



## Opérations monétaires

---

Par  
Stéphanie ROBINEAU, Support Technique 4D  
Note technique 4D-200010-27-FR  
Version 1  
Date 1 Octobre 2000

### Résumé

---

Cette note technique regroupe plusieurs méthodes utilitaires : Conversion de nombres en lettres, de chiffres arabes en chiffres romains et de vérifications de No de comptes (RIB) et de cartes bancaires.

Ces méthodes pourront être facilement implémentées dans des applications de type "eCommerce" par exemple.

## 4D Notes techniques

---

Copyright © 1985-2004 4D SA - Tous droits réservés

Tous les efforts ont été faits pour que le contenu de cette note technique présente le maximum de fiabilité possible. Néanmoins, les différents éléments composant cette note technique, et le cas échéant, le code, sont fournis sans garantie d'aucune sorte. L'auteur et 4D S.A. déclinent donc toute responsabilité quant à l'utilisation qui pourrait être faite de ces éléments, tant à l'égard de leurs utilisateurs que des tiers.

Les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SA. La fourniture du logiciel décrit dans ce document est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation afférente. Le logiciel et sa documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce document ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SA.

4D, 4D Calc, 4D Draw, 4D Write, 4D Insider, 4ème Dimension®, 4D Server, 4D Compiler ainsi que les logos 4e Dimension, sont des marques enregistrées de 4D SA.

Windows, Windows NT, Win 32s et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Power Macintosh, LaserWriter, ImageWriter, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2002 est un produit de Altura Software, Inc.

4D Write contient des éléments de "MacLink Plus file translation", un produit de DataViz, Inc, 55 Corporate drive, Trumbull, CT, USA.

XTND Copyright 1992-2002 © 4D SA. Tous droits réservés.

XTND Technology Copyright 1989-2002 © Claris Corporation.. Tous droits réservés ACROBAT © Copyright 1987-2002, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

---

## Introduction

---

Vous nous le demandez souvent au support technique : routines de conversion des numériques vers une chaîne texte (ex : impression de chèques). Nous reprendrons dans cette note une ancienne méthode traitant partiellement du sujet. Nous espérons l'avoir complétée et nous y adjoindrons quelques méthodes "monétaires" : le test des numéros de compte et de cartes bancaires, et une méthode de conversion des chiffres arabes en chiffres Romains.

## Les chiffres romains.

---

### 1 - Principe

Pour écrire les chiffres romains, nous disposons de 7 signes : I ; V ; X ; L ; C ; D ; M.

CHIFFRES ROMAINS	CHIFFRES ARABES
I	1
II	2
III	3
IV	4
V	5
VI	6
VII	7
VIII	8
IX	9
X	10
XI	11
XII	12
XX	20
XXX	30
XL	40
L	50
LX	60
LXX	70
LXXX	80
XC	90
C	100
D	500
M	1 000

### Parfois utilisés :

V	5 000
X	10 000

L	50 000
C	100 000
D	500 000
M	1 000 000

Pour écrire les chiffres arabes en chiffres romains nous opérons :

- par addition quand la valeur d'une lettre est supérieure ou égale à la suivante
- par soustraction quand la valeur d'une lettre est inférieure à la suivante.

### Exemple :

$$\text{VIII} = 5 + 3 = 8$$

$$\text{XX} = 10 + 10 = 20$$

$$\text{LXVII} = 50 + 10 + 5 + 2 = 67$$

$$\text{IV} = 5 - 1 = 4$$

$$\text{XL} = 50 - 10 = 40$$

Ce système n'est pas employé pour les milliers (M)

Remarque : Le chiffre I ne peut se retrancher que de V et X, le chiffre V ne se retranche pas.

Pour transcrire un nombre arabe en chiffres romains, nous décomposons ce nombre comme dans l'exemple suivant :

$$1988 = 1\ 000 + 900 + 80 + 8 = \text{M} + \text{CM} + \text{LXXX} + \text{VIII} = \text{MCMLXXXVIII}$$

Pour écrire un nombre de deux chiffres, nous écrivons d'abord les dizaines puis les unités:

$$99 \text{ soit } 90 + 9 = \text{XCIX}$$

Pour écrire un nombre de trois chiffres, nous écrivons d'abord les centaines, puis les dizaines et enfin les unités :

$$253 \text{ ou } 200 + 50 + 3 = \text{CCLIII}$$

Note : nous n'employons jamais quatre lettres semblables de suite, ainsi 9 s'écrit IX et non VIII.

Autrefois, les Romains utilisaient d'autres signes conventionnels pour exprimer les grands nombres, les voici :

- Un nombre surmonté d'un trait horizontal = mille :

$$\overline{\text{X}} = 10 \text{ mille}$$

$$\overline{\text{VII}} = 7 \text{ mille}$$

- Un nombre surmonté de deux traits horizontaux = million

==

VI = 6 millions

==

XV = 15 millions

- Un nombre surmonté de trois traits horizontaux = milliards

## 2 - Utilisations

Nous écrivons généralement en chiffres romains

- Les fascicules (les fascicules XX à XXIV...);
- Les parties d'un volume (livres) (le livre XII...);
- Les parties et les sections d'un ouvrage;
- Les pièces justificatives et les documents;
- Les planches (illustration);
- Les appendices joints à un ouvrage;
- Les évangiles;
- Les tomes ou les volumes;
- Les actes et scènes des pièces de théâtre;
- Les années dans le calendrier républicain;
- Les arrondissements d'une ville;
- Les conciles;
- Les dynasties et les régimes (la XXe dynastie. La IIIe république);
- Les millénaires;
- Les numéraux après un nom propre (Louis XIV);
- Les siècles;
- Les couplets, chants, et chapitres;
- Les sonnets, les stances et les strophes.

