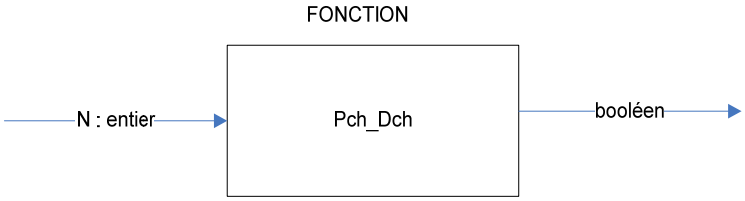
Corrigé EMD2 du 31 janvier 2012

PARTIE A

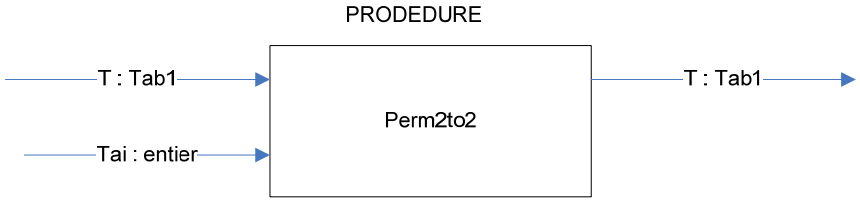
Découpage modulaire :

Les modules à construire sont pratiquement implicites :

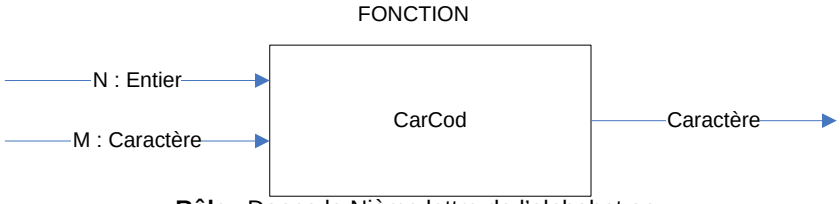
- Un module (Prem) nous permettra de savoir si un nombre est premier ou pas (existant déjà)
- Un module (Pch_Dch) pour détecter si un nombre commence et se termine par le même chiffre
- (ce module nécessitera le module qui donne le nombre de positions d'un nombre (Nb_Pos et celui qui extrait une position d'un nombre Extr_Nb _ ces 2 modules existent déjà)
- Un module (Perm2to2) qui permute 2 à 2 les éléments d'un tableau (qui va utiliser le module Permut existant)
- Un module (CarCod) qui nous permet d'avoir la Nième lettre de l'alphabet
- Et finalement un module (Decrypte) pour obtenir le message décodé



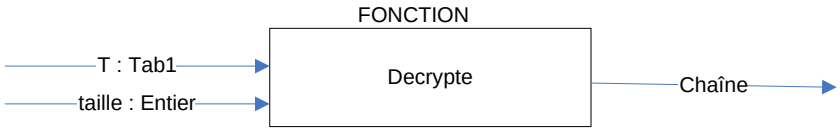
Rôle : donne VRAI si le premier et le dernier chiffre de N sont identiques



Rôle : permute 2 à 2 les éléments du tableau T (le 1er avec le 2ème, le 3ème avec le 4ème, …)



Rôle : Donne la Nième lettre de l'alphabet en majuscule si M ='M' ou en minuscule si M = 'm'



Rôle : Donne le message (contenu dans T) décodé selon la clé (suppression des nbres premiers commençant et se terminant par le même chiffre, permutation 2 à 2 des éléments du tableau, une case du tableau est une lettre dont le rang dans alphabet est = T[i] mod 52)

Les modules déjà existant sont :

Fonction Nb_Pos (N : entier) : *entier* *(* donne le nombre de positions de N *)*

Fonction Extr_Nb (N , C , P : entier) *(* extrait de N , un nombre de c chiffres à partit de la position P incluse*)*

Fonction Prem (N : entier) : Booléen *(* donne vrai le nombre N est premier *)*

Procédure Permut (Var M,N :entier) *(* Permute les éléments M et N*)*

Fonction Chr (x :entier) : caractère *(*donne le caractère s'il est dans la liste des caractères ASCII dont la valeur est X *)*

Construction du module Pch_Dch:

Analyse :

- Si le premier chiffre de N est égal au dernier chiffre de N alors on met Pch_Dch = Vrai et Faux sinon

Algorithme :

```
Fonction  Pch_Dch (N : entier ) : Booléen
Fonctions extr_nb , nb_pos.fon

DEBUT
SI Extr_nb(n,1,1) = Extr_Nb(n,1,nb_pos(n)) Alors Pch_Dch← VRAI
                                              Sinon Pch_Dch ← FAUX
FIN
```

