRI_SELECTION ;
Uses        Wincrt ;
Const          n = 20 ;
Type           Tab = Array [1.. n]  of   String ;
Var             T : Tab ;
                 i , j , posmin  :  Integer ;
                tmp : String;
Begin
       Writeln ('Remplir le tableau par ',n,' chaînes :');
       FOR  i := 1  TO  n  DO    Readln (T[i]) ;

       FOR  i := 1  TO  n-1  DO
             Begin
                  posmin := i ;
                  FOR   j := i+1  TO  n    DO
                        IF  T[j] < T[posmin]   Then  posmin := j ;
                  IF  i<> posmin  Then  Begin
                                                tmp := T[i] ;
                                                T[i] := T[posmin] ;
                                                T[posmin] := tmp ;
                                              End ;

              End ;

       Writeln ('Tableau trié :');
       FOR   i := 1  TO    n    DO   Writeln (T[i]) ;
End.
```

Exercice 19

```
Program Calcul_Moyennes ;
Uses Wincrt;
Type    T1=Array[1..40] Of Real;
          T2=Array[1..40] Of Integer;
Var      N,I :Integer;
          Note1,Note2,Note3,Moy:Real;
           V1:T1;
           V2:T2;
(*******************************************)
Function  Rang(V1:T1;N:Integer;Moy:Real):Integer;
Var    I,R:Integer;
Begin
     R:=1;
     For  I:=1 To  N  Do
         If   V1[I]> Moy  Then   R:=R+1;
     Rang:=R;
End;
(*******************************************)
Procedure  Affiche (V1:T1;V2:T2;N:Integer);
Var    I:Integer;
Begin
    Writeln(N°                 Moyenne                Rang');
     For  I:=1  To   N  Do
          Writeln(I,'                ', V1[I]:3:2,'            ',V2[I]);
End;
(*****************P.P*********************)
Begin
    Repeat
          Writeln('Donner Le Nombre Des Élèves');
          Readln(N);
     Until  N In [5..40] ;
     For  I:=1 To  N  Do
       Begin
            Writeln('Donner Les Notes Du ',I,' Éléve');
            Readln(Note1,Note2,Note3);
            V1[I]:=(Note1+2*Note2+2*Note3)/5;
       End;
     For I:= 1 To  N  Do  V2[I] := Rang (V1,N,V1[I]);
     Affiche(V1,V2,N);
End.
```

Exercice 21

```
Program  Intersection_Tab;
Uses    Wincrt;
Type    vect = Array [1..99] of Integer;
Var     T1, T2, inter : vect;
         n, m : Integer;
(*****************************************)
Procedure saisie_int (Var nf : Integer);
Begin
   Repeat
      Write ('N = ');
      Readln (nf);
   Until nf in [3..99];
End;
(*****************************************)
Procedure  remplir_tab (nf : Integer ; Var A:vect);
Var   i, j : Integer;
Begin
    FOR  i:=1  To  nf  Do
       Repeat
          Writeln ('Saisir la case ', i);
          Readln (A[i]);
          j:=1;
          While  A[i] <> A[j] Do j:=j+1;
       Until    i = j ;
End;
(*****************************************)
Procedure  intersection (nf : Integer ; A1, A2:vect;
                                            Var p : Integer ; Var B:vect);
Var     i, j : Integer;
Begin
     p:=0;
     FOR   i:=1  To  nf   Do
           Begin
              j:=0;
              Repeat
                     j:=j+1;
              Until (j=nf) Or (A1[i]=A2[j]);
              IF A1[i]=A2[j]
                     Then  Begin
                                p:=p+1;
                                B[p]:=A1[i];
                              End;
           End;
End;
(*****************************************)
Procedure  affiche_tab (nf : Integer ; A:vect);
Var   i : Integer;
Begin
   FOR  i:=1  To  nf  Do   Write (A[i], '   ');
End;
(****************** P.P. ********************)
Begin
    saisie_int (n);   remplir_tab (n, T1);  remplir_tab( n, T2);
    intersection (n, T1, T2, m, inter);
    affiche_tab (n, T1);  Writeln ; affiche_tab (n, T2);
    Writeln ;
    affiche_tab (m, inter);
End.
```

Exercice 22

```
program  tri_2_criteres;
uses  wincrt;
const n=10;
type  tab=array[1..n] of string;
var   t:tab;
      i,j,pos:integer;
      aux:string;
begin
    writeln('Remplir T :');
    for i:=1 to n do
       repeat
        write('ch = ');
        readln(t[i]);
       until t[i]<>'';
  (****************************)
   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    writeln (t[i]);
end.
```

Exercice 24

```
Program   tri_2_criteres;
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]);
        repeat  write ('couleur = '); readln (c[i]);  until c[i] in ['B','N'];
       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 23

```
Program   tri_bulles_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;

    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 25

```
program   fusion;
uses   wincrt;
type   tab=array [1..20] of integer;
var    v1,v2,v3:tab;
    n,m,c:integer;
(*******************************************)
procedure lecture (var taille:integer);
  begin
     repeat
        readln(taille);
     until taille in [2..20];
  end;
(********************************************)
procedure remplir(var t:tab; taille:integer);
  var  i:integer;
  begin
   for i:= 1 to taille do  readln(t[i]);
  end;
(********************************************)
procedure  trier (taille:integer;var t:tab);
  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;
(********************************************)
procedure fusionner(v1,v2:tab;var v3:tab;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;
(********************************************)
procedure afficher(t:tab; taille:integer);
  var  i:integer;
  begin
    writeln('Tableau fusion :');
    for i:= 1 to taille do
       write(t[i]:4);
  end;
(*********************P.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 26

```
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-s1)*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 27

```
Procedure  Trier  (n:integer ; T:tab;var rang,s:tab);
Var        i,j : integer;
BEGIN
   FOR  i:=1  TO  n  DO s[i]:=1;

   FOR  i:=1  TO  n-1  DO
      FOR j:=i+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  rang[s[i]]:=i;
END;
```

Exercice 28

```
program  long_suite;
uses   wincrt;
const  n=20;
type   tab=array[1..n] of char;
var     t:tab;  max, suite:string;   i:integer;
begin
   for i:=1 to n do  readln(t[i]);
   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;
   writeln (max[1], length(max));
end.
```

Exercice 29

```
Program  symetrique;
uses wincrt;
type    tab=array [1..200] of integer;
var      t:tab;
         n,i:integer;
(***************************************)
   Procedure  saisie(var n:integer;var t:tab);
   var   i:integer;
   begin
       repeat
          write('N = ');
          readln(n);
       until n in [5..200];
       Randomize;
       for i:=1 to n do
           T[i]:=100+Random(900);
   end;
(*************************************)
   Function  verif (x:integer):boolean;
   var ch:string;
   begin
       str(x,ch);
       verif:= ch[1] = ch[3]
   end;
(**************** P.P ****************)
BEGIN
   saisie(n,t);
   writeln('les nombres symétriques de T sont: ');
   for i:= 1 to n do
           if verif(t[i]) then write(t[i]:4);
END.
```

Exercice 30

```
Program  element_manquant ;
uses  wincrt;
type  tab=array[1..20] of integer;
var    t:tab;
        n:integer;
(*************************************)
Procedure  saisie(var n:integer;var t:tab);
var  i:integer;
begin
  repeat
     writeln('Donner le nombre d''éléments N, 2<=n<=20');
     readln(n);
  until n in [2..20];
  repeat
     write('T[1] : ');  readln(T[1]);
  until  T[1]>=0;
  for i:=2 to n do
      repeat
          write('T[',i,'] : ');
          readln(T[i]);
      until  T[i]>=T[i-1];
end;
(*************************************)
Procedure  manque (n:integer;t:tab);
var   x,i,j:integer;
begin
  write('Les entiers manquants sont : ');
  x:=0;
  for i:=2 to n do
     if (T[i]<>T[i-1]+1)
            then   for  j:=(T[i-1]+1)  to  (T[i]-1)  do
                               begin
                                    write(j,' ');
                                    x:=x+1;
                                end;
 write('; leur nombre est : ',x);
 end;
 (**************** P.P ********************)
 begin
   saisie(n,t);
   manque(n,t);
 end.
```

Exercice 31

```
Program  Sequence;
uses  wincrt;
type  tab = array[1..24] of integer;
var     T:tab;
         n,p1,p2:integer;
(**********************************************)
Procedure  Saisie (var n:integer ; var T:tab);
var  i:integer;
begin
  repeat
     write('N = ');
     readln(n);
  until   n in [2..24] ;
  for  i:=1  to  n  do
     Repeat
         write('T[',i,'] = ');
          readln(T[i]);
     Until (T[i]<>0);
end;
(**********************************************)
Procedure  Recherche (n:integer;t:tab;var p1,p2:integer );
var   s,i,j,max:integer;
begin
   max:=1;
   for i:=1 to n-1 do
      begin
        s:=0;
        for j:=i to n do
              begin
                 s:=s+T[j];
                 if (s=0) and (j-i+1>max)
                          then begin
```

Exercice 32

```
Program  El_frequent;
Uses  Wincrt ;
Type    tab1=Array [1..20]  of 0..9 ;
          tab2=Array [0..9] of  0..20 ;
Var     T:tab1;
          F:tab2;
          n:integer;
(***************************************)
Procedure Saisir (var n:integer);
begin
    Repeat
       writeln('Saisir un entier N, (5<=n<=20)');
       readln(n);
    Until n in [5..20];
end;
(***************************************)
Procedure  remplir (n:integer;var t:tab1);
var    i:integer;
begin
   randomize;
   for i:=1 to n do
      begin
         t[i]:=random(10);
         write(t[i]:3);
      end;
   writeln;
end;
(**************************************)
Procedure affiche(n:integer;t:tab1;var f:tab2);
var   i,max:integer;
Begin
   For   i:=0  To  9  Do   F[i] := 0 ;
```

```
                              p1:=i;
                              p2:=j;
                              max:=j-i+1;
                           end;
            end;

      end;
end;
(**********************************************)
Procedure  Affiche (p1,p2:integer ; t:tab);
var  i:integer;
begin
   writeln('La plus longue séquence est :');
   for   i:=p1  to  p2  do
          write(T[i],' ');
end;
(*****************P.P*************************)
BEGIN
  Saisie(n,t);
  Recherche(n,t,p1,p2);
  Affiche(p1,p2,t);
END.
```

```
  For   i:=1  To  n  Do
          F[T[i]] := F[T[i]] + 1 ;

  max:=1;
  For   i := 2 To  9  Do
        if   F[i]>F[max]  then  max:=i;

  writeln(max,', son nombre d''occurrence est ', F[max]);
End;
(*************** P.P *****************)
BEGIN
  saisir(n);
  remplir(n,t);
  affiche(n,t,f);
END.
```

Exercice 33

```
Program  Recherche_ch_tab;
uses   wincrt;
type   tab = array[1..10] of string;
var    T : tab;
        ch, message : string;
        n : integer;
(**************************************************)
Procedure  saisies (var chn:string ; var  m:integer ; var A:tab);
var    i : integer;
begin
   repeat
      write('Donner un entier : ');
      readln(m);
   until  m in [2..10];
   writeln ('Donner les éléments du tableau :');
   for  i:=1  to  m  do
       repeat
           readln(A[i]);
       until   length(A[i]) = m;
   repeat
       write('Donner la chaîne à chercher : ');
       readln(chn);
   until   length(chn) = m;
end;
(**************************************************)
Function  recherche (chn:string ; m:integer ; A:tab) : boolean;
var    i : integer;
       trouve : boolean;
       invchn : string;
   {============================}
function  inverse (chn:string):string;
var    k : integer;
       chinv :  string;
begin
   chinv := '';
   for  k:=1  to  length(chn)  do  chinv := chn[k] + chinv;
   inverse := chinv;
end;
    {==========================}
begin
   invchn := inverse(chn);
   i := 0;
   Repeat
      i:= i+1;
       Trouve := (chn=A[i]) or (invchn=A[i]);
   Until   Trouve  or  (i=m);
   recherche := trouve;
end;
(*************** P.P ***************************)
BEGIN
  saisies(ch, n, T);
  if   recherche (ch, n, T)
    then  message := 'La chaîne ' + ch + ' existe dans le tableau T'
```

Exercice 34

```
Program Exercice34 ;
uses  wincrt;
type   tab=array[1..30] of string[5];
var    t:tab;
        n:byte;
        s:longint;
(****************************************)
Procedure  saisie(var n:byte;var t:tab);
Var   i:byte;
begin
  repeat
    write ('n = ');
    readln (n);
  until   n in [2..30] ;
  writeln('Entrer ', n,' chaînes de 5 caractères au maximum');
  for i:=1 to n do
    repeat
      write ('T[',i,']= ');
      readln(t[i]);
    until  length(t[i]) in [1..5];
end;
(****************************************)
Function Somme(n:byte; t:tab):longint;
var i,j:byte;
   p,s:longint;
begin
  s:=0;
  for i:=1 to n do
    begin
      p:=0;
      for  j:=1  to  length(t[i])  do
        if   t[i][j]  in  ['0'..'9']
               then  p:=p*10+(ord(t[i,j])-ord('0'));
      s:=s+p;
    end;
  somme:=s;
end;
(**************P.P**************************)
BEGIN
  saisie(n,t);
  writeln('La somme est : ',somme(n,t));
END.
```

```
   else   message := 'La chaîne '+ ch + ' n''existe pas dans le
tableau T';
   writeln (message);
END.
```

Exercice 35

```
Program  recherche_major;
uses    wincrt;
type    tab=array [1..25] of integer;
var      t:tab;
         p, n:integer;
(*********************************************)
Procedure  saisie(var n:integer;var t:tab);
var   i:integer;
begin
   repeat
       write('n = ');
       readln(n);
   until n in [5..25];

   for i:=1 to n do  Readln(t[i])
end;
(*********************************************)
Function  major_existe(n:integer;t:tab;var p:integer):boolean;
var   i,j,occ:integer;
begin
  major_existe:=false;
  for i:=1 to n do
     begin
       occ:=0;
       for j:=1 to n do  if  t[i]=t[j]  then  occ:=occ+1;
       if  occ > (n div 2)
           then begin
                   major_existe:=true;
                   p:=i;
                end;
     end;
end;
(********************P.P*********************)
BEGIN
  saisie(n,t);
  if   major_existe(n,t,p)
          then  writeln (t[p],' est majoritaire')
          else  writeln ('pas d''élément majoritaire');
END.
```

Exercice 36

```
Program  grande_somme;
uses  wincrt;
type  tab=array[1..50] of integer;
var    n,d,f:integer;
       t:tab;
(*****************************************)
Procedure  saisies(var n:integer;var t:tab);
var i:integer;
begin
   repeat
       write('n = '); readln(n);
   until n in [5..50];
   for i:=1 to n do
      begin
         write('T[',i,'] = ');
         readln(t[i]);
      end;
end;
(*****************************************)
procedure interval(n:integer;t:tab; var d,f:integer);
var   max,i,j,s:integer;
begin
   d:=1;  f:=1; max:= T[1];
   for  i:=1  to  n do
       begin
         s :=0;
         for j:=i  to n do
           begin
              s:= s + T[j];
              if s > max
                 then begin
                         d:=i;
                         f:=j;
                         max:= s;
                       end;
           end;
      end;
end;
(*****************************************)
Procedure  affiche(n,d,f:integer;t:tab);
var i:integer;
begin
  writeln('La plus grande somme est défini par les valeurs :');
  for  i:=d  to  f  do   write(t[i],' ');
end;
(***************P.P************************)
Begin
   Saisies (N,T);
   Interval (N,T,D,F);
   Affiche (N,D,F,T);
End.
```

Exercice 37

```
PROGRAM  Segmentation ;
uses wincrt ;
type   tab=array[1..20] of integer ;
var    T: tab;    n,i : integer;
(*******************************************)
procedure saisie (var n:integer; var T:tab);
begin
   repeat
       write ('n = ');
       readln (n);
   until   n  in [5..20] ;
   for  i:=1  to  n  do
      BEGIN
          write ('T[',i,'] = ');
          readln (t[i]);
      END;
end;
(*******************************************)
procedure  segmenter (n:integer ; var  t : tab);
var   p,aux,i,j:integer;
begin
  p:=1;
  for  i:=2  to  n  do
      begin
        if   t[i] <= t[p]
             then  begin
                        aux:=t[i];
                        for  j:=I  downto  p+1 do  t[j]:=t[j-1];
                        t[p]:=aux;
                        p:=j;
                     end;
      end;
end;
(************P.P***************************)
begin
   saisie (n,t);
   segmenter (n,t);
   for   i:=1  to  n  do   write(t[i],' ');
end.
```

Exercice 39

```
Program  long_sequence;
uses  wincrt;
type  tab1=array[1..10] of string;
var    t:tab1;
       n,p1:integer;
(********************************************)
Procedure  saisies(var n:integer;var t:tab1);
Var   i,k:integer;
       verif:boolean;
begin
  repeat
     writeln('donner la taille du tableau entre 2 et 10');
     readln(n);
  until n in [2..10];
  for i:=1 to n do
     repeat
        writeln('donner la chaîne n° ',i);
        readln(t[i]);
        k:=0;
        repeat
            k:=k+1;
            verif:= t[i][k] in ['0','1'];
        until (verif=false) or (k=length(t[i]));
     until verif and (length(t[i]) in [2..8]);
end;
(*******************************************)
function  recherche (n : integer ; t : tab1):integer ;
var   ch: string ;
       i, p : integer ;
begin
  ch:='1'; p:=0;
  for  i:=1   to   n   do
     while (pos(ch,t[i])<>0) do
          begin
            ch:=ch+'1';
            p:=i;
          end;
  recherche:=p;
end;
 (*******************P.P******************)
begin
  saisies(n,t);
  p1:=recherche(n,t);
  if   p1<>0
       then  writeln ('plus longue séquence des 1 : ',t[p1])
       else   writeln ('abscence des 1');
end.
```

Exercice 38

```
Program  caracteres_communs;
uses  wincrt;
type   tab1=array[1..10] of string;
        tab2=array['a'..'z'] of integer;
var     t:tab1;
         v:tab2;
         n:integer;
(*******************************************)
Procedure  saisies(var n:integer;var t:tab1);
var   i,k:integer;
       verif:boolean;
begin
  repeat
     writeln('donner la taille du tableau entre 2 et 10');
     readln(n);
  until n in [2..10];

  for i:=1 to n do
     repeat
        writeln('donner la chaîne n° ',i);
        readln(t[i]);
        k:=0;
        repeat
           k:=k+1;
           verif:= upcase(t[i][k]) in ['A'..'Z'];
        until (verif=false) or (k=length(t[i]));
     until verif and (t[i]<>'');
end;
(*****************************************)
Procedure  commun(n:integer;t:tab1 ;var v:tab2);
var   j:char;
       i:integer;
begin
  for j:='a' to 'z' do v[j]:=0;
  for i:=1 to n do
     for j:='a' to 'z' do
        if (pos(j,t[i])<>0) or (pos(upcase(j),t[i])<>0)
            then v[j]:=v[j]+1;
  writeln('Les caractères communs : ');
  for j:='a' to 'z' do
    if v[j]=n then write(j, ' ');
end;
(*****************P.P**********************)
begin
  saisies(n,t);
  commun(n,t,v);
end.
```

Exercice 40

