Examen du Baccalauréat 2017 - Session Principale

✣ ✣ ✣

Section Sciences de l'Informatique

✣ ✣ ✣

Algorithmique et Programmation

Corrigé

Proposé par : Moez Romdhane (Moez.Romdhane@gmail.com - fb/MoezRom)

Table des Matière

Exercice 1

1) La fonction **Rectangle** permet de calculer l'aire résultante de la courbe de la fonction **f** sur un intervalle [a,b] selon la méthode des:

V	**rectangles à gauche**
F	**rectangles du point milieu**
F	**rectangles à droite**

2) Pour les valeurs **a = 1**, **b = 5** et **n = 4**, le résultat retourné par la fonction **Rectangle** est :

F	**5.5**
V	**7.7**
F	**10.12**

3) Pour appliquer la méthode des trapèzes au lieu de la méthode des rectangles, on remplace l'instruction de calcul de la somme S par :

V	**S← S + (6/(1+x) + 6/(1+x+h))/2**
F	**S← S + 6/(1+x+h))/2**
F	**S← S + (6/(1+x) - 6/(1+x+h))/2**

Exercice 2 Version Itérative

```
1) Fonction PartieDecimaleEnBinaire(x:Réel, N:Entier):Chaîne
2) Result ← "0."
3) Pour i De 1 À N Faire
        x ← x * 2
        Si x < 1 Alors
              Result ← Result + "0"
        Sinon
              Result ← Result + "1"
              x ← x - 1
        Fin Si
   Fin Pour
4) PartieDecimaleEnBinaire ← Result
5) Fin PartieDecimaleEnBinaire
```

TDO	
Result	Chaîne
i	Entier

Exercice 2 Version Récursive

```
1) Fonction Binaire(AjouterZeroVirgule: Booléen, x:Réel, N:Entier):Chaîne
2) Si AjouterZeroVirgule Alors Binaire ← "0." + Binaire(Faux, x, N)
   Sinon Si N=1 Alors Binaire ← Chr(48+Ord(X*2≥1)) { 48 peut être remplacé par Ord("0") }
   Sinon Binaire ← Chr(48+Ord(X*2≥1)) + Binaire(Faux,Frac(x*2),N-1)
   Fin Si
3) Fin Binaire
                { Appel de la Fonction: Binaire(Vrai, x, N) }
```

TDO	
?	??

Le dressage du TDO dans cet exercice n'a aucun sens, et il n'est mis ici qu'en réponse aux exigences énoncées dans l'exercice: ***"Chaque algorithme proposé DOIT ÊTRE accompagné d'un TDO"***

Exercice 3

Procédure GenererFichierResultat

```
1) Procédure GenererFichierResultat
2) Associer(FSource, "Source.txt")
3) Ouvrir(FSource)
4) Associer(FResultat, "Resultat.txt")
5) Recréer(FResultat)
6) Tant Que Non Fin_Fichier(FSource) Faire
        Lire(FSource, A)
        Lire(FSource, B)  {Ou encore: Lire(FSource, A, B)}
        {Ne pas oublier que le séparateur entre les entiers à lire avec Read ou Readln peut être:
        Un espace ou un retour à la ligne ou une tabulation, et ceci pour les fichiers textes ou même
        à partir du clavier, donc les Entiers dans notre fichier sont prêts pour la lecture successive}
        Si Factorielle(A) * Factorielle(B) MOD (A + B) Dans [A, B] Alors
             Écrire_nl(FResultat , A , "+i*" , B) {On Enregistre dans le Fichier}
             Écrire_nl(A , "+i*" , B) {Et on affiche directement sur l'écran ;) }
        Fin Si
   Fin Tant Que
7) Fermer(FSource)
8) Fermer(FResultat)
9) Fin GenererFichierResultat
```

TDO	
Factorielle	Fonction
FSource, FResultat	Texte

Fonction Factorielle

```
1) Fonction Factorielle(X:Entier):Entier
2) Si X < 2 Alors Factorielle ← 1
   Sinon Factorielle ← X * Factorielle(X-1)
   Fin Si
3) Fin Factorielle
```

