

# LE MICROMENSUEL du QUÉBEC

Volume #1 - Numéro #8  
Septembre 1993

Centre d'édition

SBE Info Services BBS - (514)278-7832/9685 - 14.4k/v.32bis  
Ouvert 24h tous les jours de la semaine.

Adresses réseau: 101:190/102

1:167/223

La nouvelle ligne voix du MMQ, (514)495-2610 24h/24h 7jours/sem  
Laissez message sur répondeur si je suis indisponible. Merci!

MAINTENANT DEUX LIGNES TÉLÉPHONIQUES POUR MIEUX VOUS SERVIR!

La ligne FAX du MMQ maintenant disponible au (514)278-1963.  
Pour avoir la plus récente copie du MMQ, faites une requête de MMQ  
sur SBE Info Services au travers de votre mailer.

Édition, rédaction et montage

L'édition et le montage de cette copie a été effectuée  
par Constantin Koutrias avec l'aide de Normand Julien  
pour la rédaction.

NOTE: Cette copie a été formatée pour impression.

Sommaire

Note de l'éditeur . . . . . Page 1  
 Formulaire d'adhésion pour le MMQ . . . . . Page 2  
 Le TurboPascal de Jean Gravel (5e partie) . . . . . Page 3  
 Le TOP 5 des Rigolignes . . . . . Page 7  
 L'histoire de Pi. . . . . Page 8  
 Piratage: Acheter, c'est payer. Pirater, c'est économiser?Page 15  
 Les Infos du mois . . . . . Page 17  
 L'informatique selon le professeur (2e partie). . . . . Page 20  
 Le RAPQList - Les fichiers du Québec. . . . . Page 31  
 Le babillard électronique du MMQ. . . . . Page 35  
 Le Courrier du lecteur. . . . . Page 36  
 Les Annonces classées. . . . . Page 37

Note de l'éditeur

Salut à vous tous!

Veillez nous excusez, encore une fois, du retard de cette édition. La qualité des articles présentés devrait toutefois vous faire oublier cet impair.

Notre nouvelle chronique ce mois-ci, la RAPQ-Liste, contient la liste des fichiers complets du RAPQ tel que compilé par Robert Marcoux et Philippe Gamache. Cette liste est d'autant plus intéressante qu'elle ne contient que des fichiers faits au Québec. Appuyons les programmeurs de chez nous!

J'aimerais remercier Gilbert Labelle qui a pris la relève de Normand Julien le mois dernier dans la correction du MMQ. Vacances obligent! Merci encore! :-)

Constantin Koutrias

Formulaire d'adhésion au MMQ

J'aimerais recevoir le MicroMensuel du Québec par mon "mailier" dès sa sortie. J'accepte qu'il soit placé en HOLD sur SBE Info Services si l'appel requis est interurbain pour SBE Info Services. Je ferai un poll le 1er de chaque mois.

Nom du babillard: [. . . . . ]  
 Nom de l'opérateur: [. . . . . ]  
 Numéro du babillard: [ . . . . . ]  
 Numéro de l'opérateur: [ . . . . . ]  
 Adresse réseau: [. . . . . ]

Ville: [ . . . . . ]

S.V.P. faire parvenir sous forme NetMail sur SBE Info Services.  
Merci de votre attention.

Constantin Koutrias

Le TurboPascal de Jean Gravel (5e partie)

## INTRODUCTION AU PASCAL: Procédures et fonctions

Jusqu'à maintenant, tous les exemples donnés se résumaient à de courts programmes, une suite d'instructions relativement simples. Nous n'avons pas encore exploité les avantages de la programmation structurée proprement dite.

Une façon de structurer un programme, c'est de le morceler en plusieurs procédures et fonctions. Tout comme l'instruction "Length" du Pascal renvoie la longueur d'une chaîne de caractères donnée, nous pouvons créer nos propres fonctions qui retourneront ce qu'on aura décidé qu'elles nous retournent. Dans le cas de la fonction "Length", on dit qu'on lui passe une variable en paramètre et qu'elle nous retourne un nombre qui représente la longueur de la chaîne.

Les suites d'instructions qu'on regroupe seront appelées "procédure" ou "fonction" selon qu'elles retournent ou non une information. Une fonction retourne une valeur donnée alors qu'une procédure ne retourne rien. Chacune, par contre, effectue une série d'instructions.

### Les fonctions

On définit une fonction comme s'il s'agissait d'un sous programme. On lui donne un nom, on lui passe des paramètres (c'est facultatifs) et on spécifie le type de résultat qui sera retourné au retour de son exécution. C'est l'objet de la ligne

```
function NombreRacines(A,B,C : real) : integer;
```

qui précise le début d'une définition de fonction (function), le nom de la fonction (NombreRacines), le(s) nom(s) du(des) paramètres (C1,C2,C3 : real) et le type de résultat qui est retourné (integer).