```
Program   primalite;
Uses wincrt;
Type   tab = array [1..400] of integer;
Var    t:tab;
       n:integer;
(**********************************************)
procedure saisie (var n:integer);
begin
  repeat
     writeln ('Donner un entier');
     readln(n);
  until  (20<=n) and (n<=400);
end;
(*********************************************)
procedure recherche (n:integer;var T:tab);
var  i, j, p : integer;
begin
  for i:=1 to n do T[i]:=i;
  p:=2;
  while  (p*p) <= n do
     begin
        j:=p*p;
        while  j<=n  do
            begin
               T[j] :=0;
               j:=j+p;
```

Exercice 42

```
Program    nombres_chanceux_ulam;
Uses wincrt;
Type   tab = array [1..400] of integer;
Var     t:tab;
         n:integer;
(*******************************************)
procedure saisie (var n:integer);
begin
  repeat
     writeln ('Donner un entier');
     readln(n);
  until  (20<=n) and (n<=400);
end;
(********************************************)
procedure recherche (n:integer ; var T:tab);
var   i, j, l, k : integer;
begin
  for i:=1 to n do T[i]:=i;
  l:=1;
  while l<=n do
     begin
        l:=l+1;
        while t[l]=0 do l:=l+1;
        k:=0;
        for j:=1 to n do
            begin
```

```
          end;
       p:=p+1;
     end;
end;
(**********************************************)
procedure affiche (n:integer;t:tab);
var i:integer;
begin
  for i:=2 to n do
        if  T[i]<>0    then   Write (T[i],' ');
end;
(******************** P.P ********************)
Begin
  Saisie(N);
  Recherche (N,T);
  Affiche (N,T);
End.
```

```
             if T[j]<>0  then k:=k+1;
             if k=l  then begin
                            t[j]:=0;
                            k:=0;
                          end;
           end;
      end;
end;
(**********************************************)
procedure affiche (n:integer ; t:tab);
var  i:integer;
begin
  for i:=1 to n do
        if   T[i] <> 0 then Write (t[i],' ');
end;
(******************** P.P ********************)
Begin
  Saisie(N);
  Recherche (N,T);
  Affiche (N,T);
End.
```

Exercice 41

```
Program  nombre_polite;
Uses  wincrt;
var    t, v : array [1..100] of integer;
       k, i, j, a:integer;
begin
  k:=0;
  for  i:=0  to  10  do
     begin
       a:=i;
       for  j:=i+1  to  15  do
         begin
           a:=a+j;
           k:=k+1;
           t[k]:=a;
           write (t[k],' ');
         end;
     end;
  writeln;
  for  i:=1  to  k   do   v[t[i]]:=t[i];

  for  i:=1  to  k  do
      if  v[i]<>0   then   write(v[i],' ');
end.
```

Exercice 43

```
Program Tri_couleur;
uses  wincrt;
Type tab = array[1..100] of char;
VAR   P ,N: integer;
    T:tab;
(*******************************************)
Procedure saisies (var n:integer;var t:tab);
var i:integer;
begin
  repeat
    write('N = ');Readln(n);
  until  (3<=n) and (n<=100);
  for i:=1 to n do
    repeat
            write('T[',i,']= ');
                  readln(t[i]);
          until T[i] in ['B','V','R'];
end;
(*******************************************)
Procedure Ordonner (c: char; var p:integer; n:integer; var t:tab);
var   i:integer;
     temp : char;
begin
  for  i:=p  to  N  do
     if  T[i] = c
         then  begin
                 if P <> i
                    then begin
```

Exercice 44

```
Program premier_absolu;
uses  wincrt;
type  tab=array[1..30] of integer;
var    n,c,d,u,r1,r2,r3,r4,r5,i:integer;
       t:tab;
(********************************************)
Procedure saisies (var n:integer;var t:tab);
var  i:integer;
begin
  repeat
     write('n = ');
     readln(n);
  until  (5<=n) and (n<=30);
  for  i:= 1  to  n  do
    repeat
       readln(t[i]);
    until  (100<=t[i]) and (t[i]<=999);
end;
(**************************************)
Function premier (x:integer):boolean;
var i,d:integer;
begin
  d:=2;
  for i:=2 to (x div 2) do
       if  x mod i =0   then d:=d+1;
  premier:= d=2;
end;
(*****************P.P**********************)
```

```
                                   Temp := T[p];
                                   T[p] := T[i];
                                   T[i] := Temp;
                                  end;
                                  P := P + 1;
                             end;
end;
(**********************************************)
Procedure Afficher (n:integer;t:tab);
var i:integer;
begin
  for i:=1 to n do write (t[i],' ');
end;
(**************P.P*****************************)
Begin
  Saisies(N,T);
  P:=1;
  Ordonner('R',P,N,T);
  Ordonner('B',P,N,T);
  Afficher(N,T);
 End.
```

```
begin
saisies(n,t);
for i:=1 to n do
     if premier(t[i])
         then  begin
                      c:=t[i] div 100;
                      d:=t[i] div 10 mod 10;
                      u:=t[i] mod 10;
                      r1:=c*100+u*10+d;
                     r2:=u*100+d*10+c;
                     r3:=u*100+c*10+d;
                     r4:=d*100+c*10+u;
                     r5:=d*100+u*10+c;
                      if premier(r1) and premier(r2) and premier(r3)
                        and premier(r4)and premier(r5)
                              then   writeln(t[i]);
                 end;
end.
```

Exercice 45

```
Program  sequence;
uses   wincrt;
type   tab=array[1..20] of integer;
var     n:integer; T:tab;
(*****************************************)
procedure saisies(var t:tab; var n:integer);
var    i:integer;
function premier(x:integer):boolean;
var   nb,i:integer;
begin
    nb:=2;
    for i:=2 to x div 2 do
            if (x mod i =0)   then nb:=nb+1;
    premier:=(nb=2);
end;
begin
    repeat
      write('n = ');readln(n);
    until (4<n)and(n<=20);
    randomize;
    for i:=1 to n do
         begin
             repeat
               T[i]:=2+random(98);
             until (premier(T[i]));
             write(t[i],' ');
         end;
    writeln;
end;
(***********************************************)
procedure affiche_seq(t:tab;n:integer );
var   i,nb:integer;
begin
   nb:=1;
   write(t[1],' ');
   for i:=2 to n do
        if T[i] >T[i-1]   then write(t[i],' ')
                              else begin
                                      writeln;
                                      nb:=nb+1;
                                      write(t[i],' ');
                                   end;
   writeln;
   writeln('Le nombre de séquences est : ',nb);
end;
(*********************P.P*********************)
Begin
     Saisies(T,N);
     Affiche_Seq(T,N);
End.
```

Exercice 46

```
Program  Porte_bonheur ;
Uses     Wincrt;
Type     Tab=Array ['A'..'Z'] Of Integer;
Var      T:Tab;
          P,N:Integer;
(***********************************************)
Procedure  Saisie (Var P,N:Integer);
Begin
       Repeat
           Write('P = '); Readln(P);
       Until  (P In [1..10]);
       Repeat
            Write('N = ');  ReadIn(N);
      Until  (N In [4..19]);
End;
(***********************************************)
Procedure   Tirage (P,N:Integer;Var T:Tab);
Var    I:Integer;
        Let:Char;
        Ch:String;
Begin
       For  Let:='A'   To   'Z'  Do   T[Let]:=0;
       Randomize;
       For   I:=1  To  N   Do
            Begin
               Ch:='';
               Repeat
                    Let:=Chr(65+Random(26));
                    Ch:=Ch+Let;
                    T[Let]:=T[Let]+1;
               Until Length(Ch)=P;
               Writeln(Ch);
           End;
End;
(**********************************************)
Procedure   Affiche(T:Tab);
Var       K:Char;
           Max:Integer;
 Begin
      Max:=T['A'];
      For   K:='A'   To   'Z'   Do
             If  T[K]>Max  Then  Max:=T[K];
       Writeln('Les Lettres Porte-Bonheur Sont: ');
       For   K:='A'   To   'Z'    Do
            If   T[K]=Max   Then   Write (K,' ');
End;
(*******************P/P********************)
Begin
   Saisie(P,N);
   Tirage(P,N,T);
   Affiche(T);
End.
```

Exercice 47

```
Program Nbr_Zigzag;
Uses   Wincrt;
Type   Tab=Array[1..25] Of  Integer;
Var     T:Tab;
        N,I:Integer;
(*****************************************)
Procedure Saisie(Var N:Integer;Var T:Tab);
Var I:Integer;
Begin
    Repeat
         Readln(N);
    Until  (5<=N) And (N<=25) ;

    For  I:=1  To  N  Do
         Repeat
             Readln(T[I]);
         Until  (100<=T[I]) And (T[I]<=Maxint) ;
End;
(****************************************)
Function   Zigzag(Nb:Integer) :Boolean ;
Var     K:Integer;
        Ch:String;
        Verif:Boolean;
Begin
   Str(Nb,Ch);
   K:=1;
   Repeat
       K:=K+1;
       Verif := ((Ch[K-1]<Ch[K])And(Ch[K]>Ch[K+1])) Or
                ((Ch[K-1]>Ch[K])And(Ch[K]<Ch[K+1]));
   Until  Not Verif   Or (K=Length(Ch)-1);
   Zigzag :=Verif ;
End;
(********************P.P*******************)
Begin
  Saisie(N,T);
  For  I:=1  To  N   Do
     If  Zigzag(T[I])   Then  Writeln(T[I]);
End.
```

Exercice 48

```
Program  Suite_Geometrique;
Uses   Wincrt;
Type   Tab=Array [1..20] Of Integer;
Var     N:Integer;
        T:Tab;
(*******************************************)
   Procedure Lecture(N:Integer;Var T:Tab);
   Var I:Integer;
   Begin
       Randomize;
       For I:=1 To N Do
           T[I]:=1+Random(100);
   End;
(**********************************************)
   Function Geometrique(N:Integer;T:Tab):Boolean;
   Var    I:Integer;
          Q:Real;
          Geo:Boolean;
   Begin
       Q:=T[2]/T[1];
       I:=3;
       Repeat
            If  T[I]/T[I-1]=Q Then
              Begin
                  Geo:= True;
                  I:=I+1;
              End
              Else Geo:=False;
       Until (I>N) Or (Geo=False);
       Geometrique:=Geo;
   End;
(*******************************************)
   Procedure Affiche(N:Integer;T:Tab);
   Var I:Integer;
   Begin
       For I:= 1 To N Do
          Write(T[I]:6);
       Writeln;
   End;
(*****************P.P************************)
Begin
 Repeat
    Write('Donner Un Entier N: ');
    Readln(N);
 Until N In [2..15];
 Lecture(N,T);
 Write('La Suite Est : ');
 Affiche(N,T);
 If  Geometrique(N,T)
    Then Writeln('C''est Une Progression Géométrique')
    Else Writeln('Ce N''est Pas  Une Suite Géométrique')
End.
```

Exercice 50

```
Program  Exercice50;
Uses     Wincrt;
Type    Tab=Array [1..39] Of Integer;
Var      P,S,N:Integer;
         T:Tab;
(*******************************************)
   Procedure Saisies (Var N:Integer; Var T:Tab);
   Var  I:Integer;
   Begin
       Repeat
            Write('N= ');  Readln(N);
       Until N In [2..39];
       For I:= 1 To N Do
          Begin
              Write('T[',I,']= ');  Readln(T[I]);
          End;
   End;
(*******************************************)
  Procedure Elt_Tab(Var P,S:Integer);
  Function Indice(X:Integer):Integer;
   Var  I:Integer;
   Begin
       I:=0;
       Repeat
            I:=I+1
       Until (X=T[I]) Or (I>N);
       If I>N Then Indice:=0 Else Indice:=I;
   End;
   Begin
       Repeat
            Write('P= '); Readln(P);
            Write('S= '); Readln(S);
       Until (Indice(P)<>0) And (Indice(S)<>0)
                      And (Indice(S)>Indice(P)+1);
   End;
(*******************************************)
   Procedure Affichage(T:Tab;N,P,S:Integer);
   Var  I:Integer;
   Begin
       I:=0;
       Repeat
             I:=I+1;
             If T[I]=P Then
               Repeat
                   Write(T[I+1],'   ');
                   I:=I+1;
               Until T[I+1]=S
       Until  I>N
   End;
 (*****************P.P*********************)
Begin
    Saisies(N,T);
    Elt_Tab(P,S);
    Affichage(T,N,P,S);
End.
```

Exercice 51

```
Program Fusion_tab;
Uses  Wincrt;
Type   Tab=Array[1..20] Of Integer;
Var     T1,T2,T:Tab;
         I,N:Integer;
(*******************************************)
   Procedure Saisie(N:Integer;Var T:Tab);
   Var I:Integer;
   Begin
       For I:=1 To N Do
           T[I]:=10+Random(90);
   End;
 (*******************************************)
   Procedure Fusion(N:Integer;T1,T2:Tab;Var T:Tab);
   Var I:Integer;

   Function Insertion(X,Y:Integer):Integer;
   Var   D1,D2,U1,U2:Integer;
   Begin
       D1:=X Div 10;
       D2:=Y Div 10;
       U1:= X Mod 10;
       U2:=Y Mod 10;
       Insertion:=D1*1000+D2*100+U1*10+U2;
   End;

   Begin
       For  I:= 1  To  N  Do
           If  T1[I]>T2[I]
               Then T[I]:=Insertion(T1[I],T2[I])
               Else  T[I]:=Insertion(T2[I],T1[I]) ;
   End;
(*******************************************)
   Procedure Affiche(N:Integer;T:Tab);
   Var I:Integer;
   Begin
       For I:= 1 To N Do   Write(T[I]:6);
       Writeln;
   End;
(*****************P.P************************)
Begin
   Repeat
     Write('Donner Un Entier N: ');
     Readln(N);
   Until N In [2..15];
   Writeln('Le 1er  Tableau: ');
   Randomize;
   Saisie(N,T1);  Affiche(N,T1);
   Writeln('Le 2ème Tableau: ');
   Saisie(N,T2);  Affiche(N,T2);
    Fusion(N,T1,T2,T);
    Writeln('Le Tableau Final: ');
    Affiche(N,T);
End.
```

Exercice 52

```
Program Classement;
Uses Wincrt;
Type
  tab1 =  Array[1..30] Of string;
  tab2 =  Array[1..30] Of Real;
Var
  nom:  tab1;
  moy:  tab2;
  n:  Integer;
(************************************************)
Procedure Saisies(Var n:Integer;Var nom:tab1;Var moy:tab2);
Var   i,j:  Integer;
      verif:  Boolean;
Begin
  Repeat
    write('Nombre d''élèves : ');
    Readln(n);
  Until n In [5..30];
```

Exercice 53

```
Program  Pluslong_sequence;
Uses  wincrt;
Type
  tab =  Array[1..50] Of Integer;
Var
  t:  tab;
  n,i,lmax,dm:  Integer;
(*******************************************)
Procedure Saisie(Var n:Integer; Var t:Tab);
Var i:  Integer;
Begin
  Repeat
    write('n = ');
    Readln(n);
  Until (5<=n) And (n<=50);
  For  i:=1 To  n  Do
    Repeat
       Readln(t[i]);
  For i:=1 To n Do
    Begin
      Repeat
        Write('Nom[',i,']= ');
        Readln(nom[i]);
        j := 0;
        Repeat
          j := j+1;
          verif := Upcase (nom[i,j]) In ['A'..'Z',' '];
        Until (verif=False) Or (j=Length(nom[i]));
      Until verif=True;
      Repeat
        Write('Moy[',i,']= ');
        Readln(moy[i]);
      Until (0<=moy[i])And (moy[i]<=20);
    End;
End;
(*********************************************)

Procedure Tri(N:Integer;Var nom:tab1;Var moy:tab2);
Var
  i,nbpermut:  Integer;
  aux1:  string;
  aux2:  Real;
Begin
  Repeat
    nbpermut := 0;
    For i:=1 To n-1 Do
      If moy[i]<moy[i+1] Then
        Begin
          aux2 := moy[i];
          moy[i] := moy[i+1];
          moy[i+1] := aux2;
          aux1 := nom[i];
          nom[i] := nom[i+1];
          nom[i+1] := aux1;
          nbpermut := nbpermut+1;
        End;
  Until (nbpermut=0);
End;
(*********************************************)
Procedure Affiche(n:Integer;nom:tab1;moy:tab2);
Var
  i,r:  Integer;
Begin
  Writeln('Le classement est : ');
  r := 1;
  write('Rang 1 : ',nom[1]);
  For i:=2 To n Do
    If moy[i]=moy[i-1] Then write(', ',nom[i])
    Else
      Begin
        Writeln;
        r := r+1;
        write('Rang ',r,' : ',nom[i]);
      End;
End;
(********************P.P********************)
Begin
  Saisies(n,nom,moy);
  Tri(n,nom,moy);
  Affiche(n,nom,moy);
End.
```

```
    Until  t[i] In [0..9];
End;
(*********************************************)
Procedure recherche(t:Tab;Var lmax,dm:Integer);
Var  i,nb:  Integer;
Begin
  nb := 1;
  lmax := 1;
  dm := 1;
  For  i:=2 To n Do
    If  t[i]>t[i-1]
       Then  Begin
                 nb := nb+1;
                 If nb>lmax
                    Then  Begin
                              lmax := nb;
                              dm := i-nb+1;
                           End;
             End
       Else  nb := 1;
End;
(************************P.P.********************)
Begin
  Saisie(n,t);
  recherche(t,lmax,dm);
  Writeln('La longueur de la plus longue suite croissante =',lmax);
  For  i:=dm  To  (dm+lmax-1) Do  write(t[i]:2);
End.
```

Exercice 54

```
Program  Exercice54;
uses   wincrt;
type   tab=array[1..99] of integer;
var     t:tab;
        n:integer;
(**********************************************)
Procedure  saisies (var n:integer;var t:tab);
var  i:integer;
begin
   repeat
     write('n = ');
     readln(n);
   until  (n in [3..99]) and (n mod 3 =0);

   Write ('Elément1 = '); Readln (T[1]);
   for  i:=2  to  n  do
       repeat
           write ('Elément',i,' = ');
           readln (T[i])
       until T[i]>T[i-1];
end;
(*******************************************)
Procedure remplir (n:integer ; var t:tab);
Var  i,j : integer;
begin
  j:=1;
  for  i:=1  to  (n div 3)  do
     begin
       t[i]:= t[j]+t[j+1]+t[j+2];
       j:=j+3;
     end;
  for i:=(n div 3)+1 to  n  do t[j]:=0;
 end;