## 3 - Méthode

```
` fonction F_Num2Roman  
  
C_TEXTE($0)  
C_ENTIER LONG($1)  
  
$0:= ""  
  
C_ENTIER LONG($digit;$a1;$r1;$a2;$r2;$a3;$r3)  
C_TEXTE($vT_digitName1;$vT_digitName2;$vT_digitName3;$vT_digitName4)  
  
$vT_digitName1:= ""  
$vT_digitName1:= ""  
$vT_digitName1:= ""  
$vT_digitName1:= ""
```

\$digit:=\$1

**Si** ((\$digit>0) & (\$digit<=3999))

\$a1:=**Ent**(\$digit/1000)  
\$r1:=\$digit%1000

**Au cas ou**

: (\$a1=1)  
\$vT\_digitName1:="M"  
: (\$a1=2)  
\$vT\_digitName1:="MM"  
: (\$a1=3)  
\$vT\_digitName1:="MMM"

**Fin de cas**

\$a2:=**Ent**(\$r1/100)  
\$r2:=\$r1%100

**Au cas ou**

: (\$a2=9)  
\$vT\_digitName2:="CM"  
: (\$a2=8)  
\$vT\_digitName2:="DCCC"  
: (\$a2=7)  
\$vT\_digitName2:="DCC"  
: (\$a2=6)  
\$vT\_digitName2:="DC"  
: (\$a2=5)  
\$vT\_digitName2:="D"  
: (\$a2=4)  
\$vT\_digitName2:="CD"  
: (\$a2=3)  
\$vT\_digitName2:="CCC"  
: (\$a2=2)  
\$vT\_digitName2:="CC"  
: (\$a2=1)  
\$vT\_digitName2:="C"

**Fin de cas**

\$a3:=**Ent**(\$r2/10)  
\$r3:=\$r2%10

**Au cas ou**

: (\$a3=9)  
\$vT\_digitName3:="XC"  
: (\$a3=8)  
\$vT\_digitName3:="LXXX"  
: (\$a3=7)  
\$vT\_digitName3:="LXX"  
: (\$a3=6)  
\$vT\_digitName3:="LX"  
: (\$a3=5)  
\$vT\_digitName3:="L"  
: (\$a3=4)  
\$vT\_digitName3:="XL"  
: (\$a3=3)  
\$vT\_digitName3:="XXX"

```
: ($a3=2)
$vT_digitName3:="XX"
: ($a3=1)
$vT_digitName3:="X"
```

**Fin de cas**

**Au cas ou**

```
: ($r3=9)
$vT_digitName4:="IX"
: ($r3=8)
$vT_digitName4:="VIII"
: ($r3=7)
$vT_digitName4:="VII"
: ($r3=6)
$vT_digitName4:="VI"
: ($r3=5)
$vT_digitName4:="V"
: ($r3=4)
$vT_digitName4:="IV"
: ($r3=3)
$vT_digitName4:="III"
: ($r3=2)
$vT_digitName4:="II"
: ($r3=1)
$vT_digitName4:="I"
```

**Fin de cas**

**Sinon**

**ALERTE**(Chaine(\$digit)+" n'est pas compris entre 1 et 3999")

**Fin de si**

\$0:=\$vT\_digitName1+\$vT\_digitName2+\$vT\_digitName3+\$vT\_digitName4

## Chiffres arabes vers chaine texte

---

### 1 -Règle :

Lorsque nous écrivons en toutes lettres les chiffres arabes, ils prennent une forme invariable, ils ne prennent donc JAMAIS la forme du pluriel.

Exemple : 99 s'écrit quatre vingt dix neuf

18 s'écrit dix huit

Etc.

MAIS : Il existe une exception à cette règle.

" Cent " prendra la forme du pluriel s'il est précédé d'un ou plusieurs chiffre(s), et qu'il n'est suivi d'aucun.

Par contre il suivra la première règle dès lors qu'il sera suivi d'un chiffre ou qu'il sera seul.

