{Séance 1 - 8h-9h}

```
{#Question 1#
Program Nomination;
Uses wincrt;
type tab=array[1..20]of string[10];
var i,N:integer;
          Nom:tab;
Begin
WriteLn('Donner le nombre de produits:');readln(N);
for i:=1 to N Do
Begin
WriteLn('Donner le nom du produit n°:',i,':');readln(Nom[i]);
End;
Spirale(Nom,N);
end.}

{#Question 2#}
Program Nomination;
Uses wincrt;
type tab=array[1..20]of string[10];
var N:integer;
          Nom,TP:tab;
Procedure Saisie(Var T:Tab;Var N:integer);
var i:integer;
Function Valide(ch:String):boolean;
var i:integer;
Begin
Valide:=true;
for i:=1 to Length(ch) Do
If Not(Ch[i] in ['A'..'Z']) then Valide:=False;
end;
Begin
repeat
WriteLn('Donner le nombre de produits:');readln(N);
Until N in [3..20];
for i:=1 to N Do
repeat
WriteLn('Donner le nom du produit n°:',i,':');readln(T[i]);
Until(Length(T[i])=10) And Valide((T[i]));
end;
Procedure Spirale(T:Tab; Var TP:Tab; N:integer);
var i,p:integer;
                 CH:String;
Function Rotation(CR:String):String;
Var CHX:String;
                 i:integer;
Begin
CHX:='';
For i:= 1 to length(CR)Div 2 Do
CHX:=CHX+CR[Length(CR)-i+1]+CR[i];
if Length(CR)Mod 2 =1 Then
```

```
CHX:=CHX+CR[Length(CR)Div 2 +1];
Rotation:=CHX;
End;
Begin
Repeat
WriteLn('Donner un entier P:');readln(P);
Until P in [1..10];
Ch:=''; for i:=1 to N Do
                                                CH:=CH+T[i][P];
TP[1]:= CH;
for i:=2 to N Do
TP[i]:=Rotation(TP[i-1]);
end;
Procedure Affichage(T:Tab; N:integer);
Var i:integer;
Begin
For i:=1 to N Do
Write(T[i],' - ');
end;
Begin
Saisie(Nom,N);
Spirale(Nom,TP,N);
Affichage(TP,N);
end.
```

{ Séance 2 9h30 - 10h30

```
# Question 1 #
Program TirArc;
Uses WinCrt;
Type Tab= Array[1..20]Of String[30];
Var n,i:integer;
                   A:Tab;
Begin
Repeat
Write('Donner le nombre de joueurs :'); Readln(N);
Until(N in [2..20]);
For i:=1 To N Do
Repeat
Write('Donner le nom du joueur n°',i,' : '); Readln(A[i]);
Until Alpha(A[i]);
Score(A,N);
end.

# Question 2 #}
Program TirArc;
Uses WinCrt;
Type Tab= Array[1..20]Of String;
                    Tbs= Array[1..20]Of integer;
Var n:integer;
                   A:Tab;
Procedure Remplir(Var A:Tab;N:integer);
```

```
    Var i:integer;
    Function Alpha(ch:String):boolean;
        var j:integer;
        ok:Boolean;
        Begin
            ok:=true;
            for j:=1 to Length(ch) Do
            If Not(Upcase(CH[j]) in ['A'..'Z',' ']) then ok:=False;
            Alpha := (ok) And (Length(CH)in[1..30]);
        end;
    Begin
    For i:=1 To N Do
        Repeat
            Write('Donner le nom du joueur n°',i,' : '); Readln(A[i]);
        Until Alpha(A[i]);
    end;
Procedure Score(A:Tab;N:integer);
    Var S:Tbs;
            i,j,E:integer;
    Procedure Tri(Var A:Tab;Var S:TBS;N:integer);
        Var aux1:String;
         aux2,i:integer;
      echange:boolean;
                Begin
                repeat
           echange:=false;
           for i:=1 to n-1 do
           if S[i]<S[i+1] then
           begin
           aux1:= A[i];
           A[i]:=A[i+1];
           A[i+1]:=aux1;
           aux2:=S[i];
           S[i]:=S[i+1];
           S[i+1]:=aux2;
           echange:=true;
           end;
           n:=n-1;
         until(echange=false)or(n=1);

                End;
    Begin
    For i:=1 To N Do
        Begin
        S[i]:=0;
        For j:= 1 To 3 Do
            Begin
                Write('Donner l''essai n° ',j,' du joueur n° ',i,':');
                Repeat
                    Readln(E);
                Until (E In [0..10]);
                S[i]:=S[i]+E;
            end;
        WriteLn('Score joueur ',i,' = ',S[i]);
        End;
    Tri(A,S,N);
    WriteLn('Le tableau du score: ');
    For i:=1 To n Do
            WriteLn(A[i],' avec un score de ',S[i]);
    End;
Begin
Repeat
Write('Donner le nombre de joueurs :'); Readln(N);
Until(N in [2..20]);
Remplir(A,N);
Score(A,N);
end.