(*******************************************)
Procedure   afficher (T:tab ; n:integer);
var   i:integer;
begin
   for   i:=1  to  n  do  write(T[i]:5);
end;
(*************** P.P. **********************)
BEGIN
  saisies (n,t);
  remplir (n,t);
  writeln ('Tableau résultat :');
  afficher (t,n);
END.
```

Solution2

```
Procedure   remplir (n:integer ; var t:tab);
Var   i, j, som : integer;
begin
  for   i:=1   to   (n div 3)   do
       begin
           som:=0;
           for  j:=(3*i-2)  to  (3*i)   do  som:=som+t[j];
           t[i]:=som;
       end;
  for   j:=i+1  to  n  do   t[j]:=0;
end;
```

Exercice 55

```
Program inverse_bloc;
Uses Wincrt;
Type
  tab =  Array[1..100] Of Char;
Var
  t:  tab;
  n,d:  Integer;
(**********************************************)
Procedure saisies (Var n,d:Integer;Var t:tab);
Var
  i:  Integer;
Begin
  Repeat
    write('n = ');
    Readln(n);
  Until (n In [4..100]) And (n Mod 4 =0);
  Repeat
    write('d = ');
    Readln(d);
  Until (n Mod d =0)And(d<>n);
  For i:=1 To n Do
    Repeat
      write ('Elément',i,' = ');
      Readln (T[i]);
    Until t[i] In ['A'..'Z'];
End;
(*******************************************)
Procedure inverser (n,d:Integer ; Var t:tab);
Var
  i,k,db,fb,mb :  Integer;
  tmp:  Char;
Begin
  For i:=1 To (n Div d) Do
    Begin
      db := d*(i-1)+1;        {début bloc}
      fb := d*i;              {fin bloc}
      mb := (db+fb) Div 2;    {milieu bloc}
      For k:=db To mb Do
        Begin
          tmp := t[k];
          t[k] := t[fb-k+db];
          t[fb-k+db] := tmp;
        End;
    End;
End;
(*******************************************)
Procedure afficher (T:tab ; n:Integer);
Var
  i:  Integer;
Begin
  For i:=1 To n Do  write(T[i]:5);
End;
(*************** P.P. **********************)
BEGIN
  saisies (n,d,t);
  inverser (n,d,t);
  Writeln ('Tableau résultat :');
  afficher (t,n);
END.
```

LES CHAINES DE CARACTERES

Exercice 2

```
Program   Palindrome;
Uses    Wincrt ;
Var      ch, inv : String;
           i : Integer;
Begin
   Writeln ('Saisir une chaîne'); Readln (ch);
   inv := '' ;
   FOR   i := Length (ch)  Downto  1  Do
           inv := inv + ch[i];
   IF ch = inv
       Then   Writeln (ch, ' est palindrome')
       Else    Writeln (ch, ' n''est pas palindrome');
End.
```

Exercice 3

```
Program   chaine_inverse;
Uses  wincrt;
Var   ch : string;
(**************************************)
Function   miroir (ch:string):string;
var   i, l : integer;  c : char;
begin
   l:=length(ch);
   for  i:=1  to  l div 2 do
      begin
         c:=ch[i];
         ch[i]:=ch[l-i+1];
         ch[l-i+1]:=c;
       end;
   miroir:=ch;
end;
(***************** P.P *****************)
BEGIN
   write('ch = ');readln(ch);
   writeln('l''inverse de ',ch,' est : ', miroir(ch));
END.
============== solution 2 ==============
Function   miroir (ch:string) : string;
var   i, l : integer;  mirch : string;
begin
   l:=length(ch);
   mirch:=ch;
   for  i:=1  to  l  do   mirch[i] := ch[l-i+1];
   miroir:=mirch;
end;
```

Exercice 4

```
Program   Chaine_Majus_Minus;
Uses   Wincrt;
Var      ch : String;
          i : Integer;
Begin
    Writeln ('Saisir une chaîne de caractères'); Readln (ch);
    FOR   i:=1   To  Length (ch)   Do
            IF  ch[i]  in ['a'..'z']
                  Then  ch[i]:=ch[i]
                  Else   ch[i]:=CHR (ORD (ch[i]) + 32);
    Writeln (ch);
    Writeln ;
    FOR   i:=1   To  Length (ch)  Do  ch[i]:=upcase (ch[i]);
    Writeln (ch);
End.
```

Exercice 5

```
Program    Chaine1 ;
Uses    Wincrt ;
Var      i : Integer ;  ch: String ;
          test : Boolean;
Begin
   Repeat
      Writeln ('Donner un mot en majuscule') ;  Readln (ch) ;
      test:=True;
      i:=0 ;
      Repeat
           i:=i+1;
           IF Not(ch[i] in ['A'..'Z'])  Then  test:=False;
      Until (test=False) Or (i=Length (ch));
   Until   test=True ;
   FOR   i := 1  To  Length (ch)    Do
           Writeln (COPY (ch, 1, i));
End.
```

Exercice 5

```
Program    Chaine2;
Uses    Wincrt;
Var      ch : String;   i, lg : Integer;
           test : Boolean;
Begin
  Repeat
        Writeln ('Donner un mot en majuscule') ; Readln (ch) ;
         test:=True;  i:=0 ;
         Repeat
                i:=i+1;
                IF Not (ch[i] in ['A'..'Z'])  Then  test:=False;
          Until  (test=False) Or (i=Length (ch));
  Until   test=True ;
  lg:=Length (ch);
  FOR   i:=1  To  lg  Do
        Writeln (COPY (ch, 1, i), COPY (ch, lg-i+1, i));
End.
```

Exercice 7

```
Program   Renverser_ch ;
Uses      Wincrt ;
Var        p : Integer ;
           chr, chd : String ;
Begin
   Writeln ('Saisir une phrase') ;  Readln (chd) ;
   chr := '' ;
   p := POS(' ', chd) ;
   While   p <> 0   Do
       Begin
         chr := ' ' + COPY (chd, 1, p-1) + chr ;
         DELETE (chd, 1, p) ;
         p := POS (' ', chd) ;
      End ;
   chr := chd + chr ;
   Writeln ('Phrase renversée est : ', chr) ;
End.
```

Exercice 6

```
Program   Espace_superflus ;
Uses      Wincrt ;
Var       ch : String;
           i, p : Integer ;
Begin
  Writeln ('Donner une chaîne'); Readln (ch);
  Repeat
       P := POS ('  ', ch);    {position de 2 espaces dans ch}
       IF  p<>0  Then  DELETE (ch, p, 1);
  Until  p=0 ;
  IF  ch[1]=' '
        Then  DELETE (ch, 1, 1);
  IF  ch[Length(ch)]=' '
          Then  DELETE (ch, Length (ch), 1);

  Writeln ('La chaîne devient : ', ch);
End.
```

Exercice 8

Exercice 9

```
Program  Occurence_car;
Uses   Wincrt;
Var    ch : String;
       i, j, n : Integer;
Begin
  Writeln ('Saisir une chaîne'); Readln (ch);
  FOR  i:=1   To  Length (ch)   Do
    Begin
      n:=0;
      FOR  j:=1 To  Length (ch)  Do
        IF ch[i]=ch[j]
             Then n:=n+1;
      IF i = POS (ch[i], ch)
         Then  Writeln ('Occurrence de ', ch[i], ' = ', n);
    End;
End.
```

```
Program   Occurrence_mot ;
uses   wincrt;
var     ch, mot : string;  nb, i, k: integer;
begin
   repeat
      writeln ('saisir un texte'); readln (ch);
    until   length (ch)>20;
    writeln ('saisir un mot'); readln (mot);
    k:=length (mot); nb:=0; i:=1;
    repeat
        if  (ch[i] = mot[1]) and (mot = copy (ch, i, k))
             then  begin
                       nb:=nb+1;
                       i:=i+k;
                    end
              else  i:=i+1;
     until   i>length(ch);
     writeln (mot, ' figure dans le texte ', nb, ' fois');
end.
```

Exercice 10

```
Program   Sans_Redondance ;
Uses    Wincrt;
Var      ch1, ch2, ch3, aux : String;  i : integer;
Begin
   Write('Chaîne 1 = ');Readln (ch1);
   Write('Chaîne 2 = ');Readln (ch2);
   if length(ch1)>length(ch2)
     then begin
          aux:=ch1;
          ch1:=ch2;
          ch2:=aux;
        end;
  ch3:='';
  FOR i:=1To Length(ch1) Do
    IF (POS(ch1[i],ch2)<>0)and(POS(ch1[i],ch3)=0)
        Then ch3:=ch3+ch1[i];
  Writeln(ch3);
End.
```

Exercice 11

```
Program    Aerer_ch ;
Uses       Wincrt ;
Var        k : Byte ;
           ch : String ;
Begin
   Writeln ('Saisir une chaîne') ;
   Readln (ch) ;

   k := 0 ;
   repeat
         k:=k+2;
         Insert (' ', ch, k) ;
    Until   k = length(ch)-1;

   Writeln ('Chaîne aérée = ', ch) ;
End.
```

Exercice 12

```
Program  Anagrammes;
uses   wincrt;
var     mot1, mot2 : string;
(**********************************)
procedure saisie_ch (var m1, m2 : string);
begin
    repeat
       writeln ('donner deux mots : ');
       readln (m1);
       readln (m2);
    until (m1 > '') and (m2 > '');
end;
(**********************************)
function  trie (mot : string) : string;
var    i, j, n : integer;
procedure permut (var a, b : char);
var aux : char;
begin
    aux:=a; a:=b; b:=aux;
end;
begin
      n:=length (mot);
      for  i:=1  to  n-1  do
            for  j:=i+1  to  n  do
                  if mot[i]>mot[j]
                         then  permut (mot[i], mot[j]);
      trie := mot;
end;
(**************************************)
begin
    saisie_ch (mot1, mot2);
    if  trie (mot1) = trie (mot2)
      then  writeln (mot2, ' est une anagramme de ', mot1)
      else   writeln (mot2, ' n''est pas une anagramme de ', mot1);
end.
```

Exercice 12

```
Program  anagrammes;
uses  wincrt;
var   mot1, mot2:string;
(*****************************************)
procedure saisie_ch(var m1,m2:string);
begin
    repeat
       writeln ('donner deux mots : ');
       readln (m1);
       readln(m2);
    until (m1 > '') and (m2 > '');
end;
(****************************************)
function  anagram (mot1,mot2:string):boolean;
var   p:integer;
begin
       anagram:=false;
       repeat
          p:=pos(mot1[1],mot2);
          if p>0
             then begin
                    delete(mot1,1,1);
                    delete(mot2,p,1);
                   end;
       until (p=0) or (mot1='');
       if (mot1='') and (mot2='') then anagram:=true;
end;
(*********************************************)
begin
    saisie_ch (mot1, mot2);
    if anagram (mot1, mot2)
       then writeln (mot1, ' est une anagramme de ', mot2)
       else writeln (mot1, ' n''est pas une anagramme de ', mot2);
end.
```

Exercice 13

Exercice 14

```
Program    Pos_ch;
Uses    Wincrt;
Var       ch1, ch2 : String;
          i, p : Integer;
Begin
    Write ('ch1 = '); Readln (ch1);
    Write ('ch2 = '); Readln (ch2);
    i:=1 ;
    p:=0;
    Repeat
        IF  ch1 = COPY (ch2, i, Length (ch1))    Then    p:=i;
        i:=i+1;
    Until   (Length (ch2)-i < Length (ch1)) Or (p<>0) ;
    Writeln ('La chaîne ', ch1, ' occupe la position ', p,
                   ' dans la chaîne ', ch2);
End.
```

```
Program    Copie_ch;
Uses    Wincrt;
Var       ch1, chr : String;
          i, p, n, l : Integer;
Begin
  Write ('ch1 = '); Readln (ch1);
  l:=Length (ch1);
  Repeat
     Write ('Position = '); Readln (p);
     Write ('Nbre caractères = '); Readln (n);
  Until  (p in [1..l]) and (n in [1..l]) and (p+n<=l+1);
  chr:='';
  FOR  i:=p  To p+n-1  Do
          chr:=chr+ch1 [i];
  Writeln ('La chaîne copiée est : ', chr);
End.
```

Exercice 15

```
Program   Jeu_pendu ;
Uses   Wincrt;
type    chain=String [50];
Var       se, ma:chain;
          rep, let : Char;
(****************************************)
Function   controle_saisie (se:chain) : Boolean;
Var  i : Integer;
    r : Boolean;
Begin
   r:=True;    i:=0 ;
   Repeat
     i:=i+1;
     IF Not (se[i] in ['A'..'Z'])  Then  r:=False;
   Until (r=False) Or (i=Length (se));
   controle_saisie := r;
End;
(****************************************)
Procedure masquer (se:chain; Var ma : chain);
Var    i : Integer;
Begin
   ma:=se;
   FOR   i:= 2   To  (Length (se)-1)  Do   ma[i]:='-';
End;
(****************************************)
Procedure  saisie_let (Var let : Char);
Begin
  Writeln ('Donner une lettre ');
  Readln (let);
  let:=upcase (let);
End;
(****************************************)
Procedure  devoiler (se : chain; let : Char; Var ma : chain);
Var    i : Integer;    r : Boolean;
Begin
    r:=False;
    FOR   i:=2  To  (Length (se)-1)   Do
           IF se[i]=let
                 Then Begin
                          ma[i]:=let;
                          r:=True;
                      End;
    IF  r=False   Then  Writeln ('Echec');
End;
(*****************************************)
Procedure partie_jeu (se, ma : chain ; let : Char);
Var   nb : Integer;
Begin
       ClrScr;
       Writeln (ma);
       nb:=0;
       Repeat
          nb:=nb+1;
          saisie_let (let);
          devoiler (se, let, ma);
          Writeln (ma);
       Until (nb=Length (se)) Or (ma=se);
       IF ma=se
```

Exercice 16

```
Program   Chaines_inverses;
Uses    Wincrt;
Type    Tab=Array [1..100] of String [50];
Var       p, q : Tab;
          i, n : Integer;
(**********************************)
Procedure   saisie_entier (Var m : Integer);
Begin
     Repeat
         Writeln ('Donner un entier');   Readln (m);
     Until  (1<m) and (m<100);
End;
(*************************************)
Procedure   saisie_tab (m : Integer; Var T : Tab);
Var    i, j : Integer;    test : Boolean;
Begin
   FOR  i:=1  To  m  Do
      Repeat
          Writeln ('Donner l''élément d''ordre ', i);  Readln (T[i]);
          j:=0; test:=True;
          While (test=True) and (j<Length (T[i])) Do
             Begin
                j:=j+1;
                IF Not (T [i, j] in ['0'..'9']) Then  test:=False;
             End;
      Until  (test=True) and (T[i] <>'');
End;
(****************************************)
Function  inverse (ch : String) : String;
Var  i : Integer;   chinv : String;
Begin
   chinv:='';
   FOR  i:=Length (ch) Downto  1  Do
                 chinv:=chinv+ch[i];
   inverse:=chinv;
End;
(**********************************)
Procedure  Affiche_tab (m : Integer; T : Tab);
Var  i : Integer;
Begin
   FOR  i:=1  To  m  Do  Write (T[i], ' ');
End;
(**************** P.P *******************)
Begin
   saisie_entier (n);
   saisie_tab (n, p);
   FOR  i:=1  To  n  Do
           q[i] := inverse (p[i]);
   affiche_tab (n, q);
End.
```

```
        Then Writeln ('Bravo, trouvé en ', nb, ' propositions')
        Else Writeln ('Perdu le mot à deviner est ', se);
End;
(****************** P. P *****************)
Begin
  Writeln ('Le jeu du pendu'); Writeln ;
  Repeat
      Repeat
        Write ('Rentrez un mot secret en majuscule : ');
        Readln (se);
      Until  controle_saisie (se);
      masquer (se, ma);
      partie_jeu (se, ma, let);
      Write ('Voulez-vous rejouer (o/n) ? ');
      Readln (rep);
  Until  Upcase (rep) = 'N';
End.
```

Exercice 17

```
Program   conversion_base1_base2;
uses   wincrt;
var     b1, b2:integer;
        nch:string;
(*********************************************)
Procedure saisie_base (var b1,b2:integer);
begin
   repeat
      write('Base b1 = ');readln(b1);
      write('Base b2 = ');readln(b2);
   until  (b1 in [2..16]) and (b2 in [2..16]) and (b1<>b2);
end;
(*******************************************)
Procedure  saisie_nombre (var  nch:string; b1:integer);
Const  chb='0123456789ABCDEF';
Var    i:integer;
       test:boolean;
begin
  repeat
     writeln ('Donner un nombre en base ', b1);
     readln(nch);
     test:=true;
     for i:=1 to length(nch) do
        if (pos(nch[i],chb)>b1) or (pos(nch[i],chb)=0)
                then test:=false;
  until test=true;
end;
```

```
(**** Conversion de la base b1 vers base 10 **********)
Function  Conv_b1_10 (nch:string; b1:integer) : longint;
var   err,i,n:integer;
      dec,puiss:longint;
begin
  dec:=0; puiss:=1;
  for  i:=length(nch)  downto  1  do
      begin
        if nch[i] in ['0'..'9']
                then   Val(nch[i], n, err)
                else   n:=ord(nch[i])-55;
        dec:=dec+n*puiss;
        puiss:=puiss*b1;
      end;
  conv_b1_10:=dec;
end;
(**** Conversion de la base 10 vers la base b2 *********)
Function  Conv_10_b2 (nd:longint; b2:integer) : string;
var  ch1, chb2:string;
      r:integer;
begin
  chb2:='';
  repeat
    r:= nd mod  b2;
    if r in [0..9]  then  str(r,ch1)
                    else  ch1:= chr(55+r);
    chb2:=ch1+chb2;        (*insert (ch1,chb2,1)*)
    nd:= nd div b2 ;
  until  (nd = 0);
  conv_10_b2:=chb2;
end;
(*************** P.P ***************************)
Begin
 saisie_base(b1,b2);
 saisie_nombre(nch,b1);
 writeln('(',nch,')',b1,' = (',
             conv_10_b2 (conv_b1_10 (nch, b1), b2), ')', b2);
End.
```

Exercice 18

```
Program   Nbre_Rond;
Uses    Wincrt;
Var     reste, c : String;
        n, m, i, j, n1, n0 : Integer;
Begin
   FOR  i:=1  To  1000  Do
    Begin
       n:=i;
       m:=i;
       reste:='';
       Repeat
          STR (n mod 2, c);
          reste:= c+reste;
          n:=n div 2;
      Until n=0;
```

```
        n1:=0; n0:=0;
        FOR  j :=1  To  Length (reste)  Do
           Begin
              IF  reste[j]='1'  Then   n1:=n1+1;
              IF  reste[j]='0'  Then   n0:=n0+1;
           End;
        IF   n1=n0  Then   Writeln (m, ' est ROND')
                    Else   Writeln (m, ' n''est pas ROND');
        Readln ;
    End;
End.
```

Exercice 22

Exercice 21