Suivant le dicton "une image vaut mille mots", examinons le prochain exemple:

```
Programme Exemple_11;
var
  C1,C2,C3 : real;  (* Coefficients du polynôme *)

  (* Vérifie s'il existe une racine, dans l'affirmative
     si elle est simple ou double. -1 = aucune racine,
     1 = racine simple, 2 = racine doubles. *)
  fonction NombreRacines(A,B,C : real) : integer;
  var
    B2,4AC : real;
  begin
    B2 := B * B;
    4AC := 4 * A * C;
    NombreRacines := (B2=4AC) - (B2<4AC) + 2*(B2>4AC);
  end;

begin
  writeln('Vérification de l''existence de racines');
  writeln('réelles d''un polynôme de degré 2 en x. ');
  writeln;
  write('Entrez les coefficients A, B et C: ');
  readln(C1,C2,C3);
  case NombreRacines(C1,C2,C3) of
    -1 : writeln('Ce polynôme ne possède aucune racine. ');
    1 : writeln('Ce polynôme a une seule racine. ');
    2 : writeln('Le polynôme a deux racines distinctes. ');
  end;
  readln;
end.
```

Bien que le programme ne calcule pas les racines, il montre comment on peut définir et utiliser une fonction qu'on a définie. Les fonctions, comme les procédures, doivent être définies dans la section des déclarations. Le programme principal s'interprète plus facilement. On lit les coefficients du polynôme, on effectue un calcul astucieux et, selon le résultat obtenu, on affiche un message particulier à l'écran. On a défini et utilisé une fonction qui reçoit 3 nombres réels en paramètre et qui retourne un nombre entier (integer) comme résultat.

#### Test et prise de décision

-----

Un programme doit pouvoir décider d'effectuer une chose ou une autre à un certain moment, pourvu que le programmeur lui ait "enseigné" comment prendre la décision. Il est souvent utile dans un programme de tester une condition donnée et de décider de la série d'instructions à suivre selon le résultat du test.

Le test "case ... of ..." utilisé dans le dernier exemple est une instruction Pascal qui prend des décisions. Ce test détermine

quelle instruction ou série d'instructions sera exécutée. On termine toujours un «case» par un "end;" ou par un "end" si une alternative est possible. Les éventualités sont listées entre le "case" et le "end;".

Le test "if ... then ... else ..." est un autre moyen pour le programmeur "d'enseigner" à son programme comment prendre des décisions. Il effectue les instructions du "then" si la condition est vraie ou les instructions du "else" si la condition est fausse.

La séquence d'instruction qui suit montre comment on peut décider de procéder au calcul des racines lorsqu'elles existent. On a créé une fonction qui retourne un nombre négatif (-1) s'il n'y a pas de racines et un nombre positif dans le cas contraire. On va utiliser le résultat de ce test pour prendre une décision.

```
if NombreRacines(C1,C2,C3) < 0
  then
  begin
    writeln('Ce polynôme ne possède pas de racines.')
    readln; (* Permet une pause *)
  end
  else
  begin
    (* Calcul des racines *)
  end;
```

On utilise la fonction "NombreRacines" dans le test. Si la fonction retourne -1, la condition est vraie et un message est affiché. Si par contre la condition se révèle fausse, c'est-à-dire que la fonction retourne 1 ou 2, ce sont les instructions du "else" qui sont exécutées.

Remarquez l'absence du point-virgule après le "end" dans la partie du "then". L'absence du point-virgule indique au compilateur la présence d'une alternative. Le compilateur ne trouvant pas le point-virgule habituel est "obligé" de lire la ligne suivante et "comprend", en y trouvant le mot "else", que le test a une alternative. Il arrive souvent qu'un test est sans alternative.

Les procédures

-----

Une procédure ne renvoie jamais de résultats. Par contre, on peut lui passer des paramètres pour qu'elle les utilise. Dans notre problème sur le calcul des racines, on pourrait diviser le programme principal en plusieurs sous-programmes qui, chacun, effectuerait une tâche particulière. Par exemple, on peut définir une procédure qui se charge de lire les coefficients, une

autre qui se charge de donner un message d'erreur lorsqu'il n'y a pas de racine, etc.

L'usage de procédures et de fonctions va nous permettre d'obtenir un programme structuré dont le corps principal parle de lui-même.

```
begin
  LitCoefficients;
  if NombreRacines < 0
    then Avertissement
    else
      case NombreRacines of
        1 : CalculeRacine(+1);
        2 : begin
              CalculeRacine(+1);
              CalculeRacine(-1);
            end;
      end;
end.
```

Le corps du programme utilise une fonction et deux procédures. Les procédures "LitCoefficients" et "Avertissement" effectuent une série d'instructions et la procédure "CalculeRacine" utilise un paramètre pour le calcul des racines. Voici ce que pourrait être le programme dans son ensemble:

```
Program Exemple_12;
var
  C1,C2,C3 : real;

procedure LitCoefficients;
begin
  writeln('Racines d'un polynôme de degré 2 en x');
  writeln;
  write('Entrez le coefficient de x^2: ');
  readln(C1);
  write('Entrez le coefficient de x: ');
  readln(C2);
  write('Entrez le terme constant: ');
  readln(C3);
end;

procedure Avertissement;
begin
  writeln('*****');
  writeln('* Ce polynôme ne possède pas de racines *');
  writeln('*****');
end;

procedure CalculeRacine(Signe : integer);
var Racine : real;
```

```

begin
  Racine:=(Signe * sqrt(C2*C2 - 4*C1*C2) - C2) / (2*C1);
  writeln(Racine trouvée: ',Racine:10:4);
end;

function NombreRacines(C1,C2,C3 : real) : integer;
var
  B2,4AC : real;
begin
  B2 := B * B;
  4AC := 4 * A * C;
  NombreRacines := (B2=4AC) - (B2<4AC) + 2*(B2>4AC);
end;

begin (* Programme principal *)
  LitCoefficients;
  if NombreRacines < 0
  then Avertissement
  else
    case NombreRacines of
      1 : CalculeRacine(+1);
      2 : begin
          CalculeRacine(+1);
          CalculeRacine(-1);
        end;
    end;
end;
end. (* Programme principal *)

```

Le dernier exemple montre bien jusqu'à quel point la structure d'un programme est importante. Lorsqu'on modifie un programme qui est bien structuré, on a du plaisir, mais lorsqu'on ne se préoccupe pas de la structure, on risque de détester à court terme la programmation.