```

{séance3 -11h -12h

```
#Question 1#
Program Plagiat;
uses wincrt;
Var T1,T2:String;
Nbc:integer;
Begin
Repeat
Readln(T1);
Readln(T2);
Until (Nbmots(T1) = Nbmots(T2));
Nbc:=Commun(T1,T2);
end.}
{# Question 2#}
Program Plagiat;
uses wincrt;
Var T1,T2:String;
Nbc:integer;
procedure Saisir(Var CH:String);
Function Valide( X:String):boolean;
var i:integer;
        ok:boolean;
Begin
ok:=True;
for i:=1 to length(X)-1 Do
If Not(X[i] in ['A'..'Z',' ']) then ok:= False;
Valide:=(ok)And(X[1]in['A'..'Z']) And(X[length(X)]='.')And(Length(X)<=200);
end;
Procedure superflus(Var X:String);
var p:integer;
Begin
Repeat
p:=pos('  ',X);
delete(X,p,1);
Until (pos('  ',X)=0);
```

```
end;
Begin
repeat
write('Donner un texte :');
Readln(CH);
until(Valide(CH));
superflus(CH);
end;
Function Nbmots(CH:string):integer;
Var NB,i:integer;
Begin
NB:=0; For i:=1 to Length(CH) Do
If CH[i] in [' ','.'] Then NB:=NB+1;
Nbmots:=NB;
end;
Function Commun(CH1,CH2:String):integer;
Var com,i,j:integer  ;
Begin
i:=1; j:=1;com:=0;
write('les mots communs aux deux textes sont :');
Repeat
If CH2[i]in[' ','.'] Then
Begin
If pos(copy(CH2,j,i-j),CH1)<>0 Then
begin
com:=com+1;
write(copy(CH2,j,i-j),' - ');
end;
i:=i+1;
j:=i;
End
else
i:=i+1;
until(i>Length(CH2));
WriteLn;
Commun:=com;
end ;
Begin
Repeat
Saisir(T1);
Saisir(T2);
Until (Nbmots(T1) = Nbmots(T2));
Nbc:=Commun(T1,T2);
If Nbc >Nbmots(T1) Div 2 Then WriteLn ('La chaine T2 est plagiat de T1')
else  WriteLn ('La chaine T2 est non plagiat de T1')
end.
```

{Séance 4 12h30 - 13h30

```
#Question 1#
Program fidelite;
Uses wincrt;
type tab=array[1..30]of string[10];
var N:integer;
 TA:tab;
Begin
Repeat
WriteLn('Donner le nombre des adhérants:');readln(N);
Until N in [5..30];
Remplir(TA,N);
Bonus(TA,N);
end.}

{#Question2 #}
Program fidelite;
Uses wincrt;
type tab=array[1..30]of string[10];
var N:integer;
 TA:tab;
Procedure Lecture(Var N:integer);
Begin
Repeat
WriteLn('Donner le nombre des adhérants:');readln(N);
Until N in [5..30];
end;
Procedure categorie(Var C:Char);
Begin
repeat
WriteLn('Donner la catégorie de l''adhérant ');readln(C);
Until C in ['A','J','E'];
end;
Function Valide(ch:String):boolean;
var i:integer;
Begin
Valide:=true;
for i:=1 to Length(ch) Do
If Not(Ch[i] in ['0'..'9']) then Valide:=False;
end;
Function recherche(T:Tab;n:integer;y:string):boolean;
{cette module n'est pas demandée dans l'examen puisqu'on a supposé que les numéros sont
distincts}
Var i :integer;
ok:boolean;
Begin
if n=0 then ok:=false
Else
Begin
ok:=false;
i:=0;
```

```
Repeat
i:=i+1
until(copy(T[i],8,3)=y) or (i=n) ;
ok:= (copy(T[i],8,3)=y);
end;
recherche:=ok;
end;
procedure remplir(Var T:Tab; N:integer);
var i,A,M:integer;
c:char;
CHA,CHM,Num:string ;
begin
for i:=1 to N Do
begin
Writeln('Donner les coordonnées de l''adhérant n° ', i,':');
categorie(C);
repeat
WriteLn('Donner l''année d''adhésion: ');readln(A);
Until (A>=2000) and (A <= 2019);
STR(A,CHA);
repeat
WriteLn('Donner le mois d''adhésion: ');readln(M);
Until (M in [1..12]);
Str(M,CHM);
if M < 10 Then CHM:='0'+CHM;
repeat
WriteLn('Donner le mnuméro d''adhésion: ');readln(Num);
Until (Recherche(T,i-1,Num)=false)and(valide(Num))And(Length(Num)=3);
T[i]:=Concat(C,CHA,CHM,NUM);
end;
end;
Procedure Bonus(T:Tab; N:integer);
var C:Char;
                CH:String ;
                i,a,m,e,B:integer;
Begin
categorie(C);
For i:=1 to N Do
If T[i][1]=C Then
        Begin
        Val(Copy(T[i],2,4),a,e);
        Val(Copy(T[i],6,2),m,e);
        B:=0;
        if(2019-a>=5)and(5-m>=1) Then
                Begin
                CH:='';
                Case C Of
                'A':CH:='Adulte';
                'J':CH:='Junior';
                'E':CH:='Enfant';
                End;
  B:= (2019-a)*12+m;
```

```
        Writeln('Pour la catégorie ',CH,', L''adhérant qui a l''abonnement n° ',T[i],' a un bonus de ',B,' heures.');
        end;
    end;
End;
Begin
Lecture(N);
Remplir(TA,N);
Bonus(TA,N);
End.