```
Program  sablier;
uses  wincrt;
var   esp,ch:string;
(*******************************************)
Procedure affiche_bas(var esp:string;ch:string);
var   ch2:string;   n,i:integer;
begin
   esp:='';n:=0;i:=1;
   writeln(ch);
   repeat
      esp:=' '+esp;
      n:=n+2;i:=i+1;
      ch2:=esp+copy(ch, i, length(ch)-n);
      writeln(ch2);
   until length(copy(ch, i, length(ch)-n))=1;
end;
(******************************************)
Procedure affiche_haut(esp,ch:string);
Var  i,n:integer;  ch2:string;
begin
   i:=-1;n:=1;
   repeat
      n:=n+2;i:=i+1;
      delete(esp,1,1);
      ch2:=esp+copy(ch, length(ch) div 2-i, n);
      writeln(ch2);
   until ch2=ch;
end;
(*******************PP**********************)
begin
   repeat
      write('CH = ');  readln(ch);
   until (ch<>'') and (odd(length(ch)));
   affiche_bas(esp,ch);
   affiche_haut(esp,ch);
end.
```

```
Program  suite_mystere;
uses   wincrt;
var    ligne, lignsuiv, c : string;
        n, l, nb, j, i : integer;
begin
    write ('N = '); readln (n);
    ligne:='1';
    for  i:=1  to  n  do
        begin
            writeln (ligne);
            l:=length (ligne);
            nb:=1;
            lignsuiv:='';
            for  j:=1  to  l  do
                if  ligne[j] = ligne[j+1]
                         then  nb:=nb+1
                         else  begin
                                     str (nb, c);
                                     lignsuiv:= lignsuiv + c + ligne[j];
                                     nb:=1;
                                end;
           ligne:= lignsuiv;
        end;
end.
```

Exercice 19

```
Program  Totalogram;
uses wincrt;
var ch:string;
(***************************************)
Function  lettre_maj_esp (ch:string):boolean;
var   i:integer;
        test:boolean;
begin
   i:=1;   test:=true;
   while (i<=length(ch)) and (test) do
          if  ch[i] in ['A'..'Z',' ']
                  then  i:=i+1
                  else test:=false;
  lettre_maj_esp:=test;
end;
(*************************************)
Function  totalogramme (ch:string):boolean;
var    p:integer;    test:boolean;
begin
  ch:=ch+' ';
  repeat
       p:=pos(' ',ch);
       test:=ch[1]=ch[p-1];
       delete(ch,1,p);
  until (test=false) or (ch='');
  totalogramme:=test;
end;
(**************** P.P *****************)
begin
 repeat
   writeln('Saisir une chaîne en majuscule :');
   readln(ch);
 until lettre_maj_esp(ch);
 if  totalogramme(ch)
        then writeln('totalogramme')
        else writeln('non totalogramme');
end.
```

Exercice 20

```
Program ch_distincte;
uses wincrt;
var ch:string;
(*************************************)
procedure saisie (var ch:string);
begin
   repeat
       writeln('Saisir une chaîne non vide :');
       readln(ch);
   until  ch<>'';
end;
(************************************)
Function  distincte (ch:string):boolean;
var    i:integer;
        test:boolean;
begin
  i:=0;
  test:=true;
  repeat
       i:=i+1;
       if pos(ch[i],ch)<>i then test:=false;
  until (test=false) or (i=length(ch));
  distincte:=test;
end;
(*************** P.P *****************)
begin
 saisie(ch);
 if  distincte(ch)
        then writeln('cette chaîne est distincte')
        else writeln('cette chaîne est non distincte');
end.
```

Exercice 23

Exercice 24

```
Program  Exercice_23;
uses   wincrt;
var     ch:string;
(*********************************************)
Procedure  saisie(var ch:string);
var    i:integer;
begin
  repeat
     writeln('saisir une chaine alphabétique :');
     readln(ch);
     i:=1;
     while upcase(ch[i]) in ['A'..'Z'] do i:=i+1;
  until  (i>length(ch)) and (length(ch) in [1..50]);
end;
(*********************************************)
Function construire (ch:string):string;
var       i, p1, p2, p3 : integer;
          res:string;
begin
   res:='';
   p1:=1;    p2:=1;    p3:=1;
   for   i:=1  to  length(ch)   do
      case   ch[i]   of
        'A'..'Z' : if not (ch[i] in ['A','E','I','O','U','Y'])
                           then  begin
                                    insert(ch[i],res,p1);
                                    p1:=p1+1;
                                    p2:=p2+1;
                                    p3:=p3+1;
                                 end
                           else   begin
                                    insert(ch[i],res,p2);
                                    p2:=p2+1;
                                    p3:=p3+1;
                                 end;
        'a'..'z' : if  not (ch[i] in ['a','e','i','o','u','y'])
                          then   begin
                                   insert(ch[i],res,p3);
                                   p3:=p3+1;
                                end
                          else  res:=res+ch[i];
      end;  {fin selon}
   construire:=res;
end;
(********************P.P.********************)
BEGIN
   saisie(ch);
   writeln ('La chaîne devient : ',construire(ch));
END.
```

```
Program  long_palindrome;
uses wincrt;
var   ch, ch1, max : string;
       nb, i, j : integer;
(*********************************************)
Function  palindrome (ch:string):boolean;
Var  i:integer;
      verif:boolean;
begin
    i:=0;
    repeat
       i:=i+1;
       verif :=  (ch[i] = ch[length(ch)-i+1]);
    until  (verif=false) or (i=length (ch) div 2);
    palindrome:=verif;
end;
(******************* P.P *******************)
begin
   write ('Donner ch = ');readln (ch);
   max:=ch[1];
   nb:=0;
   repeat
     for j:=length (ch)  downto  3  do
         begin
            ch1:=copy (ch,1,j);
            if palindrome (ch1)
              then begin
                     nb:=nb+1;
                     if length (ch1)>length (max) then max:=ch1;
                   end;
         end;
      delete (ch, 1, 1);
   until  length (ch)=2;
   writeln ('Le nombre des chaines palindromes est : ', nb);
   writeln ('La plus longue chaine palindrome est : ', max)
end.
```

Exercice 25

```
Program Cryptage ;
uses   wincrt;
var    ch,ch2:string;
(**************************************)
procedure saisie(var ch:string);
    function verif(ch:string):boolean;
    var i:integer;
        ok:boolean;
    begin
       i:=0;
       repeat
          i:=i+1;
          ok:= upcase(ch[i]) in ['A'..'Z',' ']
       until  (not ok) or (i=length(ch));
       verif:=ok;
    end;
begin
    repeat
       write('Phrase initiale = ');
       readln(ch);
    until (pos('  ',ch)= 0) AND (verif(ch)=true);
end;



(**************************************)
```

Exercice 26

```
Program  facteurs_premiers;
uses   wincrt;
var     p:integer;
         n:longint;
(*******************************************)
Procedure  saisie(var p:integer;var n:longint);
var  ch:string;
begin
   repeat
       write('p = ');
       readln(p);
   until (2<p) and (p<6);
   repeat
       write ('Donner un entier de ',p,' chiffres : ');
       readln(n);
       str(n,ch);
   until  length(ch)=p ;
end;
(*****************************************)
Function Facteurs(n:longint):string;
var  ch,chc,chd:string;
       d,c:integer;
begin
   d:=2;   ch:='';
   repeat
      if (n mod d) = 0
```

```
function crypter(ch:string):string;
var i,p:integer;
begin
  if ch[1]=' '
     then p:=0
     else p:=1;

  for i:=1 to length(ch) do
     if ch[i]<>' '
       then if (ord(upcase(ch[i]))+p) <= ord('Z')
                    then ch[i]:=chr(ord(ch[i])+p)
                   else ch[i]:=chr(ord(ch[i])+p-26)
       else p:=p+1;
  crypter:=ch
end;
(*********************P.P**********************)
BEGIN
  saisie(ch);
  ch2:=crypter(ch);
  writeln ('Phrase cryptée = ', ch2);
END.
```

```
            then begin
                   c:=0;
                   repeat
                     c:=c+1;
                     n:=n div d;
                   until (n mod d)<>0;
                   str(c,chc);
                   str(d,chd);
                   ch:=ch+chc+chd;
                 end
            else  d:=d+1;
   until (n=1);
   facteurs:=ch;
end;
(*****************P.P*************************)
BEGIN
 saisie(p,n);
 Writeln(facteurs(n));
END.
```

Exercice 27

```
Program  romain_decimal;
uses WinCrt;
var ch : string;
(******************************************)
procedure saisie(var ch : string);

function valide(ch : string):boolean;
  var i : integer;
      ok : boolean;
  begin
     i:=0;
     repeat
        i:=i+1;
        ok:= ch[i] in ['M','D','C','L','X','V','I']
     until  (not ok) or (i=length(ch));
     valide := ok;
  end;

begin
     repeat
        Write ('Entrer un nombre en chiffres romains : ');
        Readln(ch);
     until valide(ch);
end;
```

```
(*****************************************)
function convert(ch : string):integer;
var   i, s, v, v2 : integer;

function decimal(c : char):integer;
  begin
     case  c  of
          'I' : decimal := 1;
          'V' : decimal := 5;
          'X' : decimal := 10;
          'L' : decimal := 50;
          'C' : decimal := 100;
          'D' : decimal := 500;
          'M' : decimal := 1000;
     end;
  end;
begin
     s := 0;
     for i:=1 to Length(ch) do
     begin
          v := decimal(ch[i]);
          if (i < Length(ch))
              then begin
                     v2 := decimal(ch[i+1]);
                     if (v < v2) then v:= -v;
                   end;
          s := s + v;
     end;
     convert := s;
end;
(*********************P.P*****************)
begin
   Saisie(ch);
   Writeln(ch , ' = ', convert(ch));
end.
```

Exercice 28

```
Program   duplicate_chaine;
uses wincrt;
var   ch:string[20];
        res:string;
        i,j:integer;
Begin
  write('ch = ');readln(ch);
  res:='';
  for i:=1 to length(ch) do
     for j:=1 to i do
            res:=res+ch[i];
  writeln(res);
End.
```

Exercice 29

```
Program   Codage ;
uses    wincrt;
var     ch:string;  i,k:integer;
begin
  repeat
     writeln('Saisir un message en majuscule : ');  readln(ch);
     i:=1;
     while   ch[i]  in  ['A'..'Z' ,' ', '.']   do  i:=i+1;
  until  (i > length(ch));
  repeat
     write('Saisir le clé du codage : ');  readln(k);
  until  (0 <= k) and (k <= 25);
  for   i:=1   to   length(ch)  do
       if  ch[i]  in  ['A'..'Z']
            then  ch[i]:=chr(65+((ord(ch[i])-65)+k) mod 26);
   writeln('Message codé : ',ch);
end.
```

Exercice 30

```
Program Cryptage;
Uses   wincrt;
Var     msg,cle:string;
(****************************************)
Procedure saisi_msg (var msg:string);
Var i:integer;
Begin
  repeat
    write('Message à crypter : ');  readln(msg);
    i:=1;
    While (msg[i] in ['A'..'Z' , ' '])and(i<=length(msg)) do i:=i+1;
   until    i>length(msg);
end;
(********************************************)
Procedure saisi_cle (var cle:string;msg:string);
Var    i,l:integer;
Begin
   repeat
        write('Clé de cryptage : ');  readln(cle);
        i:=1;
        l:=length(cle);
        While (cle[i] in ['0'..'9']) and (i<=l) do i:=i+1;
   until (i>l) and (length(msg)=l) ;
end;
(*********************************************)
Function Crypter(msg,cle : string) : string;
var    i,j,k,e,c:integer;
       ch:string;
begin
  ch:='';
  for k :=1 to length(msg) do
     begin
       if  msg[k]=' '
           then ch:=ch+' '
           else begin
                 val(cle[k],c,e);
                 i:=ord(msg[k])-64;
                 j:=c+i;
                If  j>26  then j:=j mod 26 ;  {ou j:=j-26}
               ch:=ch+chr(j+64);
             end  ;
     end;
  Crypter:=ch;
end;
(*******************P.P***********************)
Begin
  Saisi_Msg(Msg);
  Saisi_Cle(Cle,Msg);
  Writeln(Crypter(Msg,Cle));
 End.
```

Exercice 31

```
Program  cryptage;
Uses   wincrt;
Var      ch : string;
(*******************************************)
Procedure  saisie (var ch : string);
var    i : integer;
       verif : boolean;
begin
   repeat
       write('donner ch : '); readln(ch);
       i:= 0;
       repeat
         i:=i+1;
         verif := (upcase(ch[i]) in ['A'..'Z'])and(length(ch) in [1..50]);
       until (verif = false) or (i=length(ch));
     until verif ;
end;
(********************************************)
Function  crypter(ch : string) : string;
var  i,nb : integer;
      res,ch2 : string;
begin
   i:= 1;
   res:='';
   repeat
      nb:=1;
      while (ch[i] = ch[i+1]) do
             begin
                nb:=nb+1;
                i:=i+1;
             end;
        str(nb,ch2);
        res:=res+ch2+ch[i];
        i:=i+1;
   until i>length(ch);
   crypter:=res;
end;
(******************P.P**********************)
Begin
   Saisie(Ch);
   Writeln(Crypter(Ch));
End.
```

Exercice 32

```
Program  poids_mot;
Uses   Wincrt;
Type   St=String[30] ;
Var     Mot:St;
(*****************************************)
Procedure Lecture (Var M:St );
Begin
  Writeln('Donner Votre Mot ');
  Repeat
    Readln(M);
  Until M <> '' ;
End ;
 (*****************************************)
Procedure Affiche (M:St) ;
Var  K,P:Integer;
Begin
  P:=0 ;
  For  K:=1 To Length(M)  Do
     If  Upcase(M[K]) In ['A','E','I','O','U','Y']
         Then   P:= P+ ( Ord(Upcase(M[K]))-64) * K ;

  Writeln('Le Mot "', M,'" À Pour Poids : ',P);
End ;

(***************P.P**********************)
Begin
   Lecture(Mot);
   Affiche(Mot);
End.
```

Exercice 34

```
Program BIGRAMME;
Uses Wincrt ;
Var   ch :  string;
(********************************)
Procedure saisie(Var ch:String);
Function Valide(ch:String):  Boolean;
Var   i:  Integer;
       test:  Boolean;
Begin
   i := 0;
   Repeat
     i := i+1;
     test := (ch[i] In ['a'..'z']) And (ch<>'') ;
   Until (test=False) Or (i=Length(ch));
   valide := test;
End;
Begin
   Repeat
     Writeln('donner une chaîne');
     Readln(ch);
   Until Valide(ch);
End;
(*************************************)
Procedure Afficher (ch :String);
Var   i,j,trv,occ,n:  Integer;
Begin
   n := Length(ch) ;
   For i:=1 To n-1 Do
     If ch[i]<>ch[i+1] Then
        Begin
          j := 1;
          trv := 0;
          While (j<i) And (trv=0) Do
            Begin
              If (ch[j]=ch[i]) And (ch[j+1]=ch[i+1]) Then trv := 1;
              j := j+1;
            End;
          If trv=0 Then
            Begin
              occ := 1;
              For j:=i+2 To n-1 Do
                If (ch[j]=ch[i]) And (ch[j+1]=ch[i+1])
                         Then   occ := occ+1;
```

Exercice 33

```
Program   Chaine_formee;
Uses    wincrt;
Type     tab=array[1..9] of char;
var        t:tab;
            mot:string;
            n:integer;
(************************************************)
procedure remplir (var t:tab; var n:integer);
var i:integer;
begin
  repeat
     writeln('donner la taille du tableau entre 5 et 9');
     readln(n)
  until  (n in [5..9]);
  for  i:=1  to  n  do
     repeat
        writeln('donner un caractère alphabet miniscule ',i);
        readln (t[i]);
     until   t[i] in ['a'..'z'];
end;
(************************************************)
procedure saisir (var ch:string);
var    i: integer;
         test: boolean;
begin
   repeat
      writeln('donner le mot a rechercher');
      readln(mot);
      i:=0;
      test:= true;
      repeat
          i:= i+1;
          if  not (ch[i] in ['a'..'z']) then test:= false;
     until (test=false) or (i=length(ch));
   until (test=true) and (length(ch) in [1..7]);
end;
(**********************************************)
procedure affiche (ch:string; t:tab; n:integer);
var   i,p:integer;
       cht :string;
begin
   cht:='';
   for  i:=1 to  n  do  cht:=cht+t[i];
   i:=0;
   repeat
        i:=i+1;
        p:=pos(ch[i],cht);
        if  p<>0  then write(p);
   until (p=0) or (i=length(ch));
   if  p=0
       then  writeln('la chaine ne pas être formée');
end;
(**********************P.P*******************)
Begin
   Remplir(T,N);
   Saisir(Mot);
   Affiche(Mot,T,N);
End.
```

```
  Writeln('le nombre d''occurrence de ',ch[i],ch[i+1],' est ',occ);
          End;
      End;
End;
(****************P.P**********************)
Begin
  saisie(ch);
  afficher(ch);
End.
```

Exercice 4

```
 PROGRAM point_milieu;
 uses  wincrt;
 type  point = record
                 x : real;
                 y : real;
               end ;
 var  a,b,m : point ;
 begin
   writeln ('donner les coordonnées du point A') ;
   readln(a.x, a.y) ;
   writeln ('donner les coordonnées du point B') ;
   readln(b.x, b.y) ;
   m.x := (a.x + b.x)/2 ;
   m.y := (a.y + b.y)/2 ;
   writeln('les coordonnées du point du milieu sont : ',m.x:2:2,
             '  ',m.y:2:2);
 end.
```

Exercice 5

```
PROGRAM calcul_complexe;
uses wincrt;
type  complexe = record
                    re,im:real;
                  end;
var  c1,c2,c3,c4 :complexe;
begin
writeln ('donner les parties réelle et imaginaire du 1er complexe');
readln (c1.re, c1.im) ;
writeln ('donner les parties réelle et imaginaire du 2eme complexe');
readln (c2.re, c2.im) ;
c3.re := c1.re + c2.re ; {calcul de c3 = c1 + c2}
c3.im := c1.im + c2.im;
writeln ('somme = ', c3.re:4:2,' + ', c3.im:4:2,' i');
c4.re := (c1.re*c2.re) - (c1.im*c2.im);
c4.im := (c1.re*c2.im) + (c1.im*c2.re);
writeln ('produit = ', c4.re:4:2,' + ', c4.im:4:2,' i');
end.
```

Exercice 6