Problème : Corrigé Version 01

Analyse Du Programme Principal

```
{N⇒ NbMots}
Nom: ProblèmeVersion01
Résultat=  [Associer(FChemins,"C:/Chemin.txt")
           SaisirLesMotsARechercher(NbMots, TM)
           RemplirMatrice(NbMots, NbFichiers, TM, FChemins, M)]
           Afficher(NbMots, NbFichiers, TM, FChemins, M)
Fin ProblèmeVersion01
```

TDNT	
Tab	Tableau[1..10] de Chaîne
Mat	Tableau[1..100,1..10]de Octet

TDO	
NbMots, NbFichiers	Octet
TM	Tab
M	Mat
FChemins	Fichier Texte
SaisirLesMotsARechercher, RemplirMatrice, Afficher	Procédure

Procédure SaisirLesMotsARechercher

```
1) Procédure SaisirLesMotsARechercher(Var NbMots: Octet, Var TM:Tab)
2) Répéter
        Lire(NbMots)
   Jusqu'à NbMots Dans [1..10]
3) Pour i De 1 À NbMots Faire
        Répéter
            Lire(TM[i])
        Jusqu'à TM[i]≠"" Et Alphabetique(TM[i])
   Fin Pour
4) Fin SaisirLesMotsARechercher
```

TDO			
Alphabetique	Fonction	i	Octet

Fonction Alphabetique

```
1) Fonction Alphabetique(Ch:Chaîne):Booléen
2) i ← 0
3) Répéter
        i ← i+1
   Jusqu'à Non Majus(Ch[i]) Dans ["A".."Z"] Ou i=Long(Ch)
4) Alphabetique ← Majus(Ch[i]) Dans ["A".."Z"]
5) Fin Alphabetique
```

TDO	
i	Octet

Procédure RemplirMatrice

```
1) Procédure  RemplirMatrice  (NbMots:Octet, Var NbFichiers:Octet,
                               TM:Tab, Var FChemins:Texte, Var M:Mat)
2) Ouvrir(FChemins)
3) NbFichiers ← 0
4) Tant Que Non Fin_Fichier(FChemins) Faire
       NbFichiers ← NbFichiers +1
       Pour i De 1 À NbMots Faire        {Initialisation à 0 de la ligne}
            M[NbFichiers, i] ← 0         {associée au fichier texte en cours}
       Fin Pour
       Lire_nl(FChemins, NomPhysiqueD1Fichier) {Lecture du Chemin d'1 Fichier}
       Associer(UnFichierText,NomPhysiqueD1Fichier) {..., son Association}
       Ouvrir(UnFichierTexte)      {..., son Ouverture}
       Tant Que Non Fin_Fichier(UnFichierText) Faire {et son parcours}
            Lire_nl(UnFichierText, Ligne) {Lecture d'une ligne}
            Pour i De 1 À NbMots Faire {Parcours + Rech des Mots & MAJ Matrice}
                 M[NbFichiers, i]←M[NbFichiers, i] + Ord(Pos(TM[i], Ligne)>0)
            Fin Pour
       Fin Tant Que
       Fermer(UnFichierTexte)
   Fin Tant Que
5) Fermer(FChemins)
6) Fin RemplirMatrice
```

TDO			
UnFichierText	Texte	i	Octet
NomPhysiqueD1Fichier	Chaîne[80]	ligne	Chaîne

Procédure Afficher

```
1) Procédure Afficher(NbMots, NbFichiers:Octet, TM:Tab,Var FChemins:Texte,M:Mat)
2) Pour j De 1 À NbMots Faire {Parcours des Mots}
       Écrire(TM[j], ": " ) {Affichage d'un Mot du Tableau TM}
       Ouvrir(FChemins) {Ouverture du fichier qui contient les chemins}
       Pour i De 1 À NbFichiers Faire {Parcours des Chemins}
            Lire_nl(FChemins, UnChemin) {Lecture d'un chemin}
            Si M[i, j] ≠ 0 Alors {Si la case du couple Chemin/Mot ≠ 0 ...}
                 Écrire(UnChemin, " ") {...On affiche ce Chemin suivi d'1 espace}
            Fin Si
       Fin Pour
       Écrire_nl {Retour à la Ligne}
   Fin Pour
3) Fin Afficher
```