Exemple : 100 s'écrit cent

218 s'écrit deux cent dix huit

129 s'écrit cent vingt neuf

400 s'écrit quatre cents

## 2 -Méthode :

```
`fonction F_Num2String

C_TEXTE($0)
C_REEL($1)
C_BOOLEEN($2) `Francs (Faux) ou Euros(vrai)

$0:= ""
C_ENTIER
LONG($a;$vML;$vCM;$vCMint;$vDMint;$vM;$vMint;$vC;$vCint;$vD;$vDint;$vunity;$vUniint
;$vDcent;$vUcent)
C_ALPHA(80;$dectext;;$MLText;$CMText;$DMText;$MText;$CText;$DText)

$1:=Arrondi(Abs($1);2)
$Dec:=Arrondi(Dec($1)*100;0)
$a:=Ent($1)
Si ($a<=9999999)
  `millions
  $vML:=Ent($a/1000000)
  `centaines de milliers
  $vCM:=$a-($vML*1000000)
  $vCMint:=Ent($vCM/100000)
  `dizaines de milliers
  $vDM:=$a-($vML*1000000)-($vCMint*100000)
  $vDMint:=Ent($vDM/10000)
  `milliers
  $vM:=$a-($vML*1000000)-($vCMint*100000)-($vDMint*10000)
  $vMint:=Ent($vM/1000)
  `centaines
  $vC:=$a-($vML*1000000)-($vCMint*100000)-($vDMint*10000)-($vMint*1000)
  $vCint:=Ent($vC/100)
  `dizaines
  $vD:=$a-($vML*1000000)-($vCMint*100000)-($vDMint*10000)-($vMint*1000)-($vCint*100)
  $vDint:=Ent($vD/10)
  `unités
  $vUnity:=$a-($vML*1000000)-($vCMint*100000)-($vDMint*10000)-($vMint*1000)-($vCint*100)-
($vDint*10)
  $vUniint:=Ent($vUnity/1)
  `dizaines partie décimale
  $vDcent:=Ent($Dec/10)
  `unités partie décimale
  $vUcent:=$Dec-( $vDcent*10)

  `Million
  $MLText:=""
  Au cas ou
  : ($vML>1)
  $MLText:=&tabCH_U{$vML}+" Millions "
  : ($vML=1)
  $MLText:="Un Million "
  Fin de cas
  Si (($vML>0) & ($vCM=0))
  $MLText:=$MLText+"de "
Fin de si
```



```

`Cent Mille
$CMText:=""
Au cas ou
: ($vCMInt=1)
$CMText:="Cent "
: ($vCMInt>1)
$CMText:=&tabCH_U{$vCMInt}+" Cent "
Fin de cas

`Dix Mille
$DMText:=""
Au cas ou
: (($vDMInt=0) & ($vMInt=0))
$DMText:=""
: ($vDMInt=1)
Si ($vMInt=0)
$DMText:="Dix"
Sinon
$DMText:=&tabCH_Z{$vMInt}
Fin de si
: (($vDMInt=7) | ($vDMInt=9) | ($vDMInt=8))
Au cas ou
: (($vMInt>0) & ($vDMInt=8))
$DMText:=&tabCH_D{$vDMInt}+" "+&tabCH_U{$vMInt}
: (($vMInt>1) | (($vDMInt=9) & ($vMInt>0)))
$DMText:=&tabCH_D{$vDMInt-1}+" "+&tabCH_Z{$vMInt}
: (($vMInt=1) & ($vDMInt=7))
$DMText:=&tabCH_D{$vDMInt-1}+" et "+&tabCH_Z{$vMInt}
Sinon
$DMText:=&tabCH_D{$vDMInt}
Fin de cas
: ($vDMInt>1)
Au cas ou
: ($vMInt>1)
$DMText:=&tabCH_D{$vDMInt}+" "+&tabCH_U{$vMInt}
: ($vMInt>0)
$DMText:=&tabCH_D{$vDMInt}+" et "+&tabCH_U{$vMInt}
Sinon
$DMText:=&tabCH_D{$vDMInt}
Fin de cas
: (($vDMInt=0) & ($vMInt>1))
$DMText:=&tabCH_U{$vMInt}
Fin de cas
Si ($DMText#"")
$DMText:=$DMText+" "
Fin de si

`Mille
$MText:=""
Au cas ou
: (($vMInt=1) & (($CMText="" | ($DMText="")))
$MText:="Mille "
: (($CMText#"") | ($DMText#""))
$MText:="Mille "
: ($vMInt>1)
$MText:="Mille "
Fin de cas

```

```

`Centaines
Au cas ou
: ($vClnt=1)
$CText:="Cent "
: ($vClnt>1)
$CText:=&tabCH_U{$vClnt}
Si (($vDInt>0) | ($vUnilnt>0))
  $CText:=$CText+" Cent "
Sinon
  $CText:=$CText+" Cents "
Fin de si
Sinon
  $CText:=""
Fin de cas

`Dizaines & Unités
$DText:=""
Au cas ou
: (($vDInt=0) & ($vUnilnt=0))
  $DText:"Francs"
: ($vDInt=1)
Si ($vUnilnt=0)
  $DText:"Dix Francs"
Sinon
  $DText:=&tabCH_Z{$vUnilnt}+" Francs"
Fin de si
: (($vDInt=7) | ($vDInt=9) | ($vDInt=8))
Au cas ou
: (($vUnilnt>0) & ($vDInt=8))
  $DText:=&tabCH_D{$vDInt}+" "+&tabCH_U{$vUnilnt}+" Francs"
: (($vUnilnt>1) | (($vDInt=9) & ($vUnilnt>0)))
  $DText:=&tabCH_D{$vDInt-1}+" "+&tabCH_Z{$vUnilnt}+" Francs"
: (($vUnilnt=1) & ($vDInt=7))
  $DText:=&tabCH_D{$vDInt-1}+" et "+&tabCH_Z{$vUnilnt}+" Francs"
Sinon
  $DText:=&tabCH_D{$vDInt}+" Francs"
Fin de cas
Sinon
Au cas ou
: ($vUnilnt>1)
  $DText:=&tabCH_D{$vDInt}+" "+&tabCH_U{$vUnilnt}+" Francs"
: (($vUnilnt=1) & ($a>1))
  $DText:=&tabCH_D{$vDInt}
Si ($vDInt=0)
  $DText:=$DText+"Un Francs"
Sinon
  $DText:=$DText+" et Un Francs"
Fin de si
: (($vUnilnt=1) & ($a=1))
  $DText:"Un Franc"
Sinon
  $DText:=&tabCH_D{$vDInt}+" Francs"
Fin de cas
Fin de cas

`Décimales
$decText:=""