Sur ce, je vous laisse explorer ces nouveaux concepts. D'ici la prochaine parution de MMQ, je vous propose de réfléchir à un programme qui pourrait effectuer des conversions binaires, octales, décimales et hexadécimales. Je vous donnerai les détails d'un programme qui effectue ces conversions dans le prochain numéro. Amusez-vous bien.

Jean Gravel

Le TOP 5 des rigolignes

1. A force de s'y frotter on reL.U.I.t
2. Plus fort!, j'ai des bananes dans les oreilles
3. Il ne faut pas ignorer l'ignorance...
4. Minuit l'heure du... des... users.....L.U.I.

5. EN GREVE [ FTQ (Fédération des Taglines du Québec) ]

Avis à tous ceux qui se reconnaissent. :-)

L'histoire de Pi

CURIOSITE MATHEMATIQUE = PI

---== Documentation Yves MARTIN ===--

Qui parmi nous peut se vanter d'avoir une mémoire infailible? Déjà nos professeurs de collège nous enseignaient avec talent et méthode (aussi barbares fussent-elles...) certains moyens mnémotechniques nous permettant de mémoriser la suite incohérente qu'est le nombre "π". Le premier texte, très connu, est cité par Beutel en 1913:

Que j'aime à faire connaître un nombre utile aux sages!

3. 1 4 1 5 9 2 6 5 3 5

Immortel Archimède, artiste ingénieur,

8 9 7 9

Qui de ton jugement peut priser la valeur?

3 2 3 8 4 6 2 6

Pour moi ton problème eut de pareils avantages

4 3 3 8 3 2 7 9

Tirez circonférence au diamètre etcetera.

5 0 2 8 8

Autre texte tout aussi connu:

Que j'aime à faire apprendre un nombre utile aux sages!

3. 1 4 1 5 9 2 6 5 3 5

Glorieux Archimède, artiste ingénieur,

8 9 7 9

Toi de qui Syracuse aime encore la gloire,

3 2 3 8 4 6 2 6

Soit ton nom conservé par de savants grimoires!

4 3 3 8 3 2 7 9

Jadis, mystérieux, un problème bloquait

5 0 2 8 8

Tout l'admirable procédé, l'oeuvre grandiose

4 1 9 7 1 6 9

Que Pythagore découvrit aux anciens Grecs.

3 9 9 3 7 5

O quadrature! vieux tourment du philosophe!

1 0 5 8 2 0

Insoluble rondeur, trop longtemps vous avez  
 9 7 4 9 4 4  
 Défié Pythagore et ses imitateurs.  
 5 9 2 3 0  
 Comment intégrer l'espace plan circulaire?  
 7 8 1 6 4 0  
 Former un triangle auquel il équivaldra?  
 6 2 8 6 2 0  
 Nouvelle invention: Archimède inscrira  
 8 9 9 8  
 Dedans un hexagone; appréciera son aire  
 6 2 8 0 3 4  
 Fonction du rayon. Pas trop ne s'y tiendra:  
 8 2 5 3 4 2 1 1 7  
 Dédoublera chaque élément antérieur;  
 0 6 7 9  
 Toujours de l'orbe calculée approchera;  
 8 2 1 4 8 0  
 Définira limite; enfin, l'arc, le limiteur  
 8 6 5 1 3 2 8  
 De cet inquiétant cercle, ennemi trop rebelle!  
 2 3 0 6 6 4 7  
 Professeur, enseignez son problème avec zèle!...  
 0 9 3 8 4 4

Un peu d'histoire...

O. Neugebauer aurait déchiffré sur une tablette babylonienne datant de 5 000 ans, le chiffre PI avec 5 décimales!...

Le célèbre papyrus Rhind, vieux de 35 siècles affirme: "l'aire du cercle de diamètre 9 coudées est celle du carré de 8 coudées". Étonnante affirmation pour l'époque!

Dès l'aube des temps, PI ou le rapport de la circonférence et du diamètre d'un cercle en intrigua plus d'un. Dans la version synodale protestante de la bible, on dit: "il fit aussi, en métal fondu, la grande cuve qui avait six coudées de diamètre et qui était complètement ronde; elle avait cinq coudées de haut et un cordon de trente coudées en mesurait la circonférence." Donc l'immense vasque qui ornait le temple du roi Salomon mesurait trente unités de circonférence pour un diamètre de dix unités. Soit un rapport de trois très exactement (circonférence/diamètre). Comment Dieu, grand mathématicien et créateur universel eut pu admettre une erreur aussi monumentale dans un texte tel que la bible! Ainsi pour Dieu, PI serait égal à 3.