TDO	
i, j	Octet
UnChemin	Chaîne[80]

Problème : Corrigé Version 02

Analyse Du Programme Principal

```
Nom: ProblèmeVersion02
Résultat=  [Associer(FChemins,"C:/Chemin.txt")
           SaisirLesMotsARecherche(N, TM)]  Afficher(N, TM, FChemins)
Fin ProblèmeVersion02
```

TDNT	
Tab	Tableau[1..10] de Chaîne

TDO	
N	Octet
TM	Tab
FChemins	Fichier Texte
SaisirLesMotsARechercher, Afficher	Procédure

Procédure SaisirLesMotsARechercher

```
1) Procédure SaisirLesMotsARechercher(Var N: Octet, Var TM:Tab)
2) Répéter
        Lire(N)
   Jusqu'à N Dans [1..10]
3) Pour i De 1 À N Faire
        Répéter
             Lire(TM[i])
        Jusqu'à TM[i]≠"" Et Alphabetique(TM[i])
   Fin Pour
4) Fin SaisirLesMotsARechercher
```

TDO			
Alphabetique	Fonction	i	Octet

Fonction Alphabetique

```
1) Fonction Alphabetique(Ch:Chaîne):Booléen
2) i ← 0
3) Répéter
        i ← i+1
   Jusqu'à Non Majus(Ch[i]) Dans ["A".."Z"] Ou i=Long(Ch)
4) Alphabetique ← Majus(Ch[i]) Dans ["A".."Z"]
5) Fin Alphabetique
```

TDO	
i	Octet

Procédure Afficher

```
1) Procédure  Afficher(N:Octet, TM:Tab, Var FChemins:Texte)
2) Pour j De 1 À N Faire
        Écrire(TM[j], ": " )
        Ouvrir(FChemins)
        i ← 0
        Tant Que Non Fin_Fichier(FChemins) Faire
             Lire_nl(FChemins, UnChemin)
             i ← i + 1
             M[i, j]← NbLignesContenantLeMot(TM[j], UnChemin)
             Si M[i, j] ≠ 0 Alors
                  Écrire(UnChemin, " ")
             Fin Si
        Fin Tant Que
        Écrire_nl
   Fin Pour
3) Fermer(FChemins)
4) Fin Afficher
```

TDNT	
Mat	Tableau[1..100, 1..10] De Octet
Ch80	Chaîne[80]

TDO			
UnChemin	Ch80	i, j	Octet
NbLignesContenantLeMot	Fonction	M	Mat

Fonction NbLignesContenantLeMot

```
1) Fonction NbLignesContenantLeMot(Mot, Chemin:Ch80):Entier
2) Assosier(F, Chemin)
3) Ouvrir(F)
4) Nb ← 0
5) Tant Que Non Fin_Fichier(F) Faire
        Lire_nl(F, Ligne)
        Nb ← Nb + Ord(Pos(Mot, Ligne)≠0)
   Fin Tant Que
6) Fermer(F)
7) NbLignesContenantLeMot ← Nb
8) Fin NbLignesContenantLeMot
```

TDO	
F	Texte
Nb	Entier
Ligne	Chaîne

Traduction En Pascal

```
(*
Vous pouvez Copier/Coller le Code Source dans TPW ou autre EDI, et faire l'exécution sans
la préparation préalable des fichiers ou autre, les programmes s'en chargeront :-)
*)
```