```
PROGRAM personnel;
uses wincrt;
const n = 50;
type  employe = record
         matricule :integer;
         nom :string;
         salaire:real;
         etat_civil:char;
         end;
    tab = array[1..n] of employe;
var  tabemp :tab;
    nb :integer;
(*********************************************)
Procedure remplir(var t:tab);
var  i :integer ;
begin
  for i := 1 to n do
     begin
       writeln('matricule de l''employé ',i,' :');
       readln(t[i].matricule);
       writeln('nom de l''employé ',i,' :');
       readln(t[i].nom);
       writeln('salaire de l''employé ',i,' :');
       readln(t[i].salaire);
       writeln('etat civil de l''employé ',i,' : m ou c ?');
       readln(t[i].etat_civil);
     end;
end;


(*********************************************)
Function compter (t:tab):integer;
var i,c:integer;
begin
  c:=0;
  for i := 1 to n do
    if (t[i].salaire >=800) and (t[i].etat_civil='m') then c := c+1;
  compter := c;
end;
(************ p.p *************************)
begin
 remplir(tabemp);
 nb := compter(tabemp);
 writeln('le nombre d''employés mariés ayant le salaire >= à 800
est :',nb);
end.
```

Exercice 7

```
PROGRAM personnel;
uses wincrt;
const n = 10;
type  personne = record
            nom, prenom :string[20];
            age:1..150;
            end;
    tab = array[1..n] of personne;
var  tp :tab;
    i :integer ;
(***********************************************)
Procedure remplir(var t:tab);
var i :integer ;
begin
  for i := 1 to n do
    with t[i] do
      begin
        write('nom :') ;readln(nom);
        write('prénom : ') ; readln(prenom) ;
        write('age : ') ;readln(age) ;
      end;
end;
(***********************************************)
Procedure tri (var t:tab);
{tri par échange de minimums successifs}
var i,j:integer;
    aux :personne ;
begin
  for i := 1 to n-1 do
      for j :=i+1 to n do
          if t[i].nom > t[j].nom
             then  begin
                     aux := t[i] ;
                     t[i] := t[j] ;
                     t[j] := aux ;
                   end ;
end;
(****************** p.p *******************)
begin
  remplir(tp) ;
  tri(tp) ;
  for i :=1 to n do
      with tp[i] do
          writeln(nom :25,prenom :15,age :10) ;
end.
```

Exercice 8

```
PROGRAM  fichier;
uses wincrt;
var   f: file of char;
      n: byte;
      car:char;
(****************************************)
Procedure creation;
begin
   assign (f, 'c:\alpha.txt');
   rewrite (f);
   for car:='a' to 'z' do  write (f, car);
end;
(***************************************)
Procedure afficher;
begin
   reset (f);
   n:= filesize(f);
   writeln ('nombre d''éléments : ', n);
   seek (f, (n div 2)-1);  {déplace le pointeur sur le caractère du milieu}
   read (f,car);
   writeln ('le caractère du milieu est : ', car);
   writeln ('le caractère de milieu est à la position: ', filepos(f));
end;
(***************************pp*******************************)
begin
  creation;
  afficher;
  close (f);
end.
```

Exercice 9

```
PROGRAM  fiche_employe;
uses wincrt;
type    employe = record
                    matricule:integer;
                    nom:string;
                    prenom:string;
                    grade:string;
                    salaire:real;
                  end;
        fpers = file of employe ;
var     fp : fpers ; e: employe;  mat:integer;
 (******************************************************)
 Procedure creation (var fp:fpers);
 var   nomf : string ; rep :char;
 begin
   writeln ;
   write ('entrer le nom du fichier à écrire : '); readln (nomf);
   assign (fp, nomf);
   rewrite (fp);
   repeat
     repeat
         write('voulez-vous ajouter un employé (o/n): '); readln(rep);
     until (upcase(rep)='o')or (upcase(rep)='n');
     if upcase(rep)  = 'o'
          then  with e do
                  begin
                         write ('entrez la matricule : '); readln (matricule) ;
                                     write ('entrez le nom : '); readln (nom) ;
                                     write ('entrez le prénom : ');readln (prenom) ;
                                     write ('entrez le grade : ');readln (grade) ;
                                     write ('entrez le salaire : ');readln (salaire) ;
                                     write (fp, e);
                             end;
   until upcase(rep)= 'n' ;
   reset (fp);
   writeln('le nombre des employés dans le fichier est : ', filesize(fp));
 end;
 (*******************************************************)
 Procedure consultation;
 var   nomf:string;
 begin
   write('entrer le nom du fichier à lire : '); readln(nomf);
   assign (fp,nomf);
   reset (fp);
   writeln ;
   writeln('liste des employés qui ont un salaire entre 500 et 700:') ;
   writeln('-----------------------------------------------------') ;
   writeln ;
   while  not eof(fp) do
    begin
      read (fp, e) ;
      if (e.salaire >=500) and (e.salaire <=800)
           then  begin
                 writeln ('matricule : ', e.matricule);
                 writeln ('nom : ', e.nom);
                 writeln('prénom : ', e.prenom);
                end;
    end;
    close (fp);
   end;
 (*****************************************************)
   Function recherche (mat:integer):boolean;
   var  nomf:string; trouve :boolean;
   begin
     write('entrer le nom du fichier à utilisé : ');
     readln(nomf);
     assign(fp,nomf);
     reset(fp);
     writeln ;
     read(fp, e) ;
     trouve := (e.matricule =mat);
     while (trouve = false) and not (eof(fp)) do
          begin
            read(fp, e);
            trouve := (e.matricule =mat);
          end;
    recherche := trouve;
  end;
(*************************pp*************************)
 begin
   creation(fp) ;
   consultation;
   writeln('entrer le matricule de l''employé à chercher dans le fichier :');
   readln(mat);
   if recherche(mat)
     then  writeln('l''employé de matricule ', mat , ' existe dans le fichier ')
     else  writeln('l''employé de matricule ', mat , ' n''existe pas dans le fichier ')
 end.
```

Exercice 10

```
Program  facteurs_premiers;
uses wincrt;
type  f_ent=file of longint;
var   f1:f_ent;
    f2:text;
    n,p:integer;
(*****************************************)
Procedure  saisie(var n,p:integer);
begin
  repeat
    write('n = '); readln(n);
  until (2<n) and (n<100);
  repeat
    write('p = '); readln(p);
  until (2<p) and (p<6);
end;
(*****************************************)
Procedure  Remplir_f1(var f1:f_ent;n,p:integer);
var  x:longint; i:integer; ch:string;
begin
 rewrite(f1);
 for i:=1 to n do
   begin
     repeat
       write('donner un entier de ',p,' chiffres:');
       readln(x);
       str(x,ch);
     until  length(ch)=p ;
     write(f1,x);
   end;
```

```
end;
(****************************************)
Procedure Remplir_f2(var f1:f_ent;var f2:text);
var ch,chc,chd:string;
    d,c:integer; n:longint;
begin
   reset(f1);
   rewrite(f2);
   while not eof(f1) do
      begin
          read(f1,n);
          d:=2;ch:='';
          repeat
            if (n mod d) = 0
                  then begin
                          c:=0;
                          repeat
                             c:=c+1;
                             n:=n div d;
                          until (n mod d)<>0;
                          str(c,chc);
                          str(d,chd);
                          ch:=ch+chc+chd;
                        end
               else  d:=d+1;
          until (n=1);
          writeln(f2,ch);
      end;
end;
(**********************************************)
Procedure  affiche_f2 (var f2:text);
var ch:string;
begin
   reset(f2);
   while not eof(f2) do
      begin
         readln(f2,ch);
         writeln(ch);
      end;
end;
(******************P.P************************)
BEGIN
       saisie(n,p);
       assign(f1,'c:\nombres.dat');
       assign(f2,'c:\facteurs.txt');
       remplir_f1(f1,n,p);
       remplir_f2(f1,f2);
       affiche_f2(f2);
       close(f2) ;
END.
```

Exercice 11

```
{méthode 1 : supprimer un composant d'un fichier}
PROGRAM supprim_element;
uses wincrt;
type  ft=file of integer;
var   f:ft;
(*********************************************)
Procedure saisie(var f:ft);
var n:integer;
begin
   assign(f,'c:\dest.dat');
   rewrite(f);
    repeat
        write('entrer un entier non nul (0 pour terminer la saisie) : ');
        readln(n);
        if (n <> 0) then write(f, n);
    until (n = 0);
end;
(********************************************)
Procedure supprimer (var f:ft);
var vs:integer;
(****************************)
Function trouve (vs:integer;var f:ft):boolean;
var tr:boolean;  vtemp:integer;
begin
```

```
   tr:=false;
   while (not eof(f)) and (not (tr)) do
        begin
           read (f,vtemp);
           tr:=(vtemp=vs);
        end;
   trouve:=tr;
end;
(**********************************)
Procedure suppr(vs:integer;var f:ft);
var vtemp:integer;  f2:ft;
begin
    assign(f2,'c:\dest2.dat'); rewrite(f2);
    reset(f);
    while not eof(f) do
        begin
          read(f,vtemp);
          if vtemp<>vs then write(f2,vtemp);
        end;
    erase(f);
    rename (f2,'c:\dest.dat');
end;
(********************************)
begin
   reset(f);
   writeln('l''entier à supprimer ?');
   readln(vs);
   if trouve(vs,f) then suppr(vs,f)
                   else writeln('valeur inexistante');
end;
(**************************************************************)
Procedure afficher(var f:ft);
var nb:integer;
begin
   reset(f);
   while not eof(f) do
      begin
        read(f,nb);
        write(nb,' ');
      end;
end;
(**********************pp******************************)
begin
  saisie(f);
  supprimer(f);
  afficher(f);
  close(f);
end.

{méthode 2 : supprimer un composant d'un fichier}
PROGRAM supprim_element;
uses wincrt;
type  ft=file of integer;
var   f:ft;
(**********************************************)
Procedure saisie(var f:ft);
var n:integer;
begin
   assign(f,'c:\dest.dat');
   rewrite(f);
    repeat
        write('entrer un entier non nul (0 pour terminer la saisie) : ');
        readln(n);
        if (n <> 0) then write(f, n);
    until (n = 0);
end;
(*********************************************)
Procedure supprimer (var f:ft);
var vs:integer;
(******************************)
Function trouve (vs:integer;var f:ft):boolean;
var tr:boolean;
    vtemp:integer;
begin
   tr:=false;
   while (not eof(f)) and (not (tr)) do
```

```
        begin
           read (f,vtemp);
           tr:=(vtemp=vs);
        end;
   trouve:=tr;
end;
(**********************************)
Procedure suppr(vs:integer;var f:ft);
var vtemp:integer;
begin
    while not eof(f) do
        begin
          read(f,vtemp);
          seek(f,filepos(f)-2);
          write(f,vtemp);
          seek(f,filepos(f)+1);
        end;
        seek(f,filepos(f)-1);
        truncate(f);
end;
(*******************************)
begin
   reset(f);
   writeln('l''entier à supprimer ?');
   readln(vs);
   if trouve(vs,f) then suppr(vs,f)
                   else writeln('valeur inexistante');
end;
(************************************************************)
Procedure afficher(var f:ft);
var nb:integer;
begin
   reset(f);
   while not eof(f) do
      begin
        read(f,nb);
        write(nb,' ');
      end;
end;
(*********************P.P**********************************)
begin
  saisie(f);
  supprimer(f);
  afficher(f);
  close(f);
end.
```

Exercice 12

```
PROGRAM  insertion_composant; {insertion un composant dans un fichier}
uses wincrt;
type  tf = file of integer;
      tab = array[0..20] of integer;
var   f:tf;
      i:integer;
      t:tab;
(*******************************************)
Procedure saisie(var f:tf);
var i:integer;
begin
   assign(f,'c:\dest.dat');
   rewrite(f);
   for i:=1 to 5 do  write(f,i);
end;
(*******************************************)
Procedure insertion (var f:tf;var i:integer;var t:tab);
(*on recopie l'intégralité de fichier dans t, en insérant le nombre au bon endroit*)
var nb,num:integer;
begin
   reset(f);
   writeln('donner un entier à inserer');
   readln(num);
   i:=-1;
   while not eof(f) do
      begin
         i:=i+1;
         read(f,nb);
```

```
        if nb<=num then t[i]:=nb else begin
                                 t[i]:=num;
                                 num:=nb;
                               end;
     end;
  t[i+1]:=num;
end;
(*********************************************)
Procedure recopie(var f:tf;i:integer;t:tab);
 (* on recopie à présent t dans le fichier*)
var j:integer;
begin
   rewrite(f);
   for j:=0 to i+1 do write(f,t[j]);
end;
(*********************************************)
Procedure affiche(var f:tf);
var nb:integer;
begin
   reset(f);
   writeln('contenu du fichier final');
   while not eof(f) do
      begin
        read(f,nb);
        write(nb,' ');
      end;
end;
(**********************pp*************************)
begin
   saisie(f);
   insertion(f,i,t);
   recopie(f,i,t);
   affiche(f);
   close(f);
end.
```

Exercice 12

```
PROGRAM  insertion_composant; {insertion d'un composant dans un fichier}
uses wincrt;
type  tf=file of integer;
var   f:tf;
(*********************************************)
Procedure saisie(var f:tf);
var i:integer;
begin
   assign(f,'c:\dest.dat'); rewrite(f);
   for i:=1 to 5 do  write(f,i);
end;
(*********************************************)
Procedure insertion (var f:tf);
var nb,num,p,i:integer;
begin
   reset(f);
   writeln('donner un entier à inserer'); readln(num);
   repeat
     read(f,nb);
   until (num<nb) or eof(f);;
   if not eof (f)
       then begin
              p:=filepos(f)-1;
              for i:=(filesize(f)-1) downto p do
                    begin
                      seek(f,i);
                      read(f,nb);
                      write(f,nb);
                    end;
              seek(f,p);
              write(f,num);
            end
       else write(f,num);
end;
(*********************************************)
Procedure affiche(var f:tf);
var nb:integer;
begin
   reset(f);
   writeln('contenu du fichier final');
   while not eof(f) do
      begin
        read(f,nb);
        write(nb,' ');
      end;
end;
(***********************P.P*****************************)
begin
   saisie(f);
   insertion(f);
   affiche(f);
   close(f);
end.
```

Exercice 13

```
PROGRAM   affiche_source;
uses   wincrt;
var    f:text;
         chemin:string;
(*********************************************)
Procedure source (var f:text);
var ligne:string;
begin
   while (not eof(f)) do
       begin
         readln(f,ligne);
         writeln(ligne);
       end;
end;
(******************************pp*************************)
begin
   writeln('donner le chemin d''un fichier pascal');
   readln(chemin);
   assign(f,chemin);
   reset(f);
   clrscr;
   source(f);
   close(f);
end.
```

Exercice 14

```
PROGRAM  exercice14;
uses wincrt;
var    f: text;   nom : string;
(*******************************************************)
Procedure saisie(var f:text);
var  ch :string; rep : char;
begin
      write('donner une ligne :'); readln(ch);
      while ch[length(ch)]<>'.' do
          begin
            write(f,ch);writeln(f); {ou bien writeln(f,ch);}
            write('donner une ligne :'); readln(ch);
          end;
end;
(*****************************************************)
Procedure compter (var f : text);
var   ch : string; nb,i,l:integer;
begin
  reset(f);l:=0;
  while not eof(f) do
    begin
       nb := 0;l:=l+1;
       readln(f,ch);
       i := 1;
       repeat
          if ch[i]<>' '
              then begin
                     nb:=nb+1;
                     while ch[i] <>' ' do i:=i+1;
                   end
              else  i:=i+1;
       until (i>=length(ch));
       writeln('le nombre des mots dans la ligne ',l, ' = ', nb);
    end;
```

```
end;
(******************************************************************)
Procedure frequence (var f : text);
var   ch : string; nb,i,l:integer; c:char;
begin
  reset(f);l:=0;
  while not(seekeof(f)) do
  begin
       nb := 0;l:=l+1;
       while not eoln(f) do
                begin
                  read(f,c);
                  if upcase(c)in['A','E','O','Y','U','I'] then inc(nb);
                end;
       writeln('le nombre de voyelles dans la ligne',l, ' = ', nb);
  end;
end;
(***************************pp**************************************)
begin
   write('entrer le nom du fichier : '); readln(nom);
   assign(f, 'd:\' + nom + '.fch');
   {$i-} reset(f);{$i+}
   if  ioresult<>0  then   begin
                              rewrite(f);
                              saisie(f);
                           end;
   writeln;
   compter(f);writeln;
   frequence(f);
   close(f);
end.
```

Exercice 15

```
PROGRAM  nbr_lettres ;
uses  wincrt ;
type   tab = array [ord('A')..ord('Z')] of integer ;
       chemin = string [50] ;
var    f : text ; nom : chemin ; t : tab ; ligne : string ;
(***********************************************************)
Procedure ouvre (var f:text ; nom:chemin) ;
begin
   write ('entrer le nom du fichier : '); readln (nom) ;
   nom := nom + '.txt';
   assign (f, nom) ;
   {$i-} reset (f); {$i+}
   if ioresult <> 0 then rewrite (f);
end;
(************************************************************)
Procedure saisie (var f:text ; ligne:string) ;
begin
     writeln ('pour terminer la saisie, taper le caractère *');
     writeln ;
     write ('taper une ligne de texte : '); readln (ligne) ;
     while ligne <> '*' do
           begin
                writeln (f, ligne);
                write ('taper une nouvelle ligne de texte : '); readln (ligne) ;
           end;
end;
(****************************************************************)
Procedure cumul (var f:text ; var t:tab; ligne:string);
var   i, ca : integer ;
begin
     {init du tableau à zéro }
     for i := ord('A') to ord('Z') do t[i] := 0 ;
     {remise à zéro du pointeur du fichier }
     reset (f);
     {lecture du fichier ligne par ligne et cumul du nombre des lettres}
     while  not (eof(f)) do
           begin
                readln (f, ligne);
                for i:=1 to length(ligne) do
                    begin
                        ligne[i] := upcase(ligne[i]);
                        {cumul du nombre de lettre dans le tableau t }
                        ca := ord(ligne[i]) ;
```

```
             if ca in [ord('A')..ord('Z')] then t[ca] := t[ca] + 1 ;
          end ;
      end ;
end;
(****************************************************)
Procedure affiche (t : tab);
var   nb, i : integer ; pc : real ;
begin
   {calcul du nombre total de lettres}
   nb := 0 ;
   for i:=ord('A') to ord('Z') do nb := nb + t[i] ;
   writeln('le nombre total de lettres dans le fichier est : ', nb );
   {affichage de l'occurrence et du pourcentage de présence de chaque lettre}
   for i:=ord('A') to ord('Z') do
     begin
      if nb > 0 then pc:=(t[i]*100)/nb  else pc:=0;
      if t[i]<>0
        then writeln ('la lettre: ',chr(i),' est présente: ',t[i],' fois, donc : ',pc:3:2, '%') ;
     end;
end;
{======================== p.p ========================}
begin
   ouvre (f, nom);
   saisie (f, ligne);
   cumul (f, t, ligne);
   affiche (t);
end.
```

Exercice 16