Aux Indes, les seuls documents qui nous soient connus, les Siddhantas ou systèmes astronomiques, donnent en 380 après J.-C. la

valeur "3+177/1250" ou "3.1416", basée sur une numérotation sexagésimale (base 60) comme firent les anciens Grecs.

Quant aux chinois, dont le système de numérotation a toujours été décimal, ils calculèrent la valeur de PI à partir d'un polygone de 192 côtés (en 294 après J.-C.). Deux siècles plus tard et avec un polygone de 3072 côtés, ils affirmèrent que PI est compris entre 3,1415926 et 3,1415927. La précision sur la huitième décimale ne sera atteinte en Europe qu'au XIIème siècle.

Beaucoup plus connus, grâce aux vestiges de la bibliothèque d'Alexandrie et au zèle culturel des trois premiers Ptolémée, sont les travaux des mathématiciens Grecs des premier et second siècle avant J.-C. Ils semblent avoir été les premiers à poser le problème de la quadrature du cercle, c'est-à-dire construire un carré dont la surface est égale à celle d'un cercle donné (qui hantera les chercheurs jusqu'en 1882). Antiphon, à la même époque énoncera le principe suivant: un polygone inscrit dans un cercle et dont on double à chaque fois le nombre de côtés, finit par s'identifier au cercle lui-même, et donc, sa surface est calculable par quadrature comme celle d'un polygone (\*).

\* On sait maintenant que c'est faux par la notion de limite. Le mathématicien français J. H. Lambert (1728-1777) démontra en 1768 que PI est irrationnel (Si  $\text{tg}(\pi/4) = 1$  est rationnel alors  $\pi/4$  est irrationnel.) Puis Le Gendre (1752-1833) montra que PI au carré est aussi irrationnel... En 1882, Lindemann démontrera la transcendance du nombre PI et donc l'impossibilité de la quadrature du cercle. Il aura quand même fallu attendre 25 siècles pour démontrer l'impossibilité de la quadrature du cercle!

En Europe, c'est au IIIème siècle avant J.-C. que commença, suivant l'initiative d'Euclide (IIIème siècle avant J.-C.) et d'Archimède (287-212 avant J.-C.), un véritable travail mathématique sur le cercle. Archimède proposa, dans un petit traité, la première valeur approchée significative de PI,  $PI=22/7$  qui restera longtemps en usage, résultat qui est une valeur approchée au 2500ème, remarquable pour l'époque. Archimède utilisa deux polygones réguliers de 96 côtés pour obtenir son approximation! C'est l'approfondissement de la même idée qui fera dire à Kepler (1571-1630) et Galilée (1564-1642), au XVIIème siècle, que "le cercle est un polygone à une infinité de côtés".

Au XVIème siècle, la course aux décimales commença. Adrian Anthoniszoon (1527-1607), un Hollandais, donne PI avec 6 décimales exactes,  $PI = 355/113$  qui sera en fait publié par son fils Adrian Métius en 1625 et qui expliqua que son père avait utilisé la méthode d'Archimède:  $333/106 < \pi < 377/120$  et avait fait la moyenne des numérateurs et des dénominateurs. Le français Viète (1540-1603), en 1593, en trouve 9 à l'aide d'un polygone de 393216 côtés. La même année le Hollandais Adrien Van Rooman (1561-1615) parvient à 15 décimales. Trois ans après un autre Hollandais, Ludolf Van Ceulen (1539-1610) ou

"Ludolph de Cologne", parvient à 20 décimales puis 35 (vers 1600) et à la gloire. En Allemagne, PI porte aussi le nom de "nombre de Ludolph".

John Wallis (1616-1703), à qui l'on doit, entre autres, le signe de multiplication "x", et les signes "<" et ">" d'inégalité, et surtout le signe "∞" de l'infini, en font un célèbre précurseur des mathématiques d'aujourd'hui. J. Wallis publia ses recherches sur PI en 1655 et obtint une expression sous forme d'une fraction continue :

$$PI/2 = 2.2.4.4.6.6.8.8..../1.3.3.5.5.7.7.9....$$

Newton (1642-1727) étudia l'oeuvre de Wallis durant l'hivers 1664-65 et la compléta. C'est à partir de là que Leibniz obtint sa célèbre formule d'arctangente sous forme de série...

Il y eut une période sombre où PI fut mêlé à l'argent et au péché originel, ...heureusement PI ne devait pas souffrir de ces aberrations.

Avec le calcul différentiel, le XVIIème siècle commence une nouvelle étape basée sur les séries, les calculs devenant plus faciles avec les logarithmes. Abraham Sharps (1651-1742) arriva à 72 décimales, John Machin (1680-1752) publiera 100 décimales en 1706 à partir de la différence entre deux arctangentes. En 1717, le Français de Lagny (1660-1734) en donne 127, record battu seulement en 1794 par Vega (1754-1802) avec 140 décimales et la découverte que la 113ème décimale donnée par de Lagny, un 8 est fautive, c'est un 7! Dès le XVIIIème siècle, Chinois et Japonais participent à la chasse aux décimales. Le XIXème siècle pulvérisera les records...