Exercice 02 Version Itérative

```
Uses Wincrt;
Var  X:  Real;  N:  Integer;

Function PartieDecimaleEnBinaire(X:Real; N:Integer):  String;
Var  i:  Integer; Res:  String;
Begin
   Res := '0.';
   For i:=1 To n Do
      Begin
         X := X*2;
         If X<1 Then Res := Res+'0'
         Else
            Begin
               Res := Res+'1';
               X := X-1;
            End;
      End;
   PartieDecimaleEnBinaire := Res;
End;

Begin
   Write('X: '); Readln(X);
   Write('N: '); Readln(N);
   Writeln(PartieDecimaleEnBinaire(X, N));
End.
```

Exercice 02 Version Récursive

```
Uses Wincrt;
Var X:  Real; N:  Integer;

Function Binaire(AjouterZeroVirgule:Boolean; X:Real; N:Integer):  string;
Begin
   If AjouterZeroVirgule Then Binaire := '0.'+Binaire(False, X, N)
   Else If N=1 Then Binaire := Chr(48+Ord(X*2>=1)){48 peut être remplacé par Ord("0")}
   Else Binaire := Chr(48+Ord(X*2>=1))+Binaire(False, Frac(X*2) , N-1);
End;

Begin
   Write('X: '); Readln(X);
   Write('N: '); Readln(N);
   Writeln(Binaire(True, X, N));
End.
```

Exercice 03

```
Uses Wincrt;
Var
   i:  Integer;
   FSource, FResultat:  Text;

Function Factorielle(X:Integer):Integer;
Begin
    If X<2 Then Factorielle:=1
    Else Factorielle:=X*Factorielle(X-1);
End;

Procedure GenererFichierResultat(Var FSource, FResultat:Text);
Var
   A, B:  Integer;
Begin
   Reset(FSource);
   ReWrite(FResultat);
   While Not Eof(FSource) Do
      Begin
         { Read(FSource, A, B); }
         Read(FSource, A);
         Read(FSource, B);
         If Factorielle(A) * Factorielle(B) Mod (A+B) In [A, B] Then
         Begin
           Writeln(FResultat, A,'+i*',B);
           Writeln(A,'+i*',B);
         End;
      End;
      Close(FSource);
      Close(FResultat);
End;

Begin
   Randomize;
   Assign(FSource,'Source.txt');
   Assign(FResultat,'Resultat.txt');
   Rewrite(FSource);
   {Début du Remplissage Aléatoire du Fichier FSource par des couples d'entiers}
   For i:=1 To 10 Do  Writeln(FSource, Random(7)+1, ' ', Random(7)+1);
   Write(FSource, Random(7)+1, ' ', Random(7)+1);
   {Fin du Remplissage}
   GenererFichierResultat(FSource, FResultat);
End.
```