```

{Séance 5 14h - 15h

```
#Question 1#

Program HeureMiroir;
Uses WinCrt;
Type Tab=Array[1..20]Of String[30];
Var NP,HN:Tab;
        N:Integer;
Begin
Write('Donner le nombre des nouveau-nés: '); Readln(N);
Remplissage(NP,HN,N);
Affichage(NP,HN,N);
End.

# Question 2 #}
Program HeureMiroir;
Uses WinCrt;
Type Tab=Array[1..20]Of String[30];
Var NP,HN:Tab;
        N:Integer;
Procedure Lecture(Var N:integer);
Begin
    Repeat
    WriteLn('Donner le nombre des nouveau-nés :');readln(N);
    Until N in [5..30];
End;
Procedure Remplissage(Var P,H:Tab;N:integer);
var i,HH,MM:integer;
        CHH,CHM:String;
        Function Valide(ch:String):boolean;
        var j:integer;
        ok:Boolean;
        Begin
            ok:=true;
            for j:=1 to Length(ch) Do
            If Not(Upcase(CH[j]) in ['A'..'Z',' ']) then ok:=False;
            Valide := (ok) And (CH[1] in ['A'..'Z']) And (Length(CH)in[1..30]);
        end;
Begin
For i:=1 to N Do
```

```
        Begin
        Repeat
        Writeln('Donner le nom du nouveau-né n° ',i);Readln(P[i]);
        Until(Valide(P[i]));
        repeat
                WriteLn('Donner l''heure de naissance: ');readln(HH);
        Until (HH in[0..23]);
        STR(HH,CHH);
        if HH < 10 Then CHH:='0'+CHH;
        repeat
                WriteLn('Donner la minute de naissance: ');readln(MM);
        Until (MM in [0..59]);
        Str(MM,CHM);
        if MM < 10 Then CHM:='0'+CHM;
        H[i]:= Concat(CHH,':',CHM);
        End;
End;
Procedure Affichage(P,H:Tab;N:integer);
var i,j,DB,TR,IV:integer;
                test:Boolean;
        Function Inverse(CH:String):Boolean;
        var j:Integer;
        Begin
                Inverse:=True;
                For j:=1 To Length(CH) Do
                If(CH[j]<>CH[Length(CH)-j+1]) Then Inverse:=False;
        end;
        Function Triple(C:Char;CH:String):Boolean;
        var j,NB:integer;
        Begin
                NB:=0;
                For j:=1 To Length(CH) Do
                If(CH[j]=C) Then NB:=NB+1;
                Triple:=(NB>=3);
        End;
        Function Doublee(Ch:String):Boolean;
        Begin
                Doublee:=(Copy(CH,1,2)=Copy(CH,4,2));
        end;
Begin
DB:=0;TR:=0;IV:=0;
For i:=1 To n Do
        Begin
                If Doublee(H[i]) Then
                Begin
                Writeln(P[i],' : heure miroir doublée');
                DB:=DB+1;
                End;
                If Inverse(H[i]) Then
                Begin
                Writeln(P[i],' : heure miroir inversée');
                IV:=IV+1;
                End;
                j:=1;repeat
                test:=Triple(H[i][j],H[i]) ;
                If test  Then
                Begin
                    Writeln(P[i],' : heure miroir triplée');
                    TR:=TR+1;
                End
                Else
                j:=j+1;
                Until (j>2)Or(Test=True);
        End;
Writeln('Le nombre des nouveau-nés ayant l''heure de naissance miroir doublée est ',DB);
Writeln('Le nombre des nouveau-nés ayant l''heure de naissance miroir triplée est ',TR);
Writeln('Le nombre des nouveau-nés ayant l''heure de naissance miroir inversée est ',IV);
End;
Begin
Lecture(N);
Remplissage(NP,HN,N);
Affichage(NP,HN,N);
End.
```