A partir du XVIIIème siècle, les progrès faits en mathématiques conduisirent à de nouvelles méthodes de calcul. On a découvert de nouvelles formules grâce à la trigonométrie telle celle de Gregory (en 1670) et de Leibniz (en 1673):

$$\arctg(x) = x - x^3/3 + x^5/5 - x^7/7 + x^9/9 - x^{11}/11 + \dots \text{ à l'infini} \dots$$

Les travaux de Gregory (1638-1675), professeur à l'université d'Edeinbourg, sont importants. Il est le premier à tenter de démontrer que la quadrature du cercle est impossible. Gregory développera la formule ci-dessus et Leibniz sera le premier à l'adapter pour PI sous une forme plus simple en remplaçant "x" par "1" et qui s'écrira:

$$\arctg(1) = PI/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \dots \text{ à l'infini} \dots$$

.....

En août 1949, le professeur John Von Neumann utilisa un des premiers ordinateurs: l'"ENIAC", pour calculer PI avec 2037 décimales. Les calculs prirent un weekend ou un peu plus de 70 heures, utilisant les

efforts de 4 "ENIAC" (trente tonnes d'électronique), et des ingénieurs qui assurèrent un suivi continu du calcul. Le calcul était basé sur une formule utilisée par J. Machin en 1706 :

$$\text{PI} = 16 \arctg(1/5) - 4 \arctg(1/239)$$

Cette formule permit à Shanks de déterminer A LA MAIN, 707 décimales (seulement les 528 premières étaient exactes).

Grâce à la formule d'Euler (1707-1783), développement astucieux, Euler avait réussi à calculer à la main 20 décimales en 1 heure :

$$\text{PI} = 20 \arctg(1/7) + 8 \arctg(3/79)$$

Actuellement, une méthode mise au point en 1975, permet aux puissants ordinateurs de calculer des centaines de millions de décimales en quelques heures. Aujourd'hui, plus de 16 millions de décimales sont connues...

Un pas très important a été effectué grâce à Euler qui démontra la relation entre PI et "e" (base des logarithmes népérien) :

$$e^{(i\pi)} = -1 \text{ où } i = \sqrt{-1}$$

... Qui exprime "peut-être" que "e" et PI ne sont des solutions d'aucune équation algébrique

Mais, me direz-vous, à quoi sert ce long calcul si ce n'est de satisfaire la curiosité malade des mathématiciens de tous les temps? Il existe quelques raisons intéressantes :

- les méthodes de calculs de PI ont souvent permis de nouveaux développements mathématiques contribuant ainsi à la découverte de nouvelles techniques.
- la précision du chiffre PI, avec un grand nombre de décimales, permet tout simplement (hic...!), de contrôler la bonne marche des ordinateurs.
- la course aux décimales de PI permet aux ordinateurs de grande puissance de tester leurs capacités et d'organiser des championnats internationaux. Après l'ENIAC en 1949, voici quelques records:
  - En novembre 1954 et janvier 1955, le NORC, de la marine américaine, en Virginie, calcula PI avec 3089 décimales en 13 minutes.
  - En mars 1957, le centre de calcul Ferranti, de Londres donna 10021 décimales en 33 heures avec une erreur à la 7480ème.
  - A Paris, en juillet 1958, un IBM 704 parvint à 10000 décimales en 1 heure et 40 minutes.
  - En juillet 1959, une même machine du CEA à Paris donna 16167

décimales en 4 heures et 20 minutes.

- Daniel Shanks et Wrench adoptent la formule de Störmer :

$$\text{PI}/64 = 3/4 \arctg(1/8) + 1/8 \arctg(1/57) + 1/16 \arctg(1/239)$$

... qui donnera 100265 décimales en 8 heures et 43 minutes.

- Un IBM 7030, en février 1966, a donné 250000 décimales

- Puis, en février 1967 un CDC 6600, donna 500000 décimales en 28 heures 10 minutes (d'après la formule de Störmer).

- En 1974, J. Guilloud et M. Bouyer, calculèrent 1000000 décimales avec un CDC 7600. Le calcul a demandé 22 heures 11 minutes et 1 heure 07 minutes pour la conversion en décimales. Il a été effectué à l'aide de la formule de Gauss :

$$\text{PI}/64 = 3/4 \arctg(1/18) + 1/2 \arctg(1/57) - 5/16 \arctg(1/239)$$

La vérification a été faite au CERN en utilisant la formule de Störmer.

Récemment, Darry Hills et son équipe du MIT (Massachusetts Institute of Technology) aux États-Unis ont mis au point un supercalculateur ayant les capacités de mille milliards de flops (1 flop = 1 opération arithmétique/seconde), en comparaison des 5000 flops de l'ENIAC voire des un milliard de flops du CRAY 3, ... et on attend mille fois mieux pour la fin du siècle (... du millénaire...)! - on a bien sûr permis aux mathématiques modernes d'évoluer considérablement grâce aux calculateurs puissants qui permettent des calculs très rapides et d'en trouver des applications dans la recherche, l'industrie, la météorologie, l'astronomie, etc...

Les STATISTIQUES...