```

```

$reste:=Modulo(0;10)
Si($Dec>0)
  Au cas ou
    : ($vDcent=1)
    Si($vUcent=0)
      $decText:="Dix centimes"
    Sinon
      $decText:=&tabCH_Z{$vUcent}+" centimes"
    Fin de si
    : (($vDcent=7) | ($vDcent=9) | ($vDcent=8))
    Au cas ou
      : (($vUcent>0) & ($vDcent=8))
      $decText:=&tabCH_D{$vDcent}+" "+&tabCH_U{$vUcent}+" centimes"
      : (($vUcent>1) | (($vDcent=9) & ($vUcent>0)))
      $decText:=&tabCH_D{$vDcent-1}+" "+&tabCH_Z{$vUcent}+" centimes"
      : (($vUcent=1) & ($vDcent=7))
      $decText:=&tabCH_D{$vDcent-1}+" et "+&tabCH_Z{$vUcent}+" centimes"
    Sinon
      $decText:=&tabCH_D{$vDcent}+" centimes"
    Fin de cas
  Sinon
    Au cas ou
      : ($vUcent>1)
      $decText:=&tabCH_D{$vDcent}+" "+&tabCH_U{$vUcent}+" centimes"
      : ($vUcent=1)
      Si($vDcent=0)
        $decText:=$decText+"Un centime"
      Sinon
        $decText:=&tabCH_D{$vDcent}+" et Un centimes"
      Fin de si
    Sinon
      $decText:=&tabCH_D{$vDcent}+" centimes"
    Fin de cas
  Fin de cas
  $decText:="et "+$decText
Fin de si
$0:=$MLText+$CMTText+$DMText+$MText+$CText+$DText+" "+$decText
Si($2)
  $0:=Remplacer chaine($0;"centime";"cent")
  $0:=Remplacer chaine($0;"Franc";"Euro")
Fin de si
Sinon
  $0:="Limite de la fonction : 9 999 999,99 "
Fin de si

```

## CARTES DE CREDIT

---

### 1 - Comment vérifier la validité des numéros d'une carte de crédit ?

La vérification se fait par une méthode très simple qui est décrite ci-dessous :

- Soit un numéro de carte de crédit. En partant de la droite et en allant vers la gauche, doublez un chiffre sur deux en commençant par le deuxième.

Exemple :

Le numéro de la carte est : 4992739871618 (Visa : 13 chiffres)

Ici vous commencerez par doubler le 1 puis le 1 puis le 8, et ainsi de suite.

Ce qui nous donne :

4	9	9	2	7	3	9	8	7	1	6	1	8	
				*	2	*	2	*	2	*	2	*	2
-----													
1	8	4		6		1	6		2		2		

En additionnant tous les chiffres obtenus nous arrivons à un résultat de  $(1+8)+4+6+(1+6)+2+2=30$

Note : les valeurs supérieures à 9 (deux chiffres) devront être additionnées chiffre par chiffre :

$$18 = 1+8 = 9$$

- Ensuite, vous devez ajouter à ce résultat tous les chiffres qui n'ont pas été doublés. Reprenons notre exemple :

4 9 9 2 7 3 9 8 7 1 6 1 8

donc, ici, les chiffres à ajouter sont :  $6+7+9+7+9+4+8$ , ce qui nous donne un total de 50.

- Pour terminer, pour que le numéro de votre carte de crédit soit valide, vous devez obtenir un nombre dont le modulo par 10 soit nul (50, 60, 70, 80, etc.) en faisant la somme des deux résultats calculés précédemment

Ici,  $30+50=80$

Le numéro de la carte est valide parce que  $70/10$  n'a pas de reste.