```
PROGRAM   code_bin_texte ;
uses     wincrt ;
var      f1,f2 : text ;
Procedure saisie(var f1:text);
var phrase:string;
begin
 assign (f1, 'c:\modele.txt');
 rewrite (f1);
 writeln ('taper le texte');
 repeat
   readln (phrase);
   if phrase<>'' then writeln (f1, phrase);
 until phrase = '';
end;
Procedure codage(var f1,f2:text);
var nd:integer;
  car:char;
  ch1,ch2:string;
begin
  reset (f1);
  assign (f2, 'c:\binaire.txt');
  rewrite (f2);
  while not eof (f1) do
   begin
    while  not eoln (f1)  do
      begin
       read (f1, car) ;
       nd:=ord(car);
       ch2:='';
       repeat
         str(nd mod 2,ch1);
         ch2:=ch1+ch2;
         nd:= nd div 2 ;
       until  (nd = 0);
       write (f2,ch2) ;
      end ;
    readln(f1);
    writeln(f2) ;
  end ;
end;
Procedure affiche(var f2:text);
var phrase:string;
begin
 reset (f2);
 while not eof(f2) do
   begin
    readln (f2, phrase);
```

```
      writeln(phrase);
    end;
end;


(*****************************pp******************************)
begin
  saisie(f1);
  codage(f1,f2);
  affiche(f2);
  close(f1);
  close(f2);
end.
```

Exercice 17

```
Procedure trie (var f:fch ; chemin:string);      {tri à bulle d'un fichier}
var   ch1,ch2:string;
      permutation : boolean;
      i,n :Integer;
begin
    assign(f,chemin);reset(f);
    repeat
         permutation:=False;
         n:=filesize(f);
         for i:= 0 to n-2 do
            begin
                seek(f,i);
                read(f,ch1);
                read(f,ch2);
                if ch1 > ch2
                then begin
                     seek(f,i);
                     Write(f,ch2);
                     Write(f,ch1);
                     permutation:=true;
                     end;
            end;
          n:=n-1;
    until  (not permutation) OR (n=1);
end;
(**********************************************)
Procedure fusion (var f1,f2,f3:fch);
var  fc1,fc2:string;
begin
    assign(f1,'c:\file1.dat');reset(f1);
    assign(f2,'c:\file2.dat');reset(f2);
    assign(f3,'c:\file3.dat');rewrite(f3);
    repeat
       read(f1,fc1);
       read(f2,fc2);
       if fc1<fc2  then begin
                     write(f3,fc1);
                     seek(f2,filepos(f2)-1);
                  end
              else begin
                     write(f3,fc2);
                     seek(f1,filepos(f1)-1);
                  end ;
    until (eof(f1)) or (eof(f2));
    while not eof(f1) do
       begin
          read(f1,fc1);
          write(f3,fc1);
       end;
    while not eof(f2) do
       begin
          read(f2,fc2);
          write(f3,fc2);
       end;
end;
```

Exercice 18

```
PROGRAM  ligne_impair_pair ;
uses   wincrt ;
var     f1,f2,f3: text ;
(****************************************)
Procedure saisie(var f1:text);
var  ligne:string;
       rep:char;
begin
   Assign(f1,'c:\init.txt');
   rewrite(f1);
   repeat
      writeln('Taper une ligne de texte');readln(ligne);
      writeln(f1,ligne);
      repeat
         writeln('Quitter (O/N) ?');
         readln(rep);
      until upcase(rep) in ['O','N'];
   until upcase(rep)='O';
end;
(****************************************)
Procedure copie(var f1,f2,f3:text);
var ligne:string;
begin
   reset(f1);
   Assign(f2,'c:\copie2.txt');
   rewrite(f2);
   Assign(f3,'c:\copie3.txt');
   rewrite(f3);
   while not eof (f1) do
     begin
      readln (f1, ligne);
      writeln (f2,ligne);
      readln (f1, ligne);
      writeln (f3,ligne);
     end;
end;
(*****************************************)
Procedure affiche(var f:text);
var  ligne:string;
begin
  Reset (f);
  while not eof (f) do
     begin
      readln (f, ligne);
      writeln (ligne);
     end;
  writeln;
end;
(***********************PP*************************)
begin
  saisie(f1);
  copie(f1,f2,f3);
  affiche(f2);
  affiche(f3);
  close (f1);
  close (f2);
  close (f3);
end.
```

<ins>**Exercice 19**</ins>

```
PROGRAM exercice_20;
uses  wincrt;
type f_ent=file of integer;
var f1,f2:f_ent;
    nom1,nom2:string;
(*********************************************)
Function egalf(var f1,f2:f_ent):boolean;
var a,b:integer;
    test:boolean;
begin
  repeat
     if eof(f1) or eof(f2)
        then test:=eof(f1) and eof(f2)
        else
           begin
             read(f1,a);
```

```
          read(f2,b);
          test:=a=b;
         end;
  until (test=false) or(eof(f1) and eof(f2));
  egalf:=test;
end;
(*****************************************)
begin
  write('Nom du premier fichier : '); readln(nom1);
  write('Nom du deuxième fichier : '); readln(nom2);
  assign(f1,nom1);
  reset(f1);
  assign(f2,nom2);
  reset(f2);
  if egalf(f1,f2)
      then writeln('les fichiers sont égaux')
      else writeln('les fichiers ne sont pas égaux');
end.
```

Exercice 20

```
 PROGRAM  club_sportif;
 uses  wincrt;
 type adherent =record
                nca:integer;
                nom: string;
                dat:string;
                nbh:integer;
              end;
      fa=file of adherent;
 var  f:fa;
     choix:char;
     ft:text;
 (**************************************************************)
 Procedure transfert(var ft:text;var f:fa);
 var vligne,ch:string;
    vtemp:adherent;
    err:integer;
 (********************************)
 Procedure extraire(var v,ch:string);
 var p:integer;
 begin
   p:=pos(',',v);
   ch:=copy(v,1,p-1);
   delete(v,1,p);
 end;
 (********************************)
 begin
   {$i-} reset (ft);{$i+}
   if  ioresult<>0 then  rewrite(ft);
   rewrite(f);
   while not eof(ft) do
      begin
        readln(ft,vligne);
        with vtemp do
         begin
           extraire(vligne,ch);
           val(ch,vtemp.nca,err);
           extraire(vligne,vtemp.nom);
           extraire(vligne,vtemp.dat);
           val(vligne,vtemp.nbh,err);
         end;
        write(f,vtemp);
      end;
   close(f);
   close(ft);
 end;
 (**********************************************************************)
 Procedure menu (var choix:char;var f:fa);
 (***********************************)
 Procedure affiche_menu;
 begin
   clrscr;
   writeln;
   writeln('        ***********************************************************************');
   writeln('        * A. ajouter un nouveau adhérent                         *');
   writeln('        * M. modifier quelques informations relatives à un adhérent    *');
   writeln('      * S. supprimer un adhérent                      *');
   writeln('      * T. trier les noms des adhérents               *');
   writeln('      * Q. quitter l''application                   *');
   writeln('      ***********************************************************');
 end;
 (********************************)
 Procedure ajouter(var f:fa);
 var vtemp:adherent;
 begin
   with vtemp do
    begin
     writeln('numéro carte : ');readln(nca);
     writeln('nom : ');readln(nom);
     writeln('date de naiss : ');readln(dat);
     writeln('nombre d''heure : ');readln(nbh);
    end;
   reset(f);
   seek(f,filesize(f));
   write(f,vtemp);
   close(f);
 end;
 (******************************************)
 Procedure modifier(var f:fa);
 var vtemp:adherent;
    num:integer;
 (**********************)
 Function trouve(num:integer;var f:fa):boolean;
 var vtemp:adherent;
    tr:boolean;
 begin
   tr:=false;
   seek(f,0);
   while (not eof(f)) and (not(tr)) do
         begin
           read(f,vtemp);
           tr:=vtemp.nca=num;
         end;
   trouve:=tr;
 end;
 (***********************)
 begin
   writeln('numéro de carte d''adhérent ?');readln(num);
   reset(f);
   if trouve(num,f) then
    begin
         seek(f,filepos(f)-1);
         read(f,vtemp);
     with vtemp do
           begin
            writeln('nom : ');readln(nom);
            writeln('date de naiss : ');readln(dat);
            writeln('nombre d''heure : ');readln(nbh);
           end;
     seek(f,filepos(f)-1);
     write(f,vtemp);
    end
    else begin
           writeln('numéro de carte inexistant');
           readln;
         end;
   close(f);
 end;
 (*****************************************)
 Procedure supprimer (var f:fa);
 var num:integer;
 (*************************)
 Function trouve(num:integer;var f:fa):boolean;
 var vtemp:adherent;
    tr:boolean;
 begin
   tr:=false;
   seek(f,0);
   while (not eof(f)) and (not(tr)) do
         begin
           read(f,vtemp);
           tr:=vtemp.nca=num;
         end;
   trouve:=tr;
 end;
 (***********************)
 Procedure suppr(num:integer;var f:fa);
 var vtemp:adherent;
    f2:fa;
 begin
   assign(f2,'d:\club\ftemp.dat');
   rewrite(f2);
   reset(f);
   while not eof(f) do
    begin
     read(f,vtemp);
     if vtemp.nca<>num then write(f2,vtemp);
    end;
   close(f2);
   erase(f);
   rename(f2,'d:\club\adherent.dat');
 end;
 (***********************)
 begin
   writeln('numéro de carte d''adhérent ?');readln(num);
   reset(f);
   if trouve(num,f)
     then suppr(num,f)
     else begin
            writeln('numéro de carte inexistant');
            readln;
           end;
   close(f);
 end;
 (*********************************************)
 Procedure trier (var f : fa);
 type tab=array[1..10] of adherent;
 var t:tab;
    n,i:integer;
 (********************************)
 Procedure  remplir_t (var f:fa;var t:tab;var n:integer);
 begin
    reset(f); n:=0;
    while  not (eof (f))  do
         begin
           n := n+1;
           read (f, t[n]);
         end;
 end;
 (*******************************)
 Procedure tri_insertion (var t:tab;n:integer);
 var i,j:integer;
    tmp:adherent;
 begin
  for i:=2 to n do
    if t[i].nom<t[i-1].nom
     then begin
            tmp:=t[i];
            j:=i;
            while (j>1) and (t[j-1].nom>tmp.nom) do
                begin
                  t[j]:=t[j-1];
                  j:=j-1;
                end;
            t[j]:=tmp;
           end;
 end;
 (******************************)
 begin
  remplir_t (f, t, n);
  tri_insertion (t, n);
  seek(f,0);
  for i:=1 to n do
         write(f,t[i]);
  close(f);
 end;
 (************************************)
 begin
```

```
  affiche_menu;
  repeat
      write('choisir une action : ');
      readln(choix);
  until upcase(choix) in ['A','M','S','T','Q'];
  case upcase(choix) of
    'A':ajouter(f);
    'M':modifier(f);
    'S':supprimer(f);
    'T':trier(f);
    end;
end;
{*************** programme principal **********************}
begin
    assign(f,'d:\club\adherent.dat');
    assign(ft,'d:\club\adherent.txt');
    transfert(ft,f);
    repeat
       menu(choix,f);
    until upcase(choix) = 'Q';
end.
```

Exercice 21

```
PROGRAM  cabinet_medecin;
uses wincrt;
type  date = record
                  jour:1..31;
                  mois:1..12;
                  annee:integer;
             end;
      patient=record
                num:integer;
                nom: string[30];
                date_nais:date;
                date_rdv:date;
              end;
      fp=file of patient;
var  f:fp;
     choix:integer;
(**************************************************************)
Procedure creation(var f:fp);
begin
   assign (f, 'c:\cabinet.dat');
   {$i-}
       reset (f);
   {$i+}
   if  ioresult<>0 then  rewrite(f);
end;



(***********************************************************)
Procedure choisir (var choix:integer);
begin
  clrscr;
  writeln;
  writeln('        *******************************************************************');
  writeln('        * 1. ajouter un nouveau patient                               *');
  writeln('        * 2. modifier les informations relatives à un patient         *');
  writeln('        * 3. attribuer un rdv pour un patient                         *');
  writeln('        * 4. afficher les informations relatives à un patient         *');
  writeln('        * 5. afficher la liste des patients                           *');
  writeln('        * 6. afficher la liste des patients ayant un rdv donné        *');
  writeln('        * 7. quitter l''application                                   *');
  writeln('        *******************************************************************');
  repeat
      write('choisir une action : ');
      readln(choix);
  until choix in [1..7];
end;
(***********************************************************)
Procedure ajout_patient(var f:fp);
var p:patient;
begin
   reset(f);
   with p do
   begin
     num:=filesize(f);
     writeln('saisir le nom du patient');readln(nom);
     with date_nais do
        begin
          write('jj = ');readln(jour);
          write('mm = ');readln(mois);
          write('aaaa = ');readln(annee);
        end;
     with date_rdv do
        begin
          write('jj rdv = ');readln(jour);
          write('mm rdv = ');readln(mois);
          write('aaaa rdv = ');readln(annee);
        end;
   end;
   seek(f,filesize(f));
   write(f,p);
   close(f);
end;
(********************************************************)
Procedure modifier_patient(var f:fp);
var num:integer;
   p:patient;
begin
  writeln('saisir le numéro du patient à modifier');
  readln(num);
  reset(f);
  if num in [0..filesize(f)-1]
    then begin
         seek(f,num);
         read(f,p);
         with p do
           begin
            writeln('saisir le nouveau nom ');readln(nom);
            with date_nais do
               begin
                 write('nouveau jj = ');readln(jour);
                 write('nouveau mm = ');readln(mois);
                 write('nouveau aaaa = ');readln(annee);
               end;
           end;
         seek(f,num);
         write(f,p);
       end
    else writeln('patient inexistant !!!');
  close(f);
end;
(************************************************************)
Procedure attribuer_rdv(var f:fp);
var num:integer;
   p:patient;
begin
  writeln('saisir le numéro du patient qui veut avoir un rdv');
  readln(num);
  reset(f);
  if num in [0..filesize(f)-1]
    then begin
         seek(f,num);
         read(f,p);
         with p,date_rdv do
             begin
               write('jj = ');readln(jour);
               write('mm = ');readln(mois);
               write('aaaa = ');readln(annee);
             end;
         seek(f,num);
         write(f,p);
       end
    else writeln('patient inexistant !!!');
  close(f);
end;
(**********************************************************)
Procedure affiche_patient(var f:fp);
var num:integer;
   p:patient;
```

```
begin
  writeln('saisir le numéro du patient à afficher');
  readln(num);
  reset(f);
  if num in [0..filesize(f)-1]
    then begin
         seek(f,num);
         read(f,p);
         with p do
           begin
            writeln('numéro : ',num);
            writeln('nom et prénom : ',nom);
            with date_nais do writeln('date de naissance : ', jour, '/', mois, '/', annee);
            with date_rdv do writeln('date du rdv : ',jour,'/',mois,'/',annee);
           end;
        end
    else writeln('patient inexistant !!!');
  readkey;
end;
(***********************************************************)
Procedure affiche_liste(var f:fp);
var  p:patient;
begin
  reset(f);
  while not eof(f) do
    begin
      read(f,p);
      with p do
           begin
            writeln('numéro : ',num);
            writeln('nom et prénom : ',nom);
            with date_nais do writeln('date de naissance : ', jour, '/', mois, '/', annee);
            with date_rdv do writeln('date du rdv : ',jour,'/',mois,'/',annee);
           end;
    end;
 readkey;
end;
(*************************************************************)
Procedure affiche_rdv(var f:fp);
var  p:patient;  rdv:date;
begin
  with rdv do
     begin
      write('jj = ');readln(jour);
      write('mm = ');readln(mois);
      write('aaaa = ');readln(annee);
     end;
  reset(f);
  while not eof(f) do
    begin
      read(f,p);
      if (p.date_rdv.jour = rdv.jour) and (p.date_rdv.mois = rdv.mois) and (p.date_rdv.annee = rdv.annee)
        then with p do
             begin
              writeln('numéro : ',num);
              writeln('nom et prénom : ',nom);
              with date_nais do writeln('date de naissance : ', jour, '/', mois, '/', annee);
             end;
    end;
  readkey;
end;
{******************** programme principal ******************************}
begin
   creation(f);
   repeat
     choisir(choix);
     case choix of
      1: ajout_patient(f);
      2: modifier_patient(f);
      3: attribuer_rdv(f);
      4: affiche_patient(f);
      5: affiche_liste(f);
      6: affiche_rdv(f);
      end;
  until choix = 7;
  close(f);
 end.
```

```
PROGRAM  Videotheque ;
USES  WinCrt;
TYPE  Films = RECORD
                  Reference : Word ;
                  Titre : String [40] ;
              End ;
      Fichier = File Of Films ;
      nom_physique = String [40];
VAR   choix : Byte ;
      ouvert : Boolean ;

(******************Procédure d'attente sur l'appuie sur ECHAP****************)
PROCEDURE Echap ;
VAR Rep : Char ;
BEGIN
   WriteLn ;
   WriteLn ('Appuyer sur la touche [ECHAP] pour continuer ...' : 20) ;
   Repeat
       Rep := Readkey ;
   Until Rep = CHR (27) ; {27 = code ascii de la touche Echap }
END ;

(*****************Procédure de création d'un nouveau fichier****************)
PROCEDURE Creation (VAR nl : Fichier;VAR nf : nom_physique);
VAR   L : Byte ;
BEGIN
   ClrScr ; {Efface l'écran texte }
   Writeln ('CREATION DE FICHIER' : 50);
   WriteLn ; WriteLn ;
   Writeln ('Le fichier sera créer sur la racine du disque dur C'); WriteLn ;
   WriteLn ('Pour des raison de compatibilité avec le SE DOS');
   WriteLn (' et avec les versions de Turbo Pascal sous Dos') ;
   WriteLn ('le nom du fichier est composé au maximum de 8 caractères');
   Writeln ('Si le nom saisit est plus long, il sera tronqué.') ;
   WriteLn ;
   WriteLn ('Réspecter SVP les caractères composant un nom de fichier sous DOS.') ;
   WriteLN; WriteLn ;
   Repeat
       Write('Entrer le nom du fichier : '); ReadLn (nf);
       L := Length (nf) ;
   Until (L > 0) AND (L <=8 ) ;
   nf := 'C:\'+nf+'.fch' ; {le chemin du fichier }
   Assign (nl, nf) ;
   ReWrite(nl) ;
   Writeln ;
   WriteLn ('Le fichier ', nf, ' a été est crée avec succès ');
   Writeln ;
   ouvert := true ; { drapeau qui indique que le fichier est ouvert }
   Echap ;
END;