Construction du module Perm2to2 :

Analyse :

- I = 1 *(c'est l'indice de t)*
- Tant que i < Tai *(Tai est la taille du tableau)*
  - On permute les élément T[i] et T[i+1]
  - Et on met + 2 dans i *(pour avoir i =1 puis 3 puis 5…)*

Algorithme :

```
procédure Perm2to2(var t : tab1;tai:Entier)
variables  i : entier
procédure permut

DEBUT
i← 1;
Tant Que i <tai Faire
          DTQ
          permut (t[i],t[i+1])
          i ← i + 2
          FTQ
FIN
```

Construction du module Carcod :

Analyse :

- Si N > 26 OU N < 1 on met '*' dans Carcod *(il s'agit d'une erreur, on est hors de l'alphabet)*
- Si M = 'M', carcod = Chr(N + 64) *(Les lettres en majuscules commencent dès le 65ième élément de la liste des caractères ASCII)*
  - Si M='m' , carcod = Chr(N + 96) *(Les lettres en minuscules commencent dès le 97ième élément de la liste des caractères ASCII)*

Algorithme :

```
fonction Carcod( n :  Entier ;M : caractère): caractère

DEBUT
Si (n > 26) OU (n < 1) Alors Carcod ← '*'
Sinon Si M='M' Alors Carcod ← chr(n + 64)
       Sinon Carcod ← chr(n +96 )
FIN
```

Construction du module Decrypte :

Analyse :

- On prend un tableau Tp de même taille que (Tp est un tableau intermédiaire pour faciliter la solution)

<u>On enlève tous les nbres premiers qui commencent et se terminent par le même chiffre</u>

- On parcourt le tableau T, i =1,2,3,…,taille et à chaque fois
  - Si un élément n'est ni premier ou ni il commence et se termine par le même chiffre
    - On le met dans Tp

Puis On permute 2 à 2 , les éléments de Tp en faisant

- Perm2to2 (Tp, taille de Tp)

On construit notre message

- On prend une chaîne vide Mess *(qui va contenir le message décodé)*
- On parcourt le tableau Tp, i = 1,2,3,….,tailla de Tp
  - Si l'élément Tp[i] = 99 alors on met un blanc dans Mess
    Sinon on rajoute dans Mess la lettre ayant pour rang dans l'alphabet : T[i] Mod 52
    Soit mess = mess + carcod (Tp[i] mod 52,'M') *(le 'M' pour avoir le message en majuscules)*

Algorithme :

```
function Decrypte (t:tab1; taille: entier): Chaîne
variables      Tp: tab1
                i,j : entier
              mess: chaîne
Fonction prem.

DEBUT
j ← 0
Pour  i allant de 1 à  taille Faire
      Si  (prem(t[i]) = Faux ) OU (pch_dch(t[i]) = Faux)  Alors
                              Dsi
                              J ← j +1
                              Tp[j] ← t[i]
                              Fsi
Perm2to2(Tp,j)
Mess ← ''
Pour i  allant de 1 à j Faire
      Si  Tp[i] = 99 Alors mess ← mess + ' '
      Sinon mess ← mess + carcod(Tp[i] mod 52,'M')
Decrypte ← mess
FIN
```

Construction de l'algorithme principal :

Analyse :

- On lit l tableau contenant le message codé
- On affiche le message décrypté

Algorithme :

```
Algorithme ed21112
type Tab1 = Tableau [1..1000] d'entier
variables  taille, x : Entier
           MC : tab1
           mescode :Chaîne
procédure lect1d
fonction decrypte

DEBUT
Ecrire ('Donner le tableau contenant le message : ')
lect1d(Mc,taille)
Ecrire ( 'MESSAGE DECODE : ',decrypte(Mc,taille))
FIN
```

PARTIE B

Structure de l'enregistrement

```
TYPE   tab1 = tableau [1..1000] d'entier
       M =    Enregistrement
              Code : tab1
              Decode : chaîne
              FIN
```

L'algorithme principal devient

```
Algorithme ed21112
type     Tab1 = Tableau[1..1000] d'entier
         M =    Enregistrement
                Code : tab1
                Decode : chaîne
                FIN
Variables        taille, x : Entier
                 MC : tab1
                 Message : M
procédure lect1d
fonction decrypte
   DEBUT
   Ecrire ('Donner le tableau contenant le message : ')
   lect1d(Mc, taille)
   message.code  = Mc
   message.decode =  decrypte (Mc, taille)
   Ecrire ( 'MESSAGE DECODE : ', message.decode)
   FIN
```