Les statistiques? On en fait pour les jeux de casinos, le LOTO, etc. Tout le monde en fait alors pourquoi pas avec PI! Mais à quoi cela va servir puisque nous savons que PI est un nombre transcendant? Des questions se sont posées dès que nous avons réussi à calculer "π" ou "e" ou encore "c" avec un grand nombre de décimales. Si on ne peut mettre en doute leur transcendance, les statistiques pourront peut-être trouver un petit quelque chose qui éveillera en nous la petite étincelle endormie... et c'est si facile avec un ordinateur... Bref, pour ne plus vous laisser dans le doute voici pour le premier million de décimales ce qu'on peut dire :

les "0", il y en a :	99959
les "1", il y en a :	99758
les "2", il y en a :	100026
les "3", il y en a :	100226
les "4", il y en a :	100230

les "5", il y en a :	100359
les "6", il y en a :	99548
les "7", il y en a :	99800
les "8", il y en a :	99985
les "9", il y en a :	100106

- première constatation, la répartition des chiffres est à peu près équitable.
- Pour ce même million de chiffres la valeur moyenne de la somme des carrés des écarts est 900000 très exactement.
- aucune suite n'a été trouvée sur ce petit million ...
- PI n'est pas un nombre aléatoire, à tout le moins, on ne le présente pas tel quel (puisqu'on peut le calculer). Néanmoins, pourquoi ne pas utiliser ce million de chiffres comme font certains jeux de simulation (méthode de Monte-Carlo) qui utilisent des tables de chiffres aléatoires ...

Enfin, arrêtons là nos calculs car des pages pourraient être remplies et vous n'en sauriez pas plus !

Mais PI garde encore beaucoup de secrets à découvrir, c'est un nombre irrationnel, ça n'est pas le quotient de deux nombres entiers, aucune relation n'existe entre PI et les puissances de PI, aucune suite n'existe dans l'apparition des décimales, même sur des millions de décimales...

On sait que c'est la seizième lettre de l'alphabet Grec (qui correspond au "p" français), on sait aussi que la millième décimale est un 9, la trois millième est un 1, la dix millième un 8, la cent millième un 6... (utile pour contrôler vos calculs) .

Bref pas grand chose n'est connue aujourd'hui sur PI, alors avant que la fin des temps n'arrive, tous à nos neurones.

.....

### Calcul de PI

Précision du calcul de PI avec les 1000 premières décimales.

PI = 3.

14159	26535	89793	23846	26433	83279	50288	41971	69399	37510	(50)
58209	74944	59230	78164	06286	20899	86280	34825	34211	70679	(100)
82148	08651	32823	06647	09384	46095	50582	23172	53594	08128	
48111	74502	84102	70193	85211	05559	64462	29489	54930	38196	
44288	10975	66593	34461	28475	64823	37867	83165	27120	19091	(250)
45648	56692	34603	48610	45432	66482	13393	60726	02491	41273	

72458	70066	06315	58817	48815	20920	96282	92540	91715	36436	
78925	90360	01133	05305	48820	46652	13841	46951	94151	16094	
33057	27036	57595	91953	09218	61173	81932	61179	31051	18548	
07446	23799	62749	56735	18857	52724	89122	79381	83011	94912	(500)
98336	73362	44065	66430	86021	39494	63952	24737	19070	21798	
60943	70277	05392	17176	29317	67523	84674	81846	76694	05132	
00056	81271	45263	56082	77857	71342	75778	96091	73637	17872	
14684	40901	22495	34301	46549	58537	10507	92279	68925	89235	
42019	95611	21290	21960	86403	44181	59813	62977	47713	09960	(750)
51870	72113	49999	99837	29780	49951	05973	17328	16096	31859	
50244	59455	34690	83026	42522	30825	33446	85035	26193	11881	
71010	00313	78387	52886	58753	32083	81420	61717	76691	47303	
59825	34904	28755	46873	11595	62863	88235	37875	93751	95778	
18577	80532	17122	68066	13001	92787	66111	95909	21642	01989	(1000)

Note: Je distribue "gratuitement" quelques programmes écrits en Turbo-Pascal, en C et en Assembleur sur le calcul de PI. Communiquez avec moi au 514-279-8753.

---\*\*\*---

Yves Martin

Piratage: Acheter, c'est payer. Pirater, c'est économiser?

Bonjour cher camarade lecteur,

Après avoir discuté de mon dernier article (programmer, c'est frauder...) avec quelques copains et réfléchi sur le sujet, je me suis rendu compte que, bien souvent, le pirateur pensant se "venger" de l'abus des auteurs, ou piratant pour une raison quelconque se fait bien avoir bien malgré lui.

Je m'explique. Vous avez sans doute remarqué, comme tous ceux qui se tiennent au "courant", que le prix des logiciels et utilitaires ont chuté dramatiquement. Ceci m'a amené à réfléchir sur la validité d'un des plus vieux arguments dont les pirateurs aiment se servir pour justifier leur action à savoir "c'est trop cher pour ce qu'on a". Je n'ai pas l'intention de donner de grosse lecture sur la légalité ou non du piratage, je ne suis pas votre mère ou votre ange gardien, il est du ressort de votre conscience de vous aviser sur ce point. Cependant, quelques points sont intéressants à noter.

Imaginons que "Ti-Joe connaissant" décide de pirater la dernière version de Windows de son copain parce qu'il trouve que \$69 c'est trop demander pour ce logiciel et, de toute façon, Microsoft fait des millions, etc. Combien cela risque-t-il de lui coûter? Tout d'abord, une boîte de disquettes, disons \$8.-, ensuite, évidemment après quelques semaines, il lui faudra, soit

photocopier le bouquin ce qui est relativement cher et donne une documentation plutôt "cheap" ou mieux encore, il devra acheter un livre qui lui donnera tous les "trucs" et astuces parce qu'il ne pensera pas utiliser le logiciel à son plein rendement ou aura peut-être un problème que ses copains ne peuvent solutionner et il ne peut évidemment pas appeler le support technique.