Corrigé du Problème - Version 01

```
Uses Wincrt;
Type
   Tab =  Array[1..10] Of String;
   Mat =  Array[1..100, 1..10] Of Byte;
Var
   NbMots, NbFichiers:  Byte;
   Tm:  Tab;
   M:  Mat;
   FChemins:  Text;
(*************************************************************************)
Procedure SaisirLesMotsARechercher(Var NbMots:Byte; Var TM:Tab);
{----}
    Function Alphabetique(Ch:String):  Boolean;
    Var
       i:  Integer;
    Begin
       i := 0;
       Repeat
          Inc(i);
       Until Not (Upcase(Ch[i]) In ['A'..'Z']) Or (i=Length(Ch));
       Alphabetique := Upcase(Ch[i]) In ['A'..'Z'];
    End;
{----}
Var
   i:  Integer;
Begin
   Repeat
      Write('NbMots: '); Readln(NbMots);
   Until NbMots In[1..10];
   For i:=1 To NbMots Do
      Begin
         Repeat
            Write('TM[',i,']: '); Readln(TM[i]);
         Until (TM[i]<>'') And Alphabetique(TM[i]);
      End;
End;
(*************************************************************************)
```

```
Procedure Afficher(NbMots, NbFichiers:Byte; TM:Tab;Var FChemins:Text;M:Mat);
Var
   i, j:  Byte;
   UnChemin:  String[80];
Begin
   For j:=1 To NbMots Do
      Begin
         Write(TM[j],' :');
         Reset(FChemins);
         For i:=1 To NbFichiers Do
            Begin
               Readln(FChemins, UnChemin);
               If M[i, j] <> 0 Then Write(UnChemin:Length(UnChemin)+2);
            End;
         Writeln;
      End;
End;
(*********************************************************************)
Procedure RemplirMatrice (NbMots:Byte; Var NbFichiers:Byte;
                          TM:Tab; Var FChemins: Text; Var M:Mat);
Var
   UnFichierTexte:  Text;
   NomPhysiqueD1Fichier:  String[80];
   Ligne:  String;
   i:  Byte;
Begin
   Reset(FChemins);
   NbFichiers := 0;
   While Not Eof(FChemins) Do
      Begin
         NbFichiers := NbFichiers+1;
         For i:=1 To NbMots Do M[NbFichiers, i] := 0;
         Readln(FChemins, NomPhysiqueD1Fichier);
         Assign(UnFichierTexte, NomPhysiqueD1Fichier);
         Reset(UnFichierTexte);
         While Not Eof(UnFichierTexte) Do
            Begin
               Readln(UnFichierTexte, Ligne);
               For i:=1 To NbMots Do
                  Begin
                     M[NbFichiers, i] := M[NbFichiers, i]+Ord(Pos(TM[i],Ligne)>0);
                  End;
            End;
         Close(UnFichierTexte);
      End;
   Close(FChemins);
End;
(*********************************************************************)
```

```
Procedure SePreparerPourLexecution;
Var
   F, FChemins:  Text;
   i:  Byte;
Type
   NMots =  0..15;
Const
   cListeDossiers:  Array[1..3] Of String =  ('Test', 'Test\Algo', 'Test\Prog');
   cListeFichiers:  Array[1..5] Of String =  ('Test/A.Txt', 'Test/Algo/B.Txt',
                                              'Test\Prog\C.Txt', 'Test\D.Txt',
                                              'Test\Prog\E.Txt');
   cListeMots:  Array[NMots] Of String =  ('Php','Algo','Html','Info',
                                           'Du', 'Texte', 'Pour', 'Remplir',
                                           'Les', 'Fichiers', 'Mot1', 'Mot2',
                                           'Mot3', 'Mot4', 'Mot5', 'Mot6');
Var
   Kes:  String[100];
   j, k:  Byte;
Begin
   Assign(FChemins, 'Chemin.txt');
   Rewrite(FChemins);
   Randomize;
{$I-}
   For i:=1 To 3 Do
      Mkdir(cListeDossiers[i]);
{$I+}
   IOResult;
   For i:=1 To 5 Do
      Begin
         Assign(F,cListeFichiers[i]);
         Writeln(FChemins,cListeFichiers[i]);
         Rewrite(F);
         For j:=1 To 1+Random(3) Do
            Begin
               Kes := '';
               For k:=1 To 5+Random(5) Do
                  Begin
                     Kes := Kes+cListeMots[Random(16)]+' ';
                  End;
               Writeln(F, Kes);
            End;
         Close(F);
      End;
   Close(FChemins);
End;

(************************ Programme Principal ******************************)
Begin
   SePreparerPourLexecution;
   Assign(FChemins, 'Chemin.txt');
   SaisirLesMotsARechercher(NbMots, TM);
   RemplirMatrice(NbMots, NbFichiers, TM, FChemins, M);
   Afficher(NbMots, NbFichiers, TM, FChemins, M);
End.
```