PROGRAMMATION

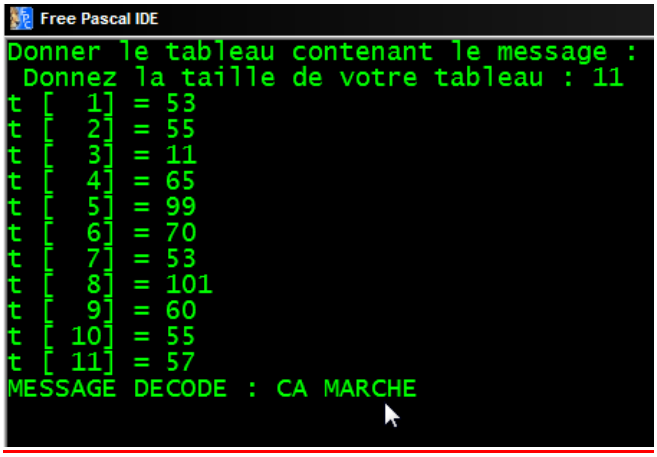
```
program ed21112;
uses crt;
type Tab1 = array [1..1000] of longint;
Var     taille,x : integer;
        MC : tab1;
        mescode :string;
{$i e:\algo\modules\lect1d.pro}
{$i e:\algo\modules\décrypte.Fon}
BEGIN
clrscr;
Writeln ('Donner le tableau contenant le message : ');
lect1d(Mc,taille);
write( 'MESSAGE DECODE : ',decrypte(Mc,taille));
readln;
END.
================================================================================
Function Pch_Dch(n:longint) : boolean;
// ---------------------------------------------------------------------
// Donne Vrai si le premier et le dernier chiffre de N sont identiques
// ---------------------------------------------------------------------

{$i e:\algo\modules\extr_nb.fon}
{$i e:\algo\modules\nb_pos.fon}
BEGIN
if Extr_nb(n,1,1) = Extr_Nb(n,1,nb_pos(n)) then Pch_Dch := true
                                           else Pch_Dch := false;
END;
```

```
=============================================================================
procedure perm2to2(var t : tab1;tai:integer);
// ---------------------------------------------------------------------------
// Permute 2 a 2 les elements de T (elem1 avec elem2, elem3 avec elem4, ...)
// ---------------------------------------------------------------------------
var i : integer;
{$i e:\algo\modules\permut.pro}
BEGIN
i:=1;
while i <tai do
                BEGIN
                permut (t[i],t[i+1]);
                i := i + 2;
                END;
END;
```

```
function Carcod(n:integer;M:char): char;
// ---------------------------------------------------------
// Donne la Nieme lettre de l'alphabet ou * si n est incorrect
// ---------------------------------------------------------
BEGIN
if (n > 26) or (n < 1) then Carcod := '*'
else if M='M' then Carcod:= chr(n + 64)
     else Carcod:= chr(n +96 )
END;
=============================================================================
function Decrypte(t:tab1;taille:integer): string;
// ---------------------------------------------------------
// Decrypte le message contenu dans le tableau T selon la cle :
// enlever les nbres premiers commencant et se terminant par le meme chiffre
// permuter 2 a 2 les elements du tableau . Chaque case est une lettre
// dont le rang dans l'alphabet est donne par T[i] Mod 52
// ---------------------------------------------------------

var     Tp:tab1;
        i,j :integer;
        mess:string;
{$i e:\algo\modules\prem.fon}
BEGIN
j := 0;
for i :=1 to taille do
        if (prem(t[i])= false) or (pch_dch(t[i])=false)  then
                                        BEGIN
                                        j:=j +1;
                                        Tp[j]:=t[i];
                                        END;
Perm2to2(Tp,j);
mess:='';
for i :=1 to j do
        if Tp[i] = 99 then mess:=mess + ' '
                      else mess:=mess + carcod(Tp[i] mod 52,'M');
decrypte:=mess;
END;
```



BAREME DE NOTATION

PARTIE A	analyse	algorithme	total
Recherche des nombres premiers commençant et se terminant par le même chiffre	0.5	1	1.5
Permutation 2 à 2 des éléments du tableau	1	2	3
Recherche des lettres de l'alphabet	1	2	3
Décryptage du message	1	2	3
Algorithme principal	0.5	1	1.5
			12
Programmation			6
		TOTAL	18

PARTIE B			total
Déclaration de l'enregistrement			1
Algorithme principal			2

ENLEVER SYSTEMATIQUEMENT 2 POINTS A TOUT TRAVAIL NON SOIGNE