## 2 - Préfixes et longueur des numéros selon le type de carte

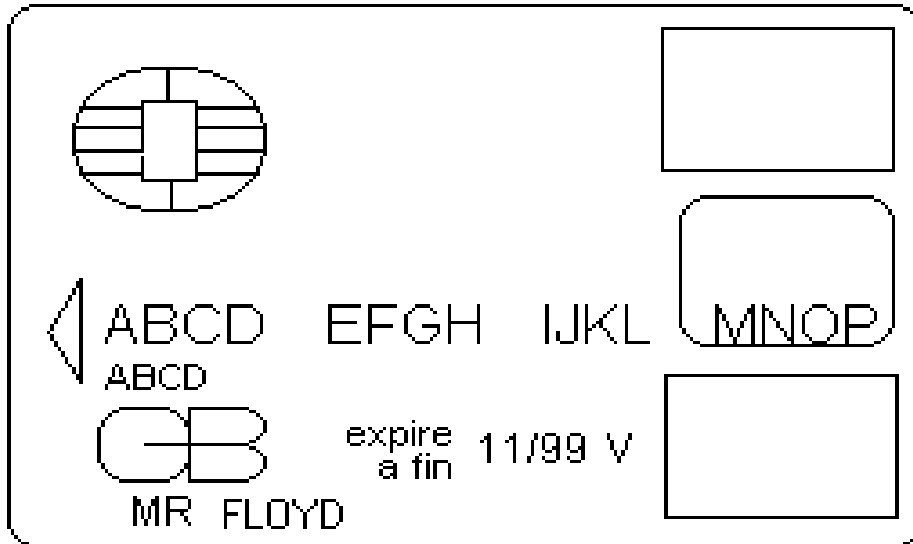
Types de carte	Préfixes	Longueur
MasterCard	51-55	16
Visa	4	13, 16
AmericanExpress	34, 37	15
Diner's Club	300-305, 36, 38	14

Comme vous l'avez déjà constaté, une carte de crédit comporte un numéro, dont le nombre de chiffres varie selon le type.

A quoi correspondent ces chiffres ?

- Le premier correspond au type de carte.
- Les 3, 4, ou 5 numéros suivants (selon les organismes qui émettent les cartes de crédit) représentent le numéro attribué aux banques et organismes.
- Les numéros suivants sont choisis au hasard sauf le dernier. Celui-ci permet de vérifier la validité de la carte de crédit. Cette clé est parfois appelée la clé de Luhn.

Résumons avec le schéma ci-dessous :



- A représente le type de carte.
- BCD+E+F représente le numéro attribué aux banques.
- P est la clé qui est aussi appelée clé de Luhn
- Le V derrière la date est une donnée pour le constructeur.

Préfixes de cartes Eurocard :

- Eurocard France : 5294, 5295, 513
- Eurocard International : 541333, 544333, 545333, 546333
- Eurocard Belgique- Luxembourg : 540054, 541327, 544327, 544327
- Eurocard AB : 5275, 53004, 541256
- Eurocard Suisse : 541325
- Eurocard Danemark : 541303
- Eurocard Pays-Bas : 541330

Préfixes de cartes American express :

- 372034, 372407, 372861, 373227

Préfixes de cartes VISA françaises :

- BNP : 4974
- Caisse d'épargne : 4978 4927
- La Poste : 4970
- Crédit Lyonnais : 4972
- Sofinco : 4976
- La Bred : 4975
- Crédit Agricole : 5131
- Société Générale : 4973

Nous n'énumérerons pas les préfixes de toutes les banques car nous pouvons en recenser plus de 1500 pour l'Amérique et l'Europe.

### 3 - Méthodes

```
`fonction F_isCardValide  
  
C_BOOLEEN($0)  
C_ALPHA(31;$1)  
  
$L:=Longueur($1)  
  
$total:=0  
`boucle inverse à partir de l'avant dernier numéro  
Boucle ($i;$L-1;-2)  
`on multiplie par deux  
$valeur:=Num($1≤$i≥)*2  
Si ($valeur>9)  
$string:=Chaine($valeur)  
Boucle ($j;1;2)  
$total:=$total+Num($string≤$j≥)  
Fin de boucle  
Sinon  
$total:=$total+$valeur  
Fin de si  
Fin de boucle  
  
`boucle inverse pour les  
`chiffres non comptabilisés  
Boucle ($i;$L;1;-2)  
$Total:=$total+Num($1≤$i≥)  
Fin de boucle  
  
$0:=(Modulo($total;10)=0)
```

```
`fonction F_TypeCarte
```

```
C_TEXTE($0;$1)
```

```

$prefix:=Sous chaine($1;1;3)
$Position:=Chercher dans tableau(tabPrefix;$prefix)
Si ($Position>0)
  $0:=tabCartes{$Position}
Sinon
  $0:="Carte inconnue"
Fin de si

```

## Comptes bancaires

---

Un RIB (relevé d'identité bancaire) est composé des éléments suivants :

- Le nom du titulaire du compte
- La domiciliation (le nom de la banque et sa ville)
- Le code banque (numéro de 5 chiffres)
- Le code agence (numéro composé de 5 chiffres)
- Le numéro de compte (composé de 11 chiffres)
- La clé RIB (numéro composé de 2 chiffres qui sont compris entre 01 et 97).

### 1 -Vérification d'un numéro de compte bancaire

- Remplacer dans le numéro de compte les éventuelles lettres par des chiffres :

- A et J seront remplacés par le chiffre 1
- B, K et S seront remplacés par le chiffre 2
- C, L et T seront remplacés par le chiffre 3
- D, M et U seront remplacés par le chiffre 4
- E, N et V seront remplacés par le chiffre 5
- F, O et W seront remplacés par le chiffre 6
- G, P et X seront remplacés par le chiffre 7
- H, Q et Y seront remplacés par le chiffre 8
- I, R et Z seront remplacés par le chiffre 9

- Concaténer tous les chiffres de votre RIB pour avoir un nombre de 23 chiffres (les 5 du code banque, les 5 du code guichet, les 11 de votre numéro de compte et les 2 de la clé RIB.