Un bon bouquin qui contient autre chose que des renseignements évidents par eux-mêmes coûte dans les \$50.-. Le coût de son piratage: \$58.-. Il aura donc sauver un gros \$11 (et on ne parle pas des coûts spirituels si Ti-Joe à le moindrement une morale sensible ou des coûts légaux si jamais on le pince à pirater) .... Whoa! Quelle "leçon" il aura donné à Microsoft cette fois-ci! Ils comprendront surement le message ... <grin>

C'est d'autant plus ironique que la plupart des pirates comme Ti-Joe ne se rendent même pas compte qu'ils se sont fait avoir car, sur le moment, la copie est effectivement moins cher que l'achat. En effet, le bouquin que Ti-Joe s'achètera est, la plupart du temps, acheter quelque temps APRES la copie, le pirateur oublie donc de compter le coût de ce livre dans le prix de son piratage... nous connaissons tous, sans nul doute, plusieurs personnes qui ont piraté un logiciel quelconque et ont ensuite dû acheter un bouquin pour devoir l'utiliser correctement ou efficacement.

Autre exemple, Dbase IV 2.0, le standard dans les bases de données est récemment tomber à \$99.-. Il lui aurait coûté environ \$6- de disquettes pour le copier, puis un livre d'introduction sur Dbase IV pour connaître les nouveautés (\$50) et probablement un autre si Ti-Joe désire faire de la programmation avancé (\$50). Coût de son "piratage", \$106.- !!! Il aurait payé MOINS en achetant le logiciel et aurait, en plus, tout le support technique (sans compter les prix préférentiels futurs, etc). Nous pourrions donner des dizaines d'exemples semblables mais je crois que vous avez compris le point que nous voulions soulever: le piratage ne paie pas plus autant qu'auparavant.

Serait-ce que les compagnies de logiciels ont finalement compris que la manière de tuer le piratage ou à tout le moins le réduire c'est de rendre celui-ci non lucratif (enfin!)?

Nous ne jouons tout de même pas à l'autruche, nous savons bien que pour ceux qui pratiquent le piratage de logiciel et qui ont des valeurs de plusieurs centaines voire millier de dollars (vous vous connaissez, pas besoin de vous nommer <grin>) celui-ci a sans doute du bon du point de vue "économique", mais pour le commun des petits pirates du dimanche il semble, au grand bonheur des acheteurs honnêtes que la guerre au piratage a finalement été déclarée. Reste à voir qui en sortira vainqueur.

Pendant ce temps, je me procure enfin les logiciels que je ne pouvais me permettre auparavant sans me ruiner et les pirates eux deviennent maintenant des copieurs .... sélectifs ;-)

Salut et au mois prochain!

Yvon Decelles

Les infos du mois
-------------------

- Avec l'apparition du PENTIUM, voici une récapitulation des différents processeurs mis sur le marché depuis l'apparition des premiers ordinateurs personnels compatibles PC:

- 1974	µP 8080	2 MHz	Bus 8 bits	0,0064 MIPS	*
- 1978	µP 8086	5 MHz	Bus 16 bits	0,33	MIPS
- 1979	µP 8088	5 MHz	Bus 16 bits	0,33	MIPS
- 1981	(IBM PC)				
- 1982	µP 80286	8 MHz	Bus 16 bits	1,2	MIPS
- 1984	(IBM PC AT)				
- 1985	µP 80386DX	16 MHz	Bus 32 bits	6	MIPS
- 1986	(COMPAQ Deskpro 386)				
- 1988	µP 80386SX	16 MHz	Bus 32 bits	2,5	MIPS
- 1989	µP 486DX	25 MHz	Bus 32 bits	20	MIPS
- 1991	µP 486SX	20 MHz	Bus 32 bits	16,5	MIPS
- 1992	µP 486DX2	50 MHz	Bus 32 bits	40	MIPS
- 1993	µP PENTIUM	66 MHz	Bus 64 bits	112	MIPS
	(...)				

\* MIPS: Million d'Instructions Par Seconde.

Depuis la sortie du PENTIUM, les ordinateurs "tête de réseau", ou destinés à faire de gros calculs en un temps record pour la CAO, les instituts de recherche, etc... sont heureux. Une firme américaine a adapté la technologie "multiprocesseurs" aux compatibles PC.

Qu'est-ce que le multiprocesseur? Cette technologie de pointe est utilisée pour de très gros systèmes. Elle permet d'utiliser plusieurs processeurs pour faire la tâche que ferait un seul processeur normalement. Ces processeurs sont mis en parallèle et traitent ainsi un plus grand nombre de données en même temps. Pour un processeur 16 bits par exemple, on utilise 16 processeurs 1 bit en parallèle et on traite ainsi des mots de 16 bits très rapidement.

Bref, cette firme américaine utilise donc DEUX 386, deux 486DX2 ou encore DEUX PENTIUMs pour accélérer ses performances. Des cartes accélératrices pour les disques durs, CD-ROM, etc... et pourvues de caches de mémoire sont utilisées pour permettre aux "DUAL" processeurs de suivre la cadence. Dans ces conditions, un "486DX2DUAL" serait plus rapide que le PENTIUM! (et je ne parle pas du "DUAL PENTIUM"...) )

Cette firme, VTECH, propose ses modèles "Platinum" avec "DUAL" processeur ou, par l'ajout d'une carte, de transformer votre "SINGLE" processeur en "DUAL" processeur. Je ne sais pas à l'heure actuelle si ces cartes sont

compatibles avec les autres modèles de PC. (des infos suivront...)