Corrigé du Problème - Version 02

```
Uses Wincrt;
Type
   Tab =  Array[1..10] Of String;
Var
   N:  Byte;
   Tm:  Tab;
   FChemins:  Text;

(*********************************************************************)
Procedure SaisirLesMotsARechercher(Var NbMots:Byte; Var TM:Tab);
{----}
    Function Alphabetique(Ch:String):  Boolean;
    Var
       i:  Integer;
    Begin
       i := 0;
       Repeat
          Inc(i);
       Until Not (Upcase(Ch[i]) In ['A'..'Z']) Or (i=Length(Ch));
       Alphabetique := Upcase(Ch[i]) In ['A'..'Z'];
    End;
{----}
Var
   i:  Integer;
Begin
   Repeat
      Write('NbMots: ');
      Readln(NbMots);
   Until NbMots In[1..10];
   For i:=1 To NbMots Do
      Begin
         Repeat
            Write('TM[',i,']: ');
            Readln(TM[i]);
         Until (TM[i]<>'') And Alphabetique(TM[i]);
      End;
End;
(*********************************************************************)
```

```
Procedure Afficher(N:Byte; TM:Tab;Var FChemins:Text);
Type
   Mat =  Array[1..100, 1..10] Of Byte;
   Ch80 =  String[80];
Var
   i, j:  Byte;
   UnChemin:  Ch80;
   M:  Mat;
{----}
    Function NbLignesContenantLeMot(Mot, Chemin:Ch80):  Integer;
    Var
       F:  Text;
       Nb:  Integer;
       Ligne:  String;
    Begin
       Assign(F, Chemin);
       Reset(F);
       Nb := 0;
       While Not Eof(F) Do
          Begin
             Readln(F, Ligne);
             Nb := Nb+Ord(Pos(Mot, Ligne)<>0)
          End;
       Close(F);
       NbLignesContenantLeMot := Nb;
    End;
{----}
Begin
   For j:=1 To N Do
      Begin
         Write(TM[j],' :');
         Reset(FChemins);
         i := 0;
         While Not Eof(FChemins) Do
            Begin
               Readln(FChemins, UnChemin);
               Inc(i);
               M[i, j] := NbLignesContenantLeMot(TM[j], UnChemin);
               If M[i, j] <> 0 Then Write(UnChemin, ' ');
            End;
         Writeln;
      End;
   Close(FChemins);
End;
(*************************************************************************)
```

```
Procedure SePreparerPourLexecution;
Var
   F, FChemins:  Text;
   i:  Byte;
Type
   NMots =  0..15;

Const
   cListeDossiers:  Array[1..3] Of String =  ('Test', 'Test\Algo', 'Test\Prog');
   cListeFichiers:  Array[1..5] Of String =  ('Test/A.Txt'     , 'Test/Algo/B.Txt',
                                              'Test\Prog\C.Txt', 'Test\D.Txt'     ,
                                              'Test\Prog\E.Txt'                   );
   cListeMots:  Array[NMots] Of String =  ('Php' , 'Algo'    , 'Html', 'Info'   ,
                                          'Du'  , 'Texte'   , 'Pour', 'Remplir',
                                          'Les' , 'Fichiers', 'Mot1', 'Mot2'   ,
                                          'Mot3', 'Mot4'    , 'Mot5', 'Mot6'   );
Var
   Kes:  String[100];
   j, k:  Byte;
Begin
   Assign(FChemins, 'Chemin.txt');
   Rewrite(FChemins);
   Randomize;
{$I-}
   For i:=1 To 3 Do Mkdir(cListeDossiers[i]);
{$I+}
   Ioresult;
   For i:=1 To 5 Do
      Begin
         Assign(F,cListeFichiers[i]);
         Writeln(FChemins,cListeFichiers[i]);
         Rewrite(F);
         For j:=1 To 1+Random(3) Do
            Begin
               Kes := '';
               For k:=1 To 5+Random(5) Do
                  Begin
                     Kes := Kes+cListeMots[Random(16)]+' ';
                  End;
               Writeln(F, Kes);
            End;
         Close(F);
      End;
   Close(FChemins);
End;

(*********** Programme Principal ****************************************)
Begin
   SePreparerPourLexecution;
   Assign(FChemins, 'Chemin.txt');
   SaisirLesMotsARechercher(N, TM);
   Afficher(N, TM, FChemins);
End.
```