- Le reste de la division de ce nombre par 97 doit être nul.

- La clé RIB doit être comprise entre 01 et 97.

- Pour vérifier la clé, couper le nombre de 23 chiffres en 4 (3 morceaux de 7 chiffres et le dernier morceau de 2 chiffres).

<- 7 -> <- 7 -> <- 7 -> <2>  
A      B      C      CLE

Clé:=97-Modulo ((62\*A)+(34\*B)+(3\*C) ; 97 )

## 2 - Méthode

```
`fonction F_CalcRIB
C_ALPHA(21;$1)
C_ENTIER LONG($value)
C_ENTIER LONG($reste;$v1;$v2;$v3)
C_ALPHA(2;$0)

Si (Longueur($1)=21)
`recherche des caracteres non numériques dans la chaine
Boucle ($i;1;21)
$char:=$1≤$i≥
Si ($char>="A")
`calcul de la valeur
$value:=Code ascii($char)-(64+(9*Num($char>"J"))+(8*Num($char>"R")))
`remplacement en local
$1:=Remplacer chaine($1;$char;Chaine($value);1)
Fin de si
Fin de boucle

`décomposition en trois blocs de 7 caractères
$v1:=Num(Sous chaine($1;1;7))
$v2:=Num(Sous chaine($1;8;7))
$v3:=Num(Sous chaine($1;15;7))

`calcul clé RIB
$0:=Chaine(97-Modulo(((v1*62)+(v2*34)+(v3*3));97);"00")

`ajout de la clé Rib
$1:=$1+$0

`test validité du compte
`en fait un modulo de 97
$reste:=(Num($1)/97)-(Num($1)\97)

Si ($reste#0)
ALERTE("R.I.B. invalide..."&Caractere(13)+"Vérifiez votre saisie.")
$0:="XX"
Fin de si

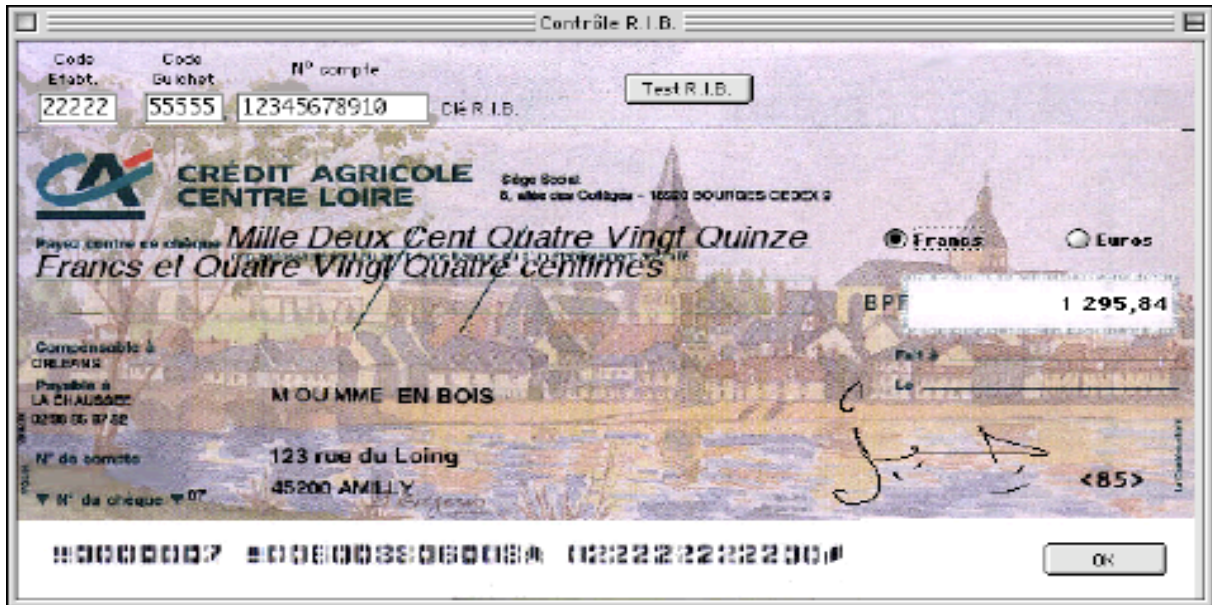
Sinon
ALERTE("Saisie du numéro de compte incomplète !")
$0:=""

Fin de si
```



## Base exemple

### Menu Chèque :



Saisie d'un numéro de compte bancaire et calcul de la clé R.I.B. par le bouton "Test RIB".

Méthode objet du bouton :

```
vCle:=F_CalcRib (vEtabl+vGuichet+vCompte)
```

Transcription en numéral d'un nombre réel.