(********************Procédure d'ouverture d'un fichier******************)
PROCEDURE Ouverture (VAR nl : Fichier;VAR nf : nom_physique);
VAR   L : Byte ;
      Rep : Char ;
BEGIN
   ClrScr ;
   Writeln ('OUVERTURE D''UN FICHIER' : 50);
   WriteLn; WriteLn ;
   Repeat
      Repeat
          Write ('Entrer le nom du fichier : '); ReadLn (nf);
          L := Length (nf) ;
      Until (L > 0) AND (L <=8 ) ;
      nf := 'C:\'+nf+'.fch';
      Assign (nl, nf);
      {$I-} ReSet (nl); {$I+}
      IF IOResult <> 0 THEN
           Begin
```

```
              Writeln ;
              WriteLn ('Le fichier ', nf, ' n''existe pas ') ; Writeln ;
              Repeat
                Write ('Vouler vous réessayer O/N ? ') ;
                Rep := upcase(Readkey) ;
                WriteLn;
              Until Rep IN ['O' , 'N'] ;
              WriteLn ;
            End;
   Until (IOResult = 0) or (Rep = 'N');
   Writeln ;
   IF IOResult = 0 Then
        Begin
         ouvert := true ;
         WriteLn ('Le fichier ', nf, ' est maintenant ouvert ');
        End;
   Writeln ;
   Echap ;
END;

(*******************Procédure de suppression d'un fichier*****************)
PROCEDURE Suppression (VAR nl : fichier ; nf : nom_physique);
VAR  L: Byte ;
    Rep : Char ;
BEGIN
   ClrScr ;
   Writeln('SUPPRESSION D''UN FICHIER': 50);
   WriteLn; WriteLn ;
   Repeat
     Repeat
       Write ('Entrer le nom du fichier à supprimer : '); ReadLn (nf);
       L := Length (nf) ;
     Until (L > 0) AND (L <=8 ) ;
     nf := 'C:\'+nf+'.fch' ;
     Assign (nl, nf);
     {$I-} ReSet(nl); {$I+}
     IF IOResult <> 0 THEN
         Begin
           Writeln ; WriteLn ('Le fichier ', nf, ' n''existe pas '); Writeln ;
           Repeat
             Write ('Vouler vous réessayer O/N ? ');
              Rep := Readkey ; Rep := UpCase (rep);
              WriteLn;
           Until Rep IN ['O','N'] ;
           WriteLn ;
         End;
   Until (IOResult = 0) or (Rep = 'N');
   Writeln ;
   IF IOResult = 0 Then
          Begin
           Erase (nl) ;
           ouvert := false ; { Aucun fichier n'est ouvert }
           WriteLn ('Le fichier ', nf, ' a été supprimer avec succès ');
           Writeln ;
          End ;
   Echap ;
END;

(****************Procédure de fermeture d'un fichier*******************)
PROCEDURE Fermeture (VAR nl : fichier ; nf :nom_physique );
VAR   Rep : Char ;
BEGIN
  ClrScr ;
  Writeln ('FERMETURE D''UN FICHIER' : 50);
  WriteLn; WriteLn ;
  {$I-}
  Assign (nl, nf);
  ReSet(nl);
  {$I+}
  IF (IOResult <> 0) or (ouvert=false)
        THEN Begin
             Writeln ;
             if ouvert = false then WriteLn (' Le fichier n''est pas ouvert ')
                        Else WriteLn ('Le fichier ', nf, ' n''existe ');
             Writeln ;
           End
       Else Begin
            Repeat
              Write ('Vouler vous fermer Le fichier ', nf, ' (O / N) ? ' );
              Rep := Readkey ; Rep := UpCase (rep);
              WriteLn;
            Until Rep IN ['O','N'] ;
            IF Rep = 'O' Then Close (nl) ;
            Writeln ; WriteLn ('Le fichier ', nf, ' a été fermer avec succès ');
          End;
  Writeln ;
  ouvert := false;
  Echap ;
END;

(*********Procédure de recherche de l'existence et de la position d'un film***************)
PROCEDURE Cherche (ref2 : Word ; enr : Films; VAR nl : Fichier ; VAR posi : LongInt ; VAR trouve : Boolean);
BEGIN
  Trouve := False ;
  Reset (nl);
  While (NOT(EOF (nl))) AND (trouve = False) DO
     Begin
        Read(nl, enr);
        IF ref2 = enr.reference then
            Begin
               trouve := True ;
               posi := FilePos(nl) -1 ;
            End;
     End;
END ;

(******************Procédure d'ajout d'un film************************)
PROCEDURE Ajouter (VAR nl : Fichier ; enr : Films ;  nf : nom_physique );
VAR   rep : Char ; trouve:boolean ; posi:longint;
BEGIN
  ClrScr ;
  WriteLn ('A J O U T E R   F I L M S':20); WriteLn ;
  Assign (nl, nf);
  Repeat
     Repeat
         Repeat
            Write ('Référence : '); ReadLn (enr.reference);
            Cherche (enr.reference, enr, nl, posi, trouve);
            IF trouve = True Then
              Begin
                 WriteLn; WriteLn ('Cette référence existe déja dans le fichier ');
                 WriteLn('Veuillez entrer une autre référence ');
                 WriteLn;
              End;
         Until trouve = False ;
         Write ('Titre : '); ReadLn(enr.titre);
     Until (enr.reference > 0) AND (enr.titre <> '') ;
     Write(nl, enr);
     Write('Continuer Oui/Non (O/N)? ');
     Repeat
         Rep := Readkey ; Rep := UpCase (Rep);
     Until Rep IN ['O', 'N'] ;
     WriteLn; WriteLn ;
  Until rep = 'N' ;
END ;

(***************Procédure qui modifie les données d'un film****************)
PROCEDURE Modifier (VAR nl : Fichier ; enr : Films ; nf :nom_physique);
VAR rep : Char ; trouve:boolean ;  posi:longint;ref2:word;
BEGIN
  ClrScr ;
  WriteLn ('M O D I F I E R   F I L M S' : 20); WriteLn ;
  Assign (nl, nf);
  Repeat
     Write ('Référence : '); ReadLn (ref2);
     WriteLn ('Recherche de la référence : ', Ref2); WriteLn ;
     Cherche (ref2, enr, nl, posi, trouve);
     IF trouve = False
       THEN Begin
          Writeln ('La référence du film n''existe pas dans le fichier '); WriteLn;
         End
       Else Begin
```

```
                  Seek (nl, posi);
                  Read (nl, enr);
                  Repeat
                        Write ('Ancienne référence : ', enr.reference,' Nouvelle référence : ') ;
                        ReadLn (enr.reference);
                        Write ('Ancien titre : ', enr.titre , ' Nouveau titre : ');
                        ReadLn (enr.titre);
                  Until (enr.reference > 0 ) AND (enr.titre <> '');
                  Seek (nl, posi); Write(nl, enr);
                End;
         Write ('Continuer Oui ou Non (O/N)? ');
         Repeat
               Rep := Readkey ; Rep := UpCase (Rep);
         Until Rep IN ['O', 'N'] ;
         WriteLn; WriteLn ;
   Until rep = 'N' ;
END ;

(************Procédure de recherche et d'affichage d'un film**************)
PROCEDURE Chercher_afficher (VAR nl : Fichier ; enr : Films ; VAR nf :nom_physique);
VAR rep : Char ;
       trouve:boolean;
       posi:longint;ref2:word;
BEGIN
     ClrScr ;
     WriteLn ('C H E R C H E R   F I L M S' : 20); WriteLn ;
     Assign (nl, nf);
     Repeat
           Write ('Référence : '); ReadLn (ref2);
           WriteLn ('Recherche de la référence : ', Ref2); WriteLn ;
           Cherche (ref2, enr, nl, posi, trouve);
           IF trouve = False
               THEN Begin
                        Writeln('La référence du film n''existe pas dans le fichier '); WriteLn;
                   End
               Else Begin
                        Seek(nl, posi);
                        Read(nl, enr);
                        WriteLn('Référence : ', enr.reference) ;
                        WriteLn('Titre : ', enr.titre );
                   End;
           Writeln; Write ('Continuer Oui ou Non (O/N)? ');
           Repeat
                 Rep := Readkey ; Rep := UpCase (Rep);
           Until Rep IN ['O', 'N'] ;
           WriteLn; WriteLn ;
     Until rep = 'N' ;
END ;

(****************Procedure qui supprime un film**********************)
PROCEDURE Supprime (VAR nl : Fichier ; enr : Films ; nf :nom_physique);
VAR  rep : Char ;  taille, i : Word ; trouve : boolean ; posi : longint ; ref2:word;
BEGIN
    ClrScr ;
    WriteLn('S U P P R I M E R   F I L M S':20); WriteLn ;
    Assign (nl, nf);
    REPEAT
          Write ('Référence : '); ReadLn (ref2);
          WriteLn ('Recherche de la référence : ', Ref2); WriteLn ;
          Cherche (ref2, enr, nl, posi, trouve);
          IF trouve = False
              THEN Begin
                       Writeln('La référence du film n''existe pas dans le fichier '); WriteLn;
                  End
              Else Begin
                       Seek(nl, posi) ; Read (nl, enr);
                       WriteLn ('Suppression du film : ', enr.reference, ' = ', enr.titre);
                       taille := FileSize (nl);
                       For i:= posi+1 to taille-1 do
                           Begin
                                Seek (nl, i); Read (nl, enr);
                                Seek (nl, i-1); Write(nl, enr);
                           End;
                       Seek (nl, taille-1); Truncate (nl);
                       Reset(nl);
                       Taille := FileSize(nl);
```

```
            For i:=0 to taille-1 do
              Begin
                 Seek(nl, i) ; Read (nl, enr);
                 WriteLn ('Référence : ', enr.reference) ;
                 WriteLn ('Titre : ', enr.titre );
              End;
         End;
     Writeln;
     Write('Continuer Oui ou Non (O/N)? ');
     Repeat
        Rep := Readkey ; Rep := UpCase (Rep);
     Until Rep IN ['O', 'N'] ;
     WriteLn; WriteLn ;
  UNTIL rep = 'N' ;
END ;

(***********Procédure qui affiche un message d'erreur si le fichier n'est pas ouvert********)
PROCEDURE Erreur ;
BEGIN
  ClrScr ;
  Writeln (' E R R E U R ');
  Writeln ('Aucun fichier n''est ouvert ');
  Writeln ;
  Echap ;
END;

(*************Procédure du menu de choix*************************)
PROCEDURE Menu (VAR choix : Byte ) ;
VAR  Rep : char ;
   nl : Fichier;
   nf : nom_physique ;
   enr : Films ;
BEGIN
  REPEAT
     ClrScr ;
     WriteLn; Writeln ('___ M E N U ___ ' : 50);
     WriteLn ; WriteLn ;
     WriteLn (' F I C H I E R S                   E D I T I O N          Q U I T T E R'); Writeln;
     WriteLn (' 1 - Créer un nouveau fichier       5 - Ajouter un film        9 - Sortir');
     WriteLn (' 2 - Ouvrir un fichier existant     6 - Modifier un film ');
     WriteLn (' 3 - Supprimer un fichier           7 - Chercher un film ');
     WriteLn (' 4 - Fermer un fichier              8 - Supprimer un film '); Writeln ; WriteLn;
     Write (' Votre choix ? ');
     Repeat
        Rep := ReadKey;
     Until Rep IN ['1' .. '9'];
     Case Rep Of
       '1' : Creation (nl,nf) ;
       '2' : Ouverture (nl,nf) ;
       '3' : Suppression (nl, nf) ;
       '4' : Fermeture (nl, nf) ;
       '5' : If ouvert = true then Ajouter (nl, enr, nf) Else Erreur;
       '6' : If ouvert = true then Modifier(nl, enr,nf) Else Erreur;
       '7' : If ouvert = true then Chercher_afficher (nl, enr,nf) Else Erreur;
       '8' : If ouvert = true then Supprime (nl, enr,nf) Else Erreur;
       '9' : Begin
                ClrScr;
                Write(' A U R E V O I R ':50);
                If ouvert Then close(nl);
                Halt ; { Halt : procédure pré définie qui stoppe le programme et ramène au système }
             End;
     End ;
  UNTIL Rep = '9';
  If ouvert Then close(nl);
END;

{ =================== Programme Principal ======================= }
BEGIN
   ouvert := false ; { init : aucun fichier n'est ouvert }
   Menu (choix);
END.
```

LA RECURSIVITE

Exercice 1

```
Procedure  afficher ;
Var   n:integer;
begin
    for  n:=10  downto  1  do    writeln(n);
end;
========== Solution récursive ==========
Procedure  afficher (n:integer);
begin
    if   n >=1
          then  begin
                     writeln (n);
                     afficher (n-1);
                   end;
end;
L'appel : afficher (10) ;
```

Exercice 2

```
Procedure  affiche (a, b: integer);
begin
  if  (a<4)
      then  if  (b<10)
                   then    begin
                                writeln (a * 10 + b);
                                affiche (a, b + 1);
                             end
                   else    affiche (a + 1, 0);
end;

L'appel : affiche (0,0) ;
```

Exercice 3

```
Procedure  saisir (var n : byte);
begin
    writeln ('Taper un entier');
    readln (n);
    if   (n<1) or (n>31) or (n mod 2 <>0)
           then   saisir (n) ;
end_;
```

Exercice 4.2

```
Procedure          affiche     (T     :     tab     ;     n     :     integer);
begin
      if  n>=1   then  begin
                            write (T[n]:4);
                            affiche (T, n-1) ;
                         end ;
end;
=============solution 2===============
procedure affiche (T:tab ; i, n:integer);
begin
    if i<=n
       then   begin
                   write (T[i]:4);
                   affiche (T,i+1,n);
                end;
end;
L'appel : affiche (t, 1, n);
```

Exercice 4

```
Procedure   remplir (n : integer ; var T:Tab);
begin
    if  n>=1 then  begin
                          readln (T[n]);
                          if  T[n]<0
                               then remplir (n, T)
                               else   remplir (n-1, T) ;
                      end ;
end;
=============solution 2==============
Procedure  remplir (var T:tab ; i, n:integer);
begin
   if i<=n
      then begin
                readln (T[i]);
                if  T[i]<0
                     then  remplir (T, i, n)
                     else   remplir (T, i+1, n)
                 end;
end;

L'appel : remplir (t, 1, n);
```

Exercice 5

```
Function    fact (n : byte) : real ;
begin
  if   n=0    then   fact := 1
                else    fact := n * fact(n - 1) ;
end;
```

Exercice 6

```
Function  palindrome (ch: string): boolean;
begin
  if  (length(ch)<2)   then   palindrome := true
                             else   if (ch[1] = ch[length(ch)])   then    palindrome := palindrome (copy(ch, 2, length(ch) - 2))
                                                                          else    palindrome := false;
end;
============================== {solution 2} ==============================
Function  palindrome (ch: string): boolean;
begin
    if   length (ch) < 2
         then   palindrome := true
         else   palindrome := (ch[1] = ch[length(ch)])   and  (palindrome (copy(ch, 2, length(ch) - 2)));
end;
```

Exercice 7

```
Function  pair (n:integer) : boolean;
begin
    if  (n=0)   then   pair := true
                  else   if  (n=1)
                              then  pair := false
                              else   pair := pair (n-2) ;
end ;
************** autre solution ****************
if  (n=0)  or  (n=1)
```

```
===== Solution 2 : récursivité mutuelle =====
Function   pair (n:integer) : boolean;
      Function   impair (n:integer) : boolean;
      begin
          if  (n=0)   then   impair := false
                           else    impair := pair (n-1) ;
      end ;
begin
    if  (n=0)   then   pair := true
                    else    pair := impair (n-1) ;
          then  pair := (n=0)
          else  pair := pair (n-2) ;
```

```
end ;
```

Exercice 8

```
Function  prosom (p, q : integer) : integer;
begin
 if  (p=0) or (q=0)  then  prosom := 0
             else  if q=1 then  prosom := p
             else  prosom := p + prosom(p, q-1) ;
end ;
```

Exercice 9

```
Function  somme (n : integer) : longint;
begin
 if  (n=0)
     then  somme := 0
     else  somme := n + somme (n-1) ;
end ;
```

Exercice 10

```
Function  puissance (x :real ; n : integer) : real;
begin
 if (n=0)  then  puissance := 1
    else  if  (n<0)
          then  puissance := 1/puissance(x,-n)
          else  puissance := x*puissance (x, n-1) ;
end ;
```

1/x * puissance(x, n+1)

Exercice 11 **PGCD EUCLIDE**

```
Function  Pgcd ( a, b: Integer ): Integer;
begin
   if  (a=b) or (b=0)
     then  Pgcd := a
     else   Pgcd := Pgcd ( b, a mod b ) ;
end;
```

Exercice 11 **PGCD Différence**

```
Function  Pgcd (a, b : integer) : integer ;
begin
  if  (a=b)  then  pgcd := a
            else if  (a>b)
                  then  pgcd := pgcd (a-b, b)
                  else  pgcd := pgcd (a, b-a) ;
end;
```

Exercice 12

```
Function  ppcm (a, b, p : integer) : integer;
begin
 if  (p mod b)=0
     then  ppcm := p
     else  ppcm := ppcm (a, b, p+a) ;
 end;
```

L'appel sera : Writeln ('PPCM = ', ppcm (a, b, a));

Exercice 12

```
Function  ppcm (a, b, c : integer) : integer;
begin
 if  (a*c) mod b =0
     then  ppcm := a*c
     else  ppcm := ppcm (a, b, c+1) ;
 end;
```

L'appel sera : Writeln ('PPCM = ', ppcm(a, b, 1));

Exercice 13

```
Function  fibonacci (n : integer) : integer;
begin
  if  (n <= 2)
     then  fibonacci := 1
     else  fibonacci := fibonacci (n - 1)+fibonacci (n - 2);
end;
```

Exercice 14

```
Procedure  fact_prem (n, i : integer);
begin
   if (n > 1) then
      if (n mod i)=0
         then begin
                  Write(i,' ');
                  fact_prem(n div i, i);
              end
         else fact_prem(n, i + 1);
end;
```

l'appel sera : **fact_prem(n, 2);**

Exercice 15

```
Function   sommediv (i, n:integer):integer;
begin
  if i>n div 2
      then  sommediv := 0
      else if n mod i=0
              then  sommediv := sommediv (i+1, n)+i
              else  sommediv := sommediv (i+1, n);
end;
(**************** P.P ****************)
begin
  write ('N = ');readln (n);
  if  n = sommediv (1, n)
       then writeln (n,' est parfait')
       else writeln (n, ' n''est pas parfait');
end.
```

Exercice 16 **(1ere solution)**

```
Function  nb_div (i, n:integer):integer;
begin
    if (i >= n div 2)
       then   nb_div := 2
       else   if (n mod i=0)
              then  nb_div := nb_div (i+1, n)+1
              else  nb_div := nb_div (i+1, n);
end;
(***************** P.P *********************)
begin
 write ('N = '); readln (n);
 if  nb_div (2, n) = 2
      then  write (n,' est premier')
      else  write (n,' est non premier');
end.