Méthode objet de la variable vBPF :

```
vTexte:=" "*20+F_Num2String (vBPF;(bt2=1))
```

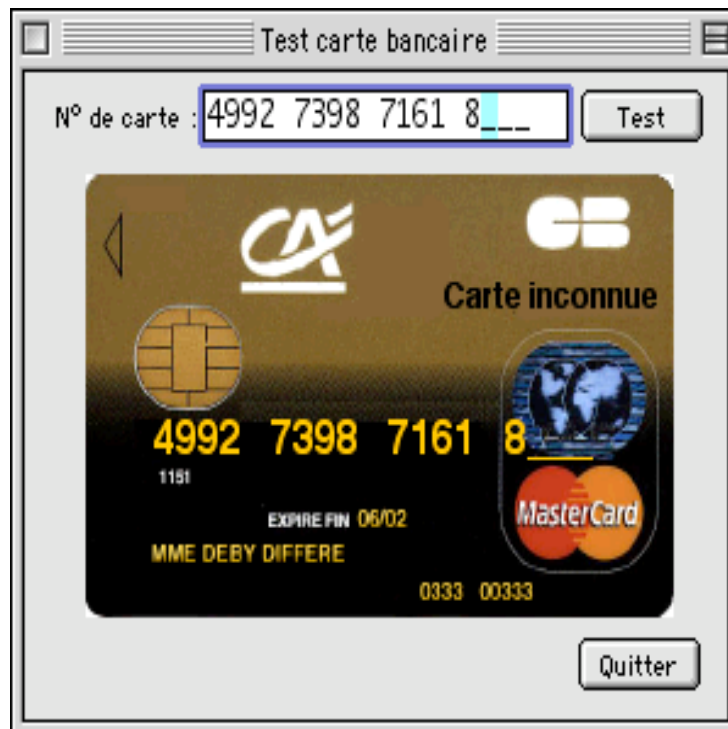
A noter que le deuxième paramètre est défini par les boutons radio francs/euros. Traité dans la fonction F\_Num2String par ce code :

```
.....  
Si ($?) \ on passe en Euros  
    $0:=Remplacer chaine($0;"centime";"cent")  
    $0:=Remplacer chaine($0;"Franc";"Euro")  
Fin de si  
.....
```

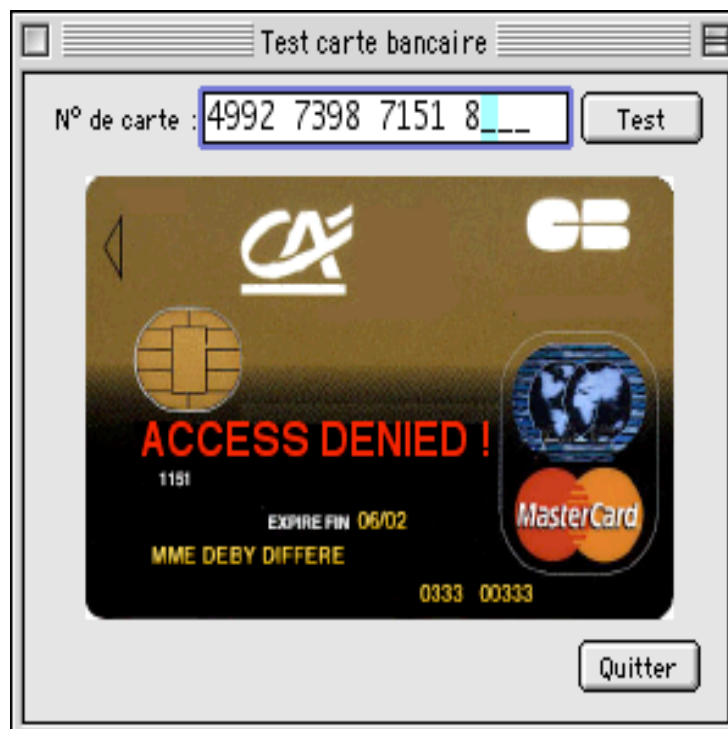


## Menu Carte :

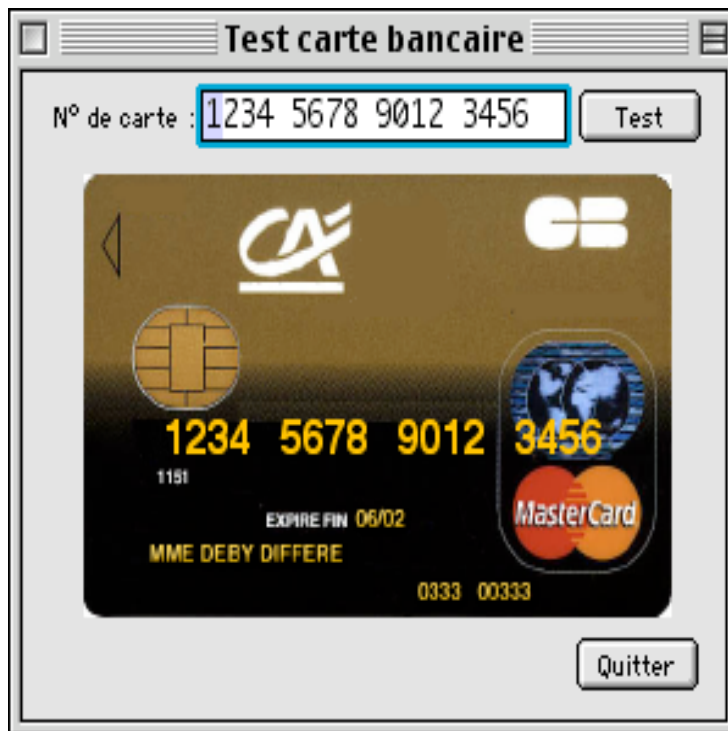
Avec notre exemple :



En changeant un des chiffres :



Avec un autre numéro (présumé valide) :



Méthode objet du bouton "Test" :

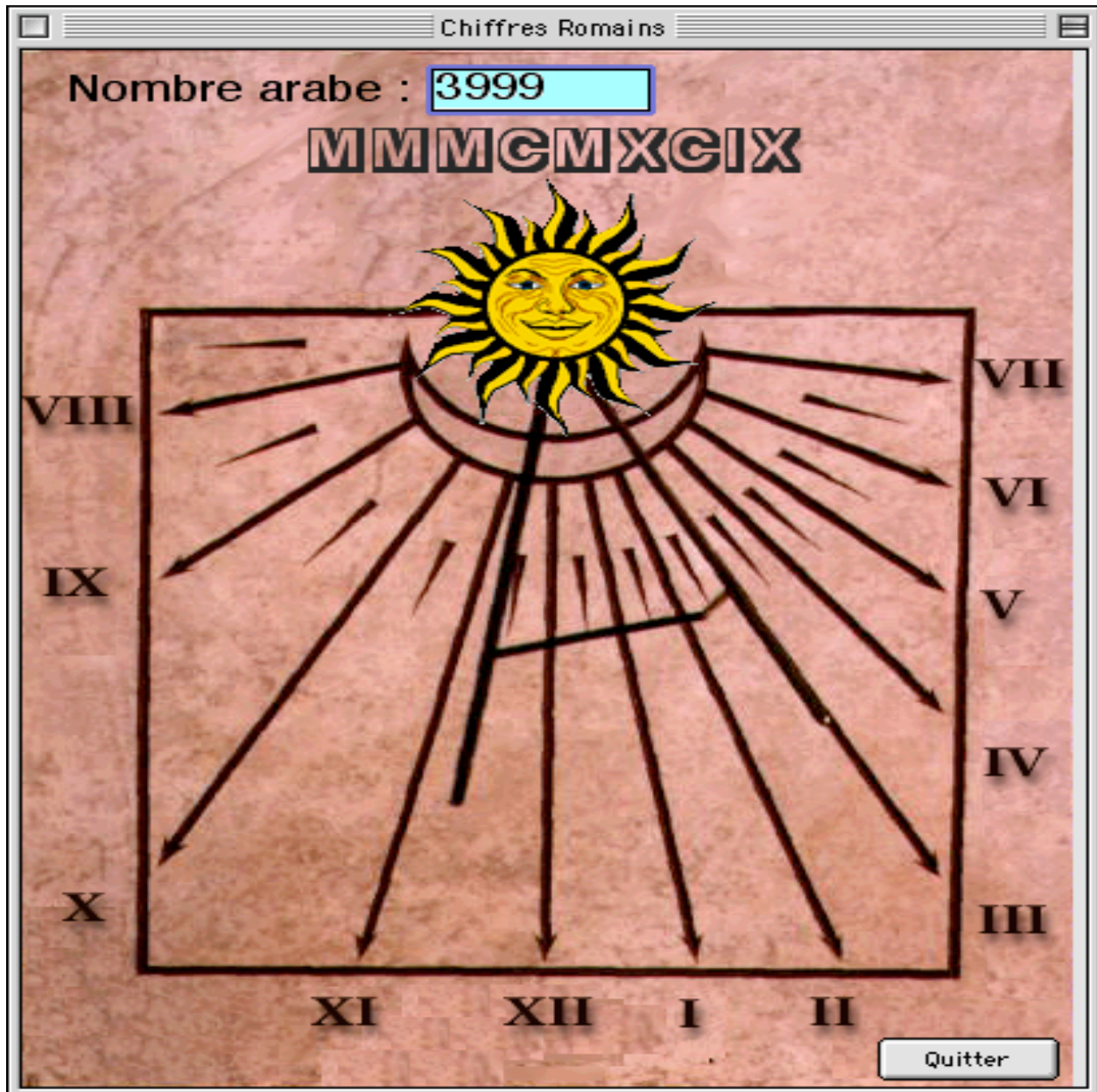
```
$ok:=F_isCardValide (vCarte)

Si ($ok)
  CHOIX VISIBLE(*;"numCarte";Vrai)
  CHOIX VISIBLE(*;"Refus";Faux)
  TypeCarte:=F_TypeCarte (vCarte)
Sinon
  BEEP
  CHOIX VISIBLE(*;"Refus";Vrai)
  CHOIX VISIBLE(*;"numCarte";Faux)

Fin de si
```

Note : ici nous devrions utiliser une méthode "Choix formatage" qui, dynamiquement suivant le prefixe en cours de saisie, modifie le nombre de chiffres permis (à vos claviers).

## Menu Romain :



Méthode objet de la variable vArabe :

```
vRomain:=F_Num2Romain (vArabe)
```