(*************** 3eme solution **************)
function  nb_prem (i, n:integer):boolean;
```

Exercice 16 **(2eme solution)**

```
Function  exist_div (i, n:integer):boolean;
begin
    if (i >n div 2)
        then  exist_div := false
        else   if (n mod i=0)
                   then  exist_div := true
                   else   exist_div := exist_div (i+1, n);
end;
(******************* P.P *******************)
begin
 write ('N = '); readln (n);
 if   exist_div (2, n)
      then  write (n,' est non premier')
      else  write (n,' est premier');
end.
```

```
begin
    if  (n mod i<> 0)
        then  nb_prem:=nb_prem(i+1,n)
        else  if  (i = n)
                    then  nb_prem:=true
                    else  nb_prem:=false;
end;
```

L'appel :

```
  if  nb_prem(2,n)
        then  write (n,' est premier')
        else  write (n,' est non premier');
```

Exercice 17

```
Function  expn (x : real; n : integer) : real;
(*************************************)
function  fact (n : integer) : integer;
var    i, f : integer;
begin
    f := 1;
    for  i := 2 to n do   f := f * i;
    fact := f;
end;
function  puiss (x : real; n : integer) : real;
var   i : integer; p : real;
begin
    p := 1;
    for  i := 1  to  n  do   p := p * x;
    puiss := p;
end;
(*************************************)
begin
   if n=0   then   expn:=1
            else   expn:=expn(x,n-1)+ puiss(x,n) / fact(n);
end;
```

Exercice 20

```
Function   max (t:tab ; n:integer):integer;
var    m:integer;
begin
    if  n=1
        then   max := t[1]
        else   if  max (t, n-1)> t[n]
                     then  max := max (t, n-1)
                     else  max := t[n] ;
end;
============ Solution 2 ============
Function  max (t:tab ; m,i,n:integer):integer;
begin
    if  (i<=n)
        then  if  (t[i]>m)
                  then  m:=max(t, t[i], i+1, n)
                  else  m:=max(t, m, i+1, n);
     max:=m;
end;
```

L'appel : write (max(t, t[1], 2, n)) ;

Exercice 18

```
Function  som (n : integer):real ;
begin
    if  n=1
        then   som := 1
        else   som := 1/n + som (n-1);
end;
```

Exercice 19

```
Function  som (n : integer):real ;
Var   terme :real;  signe:integer;
begin
     if  n=0
        then  som:=1
        else  begin
                  if  n mod 2 = 0
                      then  signe := 1
                      else  signe :=-1;
                  terme:=signe * 1/(2*n+1) ;
                  som:= terme + som(n-1);
                end;
end;
```

Exercice 21

```
Function  plus_proche (M,n:integer ; T:tab ):integer;
begin
   if  (n=1)
      then  plus_proche:=T[1]
      else  if abs (M - T[n]) < abs(M - plus_proche(M,n-1,T))
               then   plus_proche := t[n]
               else   plus_proche := plus_proche(M,n-1,T);
end ;
```

Exercice 22

```
Function  cherche (elm, n : integer ; t:tab) : boolean ;
begin
   if  n=0
        then  cherche:=false
        else   if  elm = t[n]
                    then  cherche := true
                    else  cherche := cherche (elm, n-1, t);
end;
(*****************************************)
```

solution 2 :

```
Function  cherche (i, n, elm :integer ; t : tab):integer;
begin
   if  elm=T[i]
       then cherche:=i
       else  if  i=n
                 then cherche:=0
                 else cherche:=cherche (i+1, n, elm, t);
end;
```

Exercice 23

```
Function  Dichotomique
                   (deb, fin, val : integer ; t:tab) : boolean;
var   milieu:integer;
begin
    milieu:=(deb+fin) div 2;
    if val = T[milieu]
        then dichotomique := true
        else if (val < T[milieu]) and (deb < milieu)
        then
        dichotomique := dichotomique (deb, milieu-1,val,T)
        else if (val > T[milieu]) and (fin > milieu)
        then
        dichotomique := dichotomique (milieu+1, fin,val,T)
        else  dichotomique := false;
end;
```

Exercice 24

```
Procedure  Decaler  (Var t : Tab; p, n : integer);
Begin
     If p <= n Then
     Begin
          Decaler (t, p+1, n);
          t[p+1] := t[p];
     End;
End;
```

Exercice 25

```
Procedure  Inverser (Var t : Tab; p, n : Integer);
Var aux : Real;
Begin
     If p < n Then
     Begin
          aux := t[p];
          t[p] := t[n];
          t[n] := aux;
          Inverser (t, p + 1, n - 1);
     End;
End;
```

Exercice 26

```
Procedure  permuter (var  t : tab ; d, f : integer) ;
var aux:integer;
begin
    if   d < f  then
                    begin
                        aux:=t[d];
                        t[d]:=t[d+1];
                        t[d+1]:=aux;
                        permuter(t,d+2,f);
                     end;
end;
```

Appel : permuter (t, 1, n) ;

Exercice 27

```
program   maximum;
uses   wincrt;
var     a,b,acc:integer;
Function  max (a,b,acc:integer):integer;
begin
   if   a=0
        then  max:=b+acc
        else  if  b=0  then  max:=a+acc
                         else   max:=max(a-1, b-1, acc+1);
end;
begin
   write('a = ');readln(a);
   write('b = ');readln(b);
   writeln('La valeur max = ',max(a,b,0));
end.
```

Exercice 28

```
Function   inverse (st: string): string;
begin
      if   st = ''   then inverse := ''
                     else  inverse := st[length(st)] +
                                  inverse(copy(st, 1, length(st) - 1));
end;
```

============ Solution 2 ==============

```
Procedure  inverse (var st: string);
Var   c: char;
begin
        if   st <> ''  then    begin
                                  c := st[length(st)];
                                  delete(st, length(st), 1);
                                  inverse(st);
                                  st := c + st;
                                end;
end;
```

Exercice 29

```
Procedure  tri_selection (n:integer; var  T:tab);
var    p:integer;
  (*************************)
  function  posmax (n:integer; t:tab):integer;
  var  max, j:integer;
  begin
       max:=1;
       for  j:=2  to n  do
               if t[j]>t[max]  then max:=j;
        posmax := max;
  end;
  (*************************)
  procedure permuter (x,y:integer; var t:tab);
   (*************************)
begin
   If  n>1  then
      begin
         p := posmax (n, t);
         if  p<>n   then  permuter (n, p, t);
         tri_selection (n-1, t);
      end;
end;
```

Appel : tri_selection (n, T);

=============solution 2================

```
Procedure  tri_selection (deb,n:integer ; var T:tab);
var  pos:integer;
  (*************************)
  function  posmax(deb,n:integer;T:tab):integer;
  var max,j:integer;
  begin
     max:=deb;
     for j:=deb+1  to n do
             if T[j]>T[max]  then max:=j;
     posmax:=max;
  end;
  (*************************)
  procedure permuter (var  x,y:integer);
  (*************************)
begin
  if  deb<>n  then
      begin
         pos:=posmax (deb,n,T);
         if pos<>deb then permuter (T[deb], T[pos]);
         tri_selection (deb+1, n, T);
      end;
end;
```

Appel : tri_selection (1, n, T);

Exercice 30

```
Procedure  tri_insertion (n:integer ; var t:tab);
(**********************************)
 procedure  inserer (n:integer ; var t:tab);
 var  tmp,j:integer;
 begin
   if   t[n]<t[n-1]  then
      begin
        tmp:=t[n];   j:=n;
        while  (j>1) and (t[j-1]>tmp)  do
            begin
               t[j]:=t[j-1];
               j:=j-1;
            end ;
        t[j]:=tmp;
      end;
 end ;
(*******************************)
begin
  if  n>=2   then   begin
                       tri_insertion (n-1, t);
                       inserer (n, t);
                   end;
end;
```

==============solution 2=================

```
Procedure  tri_insertion (i, n:integer ; var t:tab);
var   j, tmp : integer ;
begin
 if   (i<=n)
       then  begin
                  if t[i]<t[i-1]
                       then begin
                                tmp:=t[i];
                                j:=i ;
                                while  (t[j-1]>tmp) and (j>1)  do
                                        begin
                                          t[j]:=t[j-1];
                                          j:=j-1 ;
                                        end;
                                 t[j]:=tmp;
                                end;
                    tri_insertion(i+1, n, t);
                  end;
end;
```

Appel : tri_insertion (2, n, t);

Exercice 31

```
Procedure  tri_bulles
                    (var t:tab ; n:integer ; permut:boolean);
var i, aux : integer;
begin
  if (n>1)
      then begin
            for  i:=1  to  n-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;
            if (permut=true)  then  tri_bulles (t, n-1, false);
           end;
end;
```

Appel : tri_bulles (t, n, false);

=========== solution 2=================

```
Procedure  tri_bulles (var t : tab ; n, i, j : integer);
(*************************)
 procedure permut (var  x,y:integer);
(*************************)
begin
   if   (i<=n-1)
         then  begin
                  if  (j<=n-i)
                        then  begin
                                 if (t[j]>t[j+1])
                                        then  permut (t[j], t[j+1]);
                                 tri_bulles (t, n, i, j+1);
                              end ;
                tri_bulles (t, n, i+1, 1);
              end;
end;
```

Appel : tri_bulles (t, n, 1, 1);

Exercice 32

```
Procedure Tri_Shell_Rec  (Var t:TAB; n, pas :integer);
Var aux,i : integer;
begin
    If pas > 0 Then
    Begin
        If n > pas Then
            begin
                Tri_Shell_Rec (t, n - pas, pas);
                If t[n] < t[n - pas] Then
                Begin
                  aux:= t[n];
                  i := n;
                  Repeat
                     t[i] := t[i - pas];
                     i := i - pas;
                  Until (i = pas) or (aux > t[i - pas]);
                  t[i] := aux;
                End;
            End;
        Tri_Shell_Rec (t, n, pas Div 3);
    End;
End;
```

Exercice 33

```
Procedure  tri_fusion  (d, f :integer ; var T:tab);
var  m : integer;
(**************************)
procedure fusionner(var t:tab;d,m,f:integer);
var  temp:tab;    i,j,k:integer;
begin
   i:=d;
   j:=m+1;
   for k:=d to f do
     if (j>f) or((i<=m) and (T[i]<T[j]))
           then  begin
                    temp[k]:=T[i];
                    i:=i+1;
                  end
           else  begin
                    temp[k]:=T[j];
                    j:=j+1;
                  end ;
   for k:=d to f do  T[k]:=temp[k];
end;
(*************************)
Begin
    if   f>d  then
       begin
         m:=(d+f) div 2;
         tri_fusion (d,m,t);      (* trier partie gauche *)
         tri_fusion (m+1,f,t);  (* trier partie droite *)
         fusionner (t,d,m,f);     (* fusionner *)
       end ;
End;
```

Exercice 34

```
Procedure supprim(c:char ; n:integer ; var ch:string);
begin
  if  n<>0 then
           if ch[n]=c
              then begin
                     delete(ch,n,1);
                     supprim (c, n-1, ch);
                   end
              else  supprim (c, n-1, ch);
end;
```

Exercice 35

```
Function  nb_occ (n:byte ; ch:string ; c:char):byte;
begin
  if   n=0  then  nb_occ:=0
            else  if ch[n]=c
                  then  nb_occ:=nb_occ (n-1, ch, c)+1
                  else   nb_occ:=nb_occ (n-1, ch, c)
end;
```

Appel : writeln(nb_occ (length(ch), ch, c)) ;

Exercice 36

```
Function  sc (n:integer):integer;
begin
   if   n<10
       then  sc := n
       else   sc := (n mod 10) + sc (n div 10);
end;
```

Exercice 37

```
Function  nbc (n:integer):integer;
begin
  if   n <10
         then  nbc := 1
         else   nbc := 1+ nbc (n div 10);
end;
```

Exercice 38

```
Function  anagram (mot1,mot2:string):boolean;
Var   p:integer;
Begin
       if (mot1='') and (mot2='')
          then anagram:=true
          else begin
                 p:=pos(mot1[1],mot2);
                 if (mot1<>'')and(p>0)
                      then begin
                             delete(mot1,1,1);
                             delete(mot2,p,1);
                             anagram:=anagram(mot1,mot2);
                           end
                      else anagram:=false;
               end;

end;
```

Exercice 39

```
Function  teste (i:integer ; c:char; ch: string):boolean;
begin
       if   c = ch[i]
             then   teste:= true
             else   if  i < length(ch)
                        then  teste := teste (i+1, c, ch)
                        else  teste := false;
end;
============= solution 2 ==============
Function  teste (c:char ; ch:string):Boolean;
begin
  if (ch='') or ((length(ch)=1)and(ch[1]<>c))
       then teste:=false
       else if ch[1]=c
            then  teste:=true
            else   teste:= teste (c, copy(ch,2,length(ch)-1));
end;
```

Exercice 40

```
Function  suppr_car (c:char;ch : string):string;
begin
  case length(ch) of
     0 : suppr_car:='';
     1 : if ch[1]=c  then suppr_car:=''
                     else suppr_car:=ch;
     else if ch[1]=c
              then suppr_car:=copy(ch,2,length(ch)-1)
              else suppr_car := copy(ch,1,1) +
                      suppr_car(c,copy(ch,2,length(ch)-1));
  end;
end;
```

Exercice 41

```
Function  Constructible (mot,lettres : string):boolean;
    function suppr_car(c:char;ch : string):string;
    function teste (c:char ; ch:string):Boolean;
begin
    if mot=''
       then constructible:=true
       else if teste(mot[1],lettres)
              then if length(mot)=1
                     then constructible:=true
                     else constructible :=
                          constructible(copy(mot,2,length(mot)-1),
                                          suppr_car(mot[1],lettres))
              else constructible:=false;
end;
```

Exercice 42

```
Function   val_triangle (c, l: integer):integer;
begin
  if (c=1) or (l=c)
      then  val_triangle := 1
      else  val_triangle := val_triangle(c,l-1)+
                                     val_triangle (c-1,l-1);
end;
```

Exercice 43

```
Function   MacCarthy (n: integer): integer;
begin
   if  n >100
        then   MacCarthy := n - 10
        else   MacCarthy := MacCarthy (MacCarthy (n+11));
end;
```

Exercice 44

```
Function   Ack (m, n : integer) : integer;
begin
  if   m = 0
      then   Ack := n+1
      else   if  n = 0
             then   Ack := Ack (m-1,1)
             else    Ack := Ack (m-1, Ack (m, n-1))
end;
{======Implémentation itérative=======}
Function    ack (m, n :integer):integer;
begin
     while   m <> 0  do
        begin
          if  n = 0   then   n := 1  else   n := ack (m, n - 1) ;
          m := m – 1;
```

Exercice 45

```
Function    eval (st: string): integer;
var   a, i, e: integer;
begin
  if  length(st) = 0
        then   eval := 0
        else  begin
                i:=1;
                repeat
                   i:=i+1;
                until (st[i] in ['+','-']) or (i>length(st));
                val (copy(st, 1, i - 1),a,e);
                delete(st, 1, i-1 );
                eval := a + eval (st);
              end;
end;
```

```
       end;
      ack:= n + 1;
end;
```

Exercice 46

```
Program  mini_cal;
uses  wincrt;
var    e:integer;  st:string ;
Function  eval0(st: string): integer;
var   a, i: integer;  signe: char;
begin
   i := length(st);
   repeat
      i := i-1
   until  (i = 0) or (st[i] in ['*', '/']);
   val (copy(st, i + 1, length(st) - i), a, e);
   delete(st, i, length(st) - 1 + i);
   if  st[i] = '*'
       then  eval0 := eval0 (st) * a
       else  if  st[i] = '/'
            then  eval0 := eval0 (st) div a
            else  eval0 := a;
end;
```

```
Function   eval (st: string): integer;
var   a, i, e: integer;
begin
 if  length(st) = 0 then    eval := 0
     else  begin
                i:=1;
                repeat
                  i:=i+1;
                until (st[i] in ['+','-']) or (i>length(st));
                a:=eval0(copy(st, 1, i - 1));
                delete(st, 1, i-1 );
                eval := a + eval(st)
              end;
end;
begin
  write ('Entrer une chaîne formée d''une somme de
        nombre  :'); readln (st);
  write( 'La somme =  : ', eval (st));
end.
```

Exercice 48

```
Function    valeur(c: char): integer;
begin
  case  c  of
    'M': valeur := 1000;
    'D': valeur := 500;
    'C': valeur := 100;
    'L': valeur := 50;
    'X': valeur := 10;
    'V': valeur := 5;
    'I': valeur := 1;
  end;
end;
function  eval(s: string): integer;
var  n1: integer;
begin
  if length(s) = 1
      then   eval := valeur(s[1])
      else  begin
               n1 := valeur (s[1]);
               if  n1 < valeur (s[2])  then n1 := -n1;
               eval := n1 + eval (copy(s, 2, length(s) - 1));
             end;
end;
```

Exercice 47

```
Procedure   combinaison(ch, tete: string);
var    i: integer;
begin
  if   length(ch) = 1
       then   begin
                  ch:=tete+ch;
                  writeln(ch)
                end
       else
       for  i := 1  to  length(ch)  do
         begin
          combinaison(copy(ch, 2, length(ch)-1), tete+ch[1]);
          ch := copy(ch, 2, length(ch)-1) + ch[1] ;
          end ;
end;
```

Appel : combinaison (ch, '');

Exercice 49

```
Program    Chaine;
Uses    Wincrt ;
Var     ch: String ;
(****************************************)
Procedure  affiche (ch:string);
begin
   if  ch <> ''
      then   begin
                Writeln (ch);
                Affiche (copy(ch, 1, length(ch)-1));
             end;
end;
(****************************************)
Begin
   Readln (ch);
   Affiche (ch);
End.
```

STRUCTURE GENERALE D'UN PROGRAMME PASCAL

PROGRAM Nom_programme ; *{En-tête du programme}**
Uses ... ; *{Utilisation des unités / bibliothèques}**
Const ... ; *{Déclaration des constantes}**
Type ... ; *{Déclaration des types}**
Var ... ; *{Déclaration des variables}**

*{================ Définition des procédures ========================}**
Procedure Nom_procédure (pf_1 : $type_1$; **Var** pf_2 : $type_2$; … ; pf_n : $type_n$) ;
  *{Déclarations locales : **Const, Type, Var, Function, Procedure**, ...}**
Begin
    Instructions de la procédure ;
End ;

*{================= Définition des fonctions ========================}**
Function Nom_fonction (pf_1 : $type_1$; pf_2 : $type_2$; … ; pf_n : $type_n$) : Type_résultat ;
  *{Déclarations locales : **Const, Type, Var, Function, Procedure**, ...}**
Begin
    Instructions de la fonction ;
    Nom_fonction := résultat ;
End ;

*{======================== **P. P.** ================================}*
BEGIN *{Début du programme principal}*
    Instructions ;
    ………………………………… ;
    ………………………………… ;
  {Bloc principal du programme avec appel des procédures et des fonctions}
END. *{Fin du programme}*

*** : facultatif**