- Covox Inc. a sorti son "Voice Blaster". C'est un ensemble permettant de reconnaître plusieurs centaines de commandes vocales, et qui est adapté pour Windows via un logiciel et un petit module que l'on branche sur son port parallèle. Ce module est transparent pour l'imprimante puisqu'il ne prive pas pour autant ce port. Sur ce module est branché un casque muni d'un micro. Fini les commandes compliquées utilisant des triples fonctions avec des clés cachées. En un mot votre ordinateur vous comprends...

- IBM a aussi sorti un "Voice Type Control" pour Windows et adapté à la "Sound-Blaster". Un autre type de reconnaissance vocal.

- Apple et Sharp ont mis au point le "Newton". C'est un bloc-note intelligent. Plus de clavier, une reconnaissance de l'écriture, un écran tactile et son crayon, un processeur Risc d'ARM à 20 MHz font de cet ordinateur le bloc-note de 94.

- L'ordinateur le plus rapide du monde est arrivé. Le DECpc AXP-150, destiné à Window NT, travaille à une fréquence de 150 MHz et à une vitesse de 300 MIPS !

- Philips sort ses nouvelles cartouches numériques en 3DO. Ce nouveau format numérique est destiné au standard multimédia mais aussi permet d'utiliser des cassettes numériques pour voir un film ou jouer sur un jeu. Depuis le succès du CD-I, Philips affirme en avoir vendu plus de cent mille à travers le monde. Le numérique est omniprésent. Philips a d'ailleurs signé avec la "Paramount Pictures" pour adapter son catalogue de films CD-I et pour le futur 3DO.

- Hewlett Packard distribue son HP96LX destiné aux hommes d'affaires. Ce calculateur est équipé d'un récepteur qui permet (avec un abonnement adéquat) de recevoir les cours de la bourse, les dépêches, les taux d'intérêts dans le monde, etc... et le tout tient dans la poche.

- Après l'annonce d'IBM concernant son DOS 6.2, 32 bits et multitâches, Novell dévoile son futur DOS 7. Après avoir racheté Digital Research (DR DOS), le futur système d'exploitation s'appellera donc "Novell DOS 7", capable de faire fonctionner son système sous réseau et sous Windows. Annoncé pour septembre, et au même prix que DOS 6.0. Microsoft a de quoi se faire des cheveux blancs!

- Atari annonce "Jaguar", sa future console de jeux conçue autour d'un processeur Risk 64 bits.

- La sortie du PENTIUM et des "Overdrives" pour 486 ont été un peu ralenties à cause de problèmes de fabrication liés à la surchauffe. L'overdrive a même été redessiné.

- Attention aux méprises futures... Intel remplace ses 486 SX, DX, et

DX2 par des 486SL intégrant des systèmes d'économie d'énergie et de basse tension (3.3V et 5V) et ce, sans supplément de prix.

- AST achète Tandy... contrôlant ainsi les chaînes de magasins tel que Radio-Shack et Tandy en France.

- "AMD n'avait aucun droit sur la technologie 386 d'Intel". Ce procès qui a commencé en 1987, ressemble à un digne feuilleton américain! La cour d'appel du sixième district de Californie vient d'annuler une décision d'arbitrage datant de février 1992 et qui accordait à AMD le droit de produire et de vendre des processeurs compatibles 386 en utilisant la technologie Intel. AMD compte bien faire appel auprès de la Cour suprême de Californie. A suivre...

- Le bloc-notes du mois: le "Philos 33", boîte à malice d'Olivetti. De plus en plus, les bloc-notes ressemblent à de grand jeux de constructions, et au grand plaisir des utilisateurs. Philos 33 est doté d'un 386SL cadencé à 20 MHz, d'un clavier amovible sous lequel se cache un lecteur 3.5", la batterie, un disque dur de 80 Mo, et une alimentation extra plate permettant ainsi d'éviter de transporter son alimentation externe, le tout extractible très facilement. En plus du clavier, un carré de 16 touches supplémentaires permettent d'accéder directement au chiffres et curseurs sans être obligé d'activer des fonctions spéciales propres aux bloc-notes. Un écran de control LCD permet de voir l'état des différents organes du bloc-notes. Des emplacements sont prévus pour un co-processeur et des barrettes de mémoires de 4 Mo chacune (extensible à 32 Mo). Comme son petit frère, le "Quaderno", "Philos" possède un haut-parleur interne et un micro. VoiceRec, fournis avec le Philos, est un logiciel qui permet de digitaliser la voix et de l'enregistrer sur le disque dur à raison de 10 minutes par Mo. Le Philos 33 est aussi livré en standard avec DOS 6.0, Windows 3.1 et Lotus Organizer.

Vous avez peut-être entendu parler de la carte "Vsound 3D" ou lu des articles dans le journal de Montréal? Non? ... alors je vais vous en dire plus. La Vsound est un produit québécois (savez-vous que les cartes graphiques pour PC les plus rapides au monde sont québécoises? ... mais ça c'est une autre affaire...). La carte Vsound devrait, selon son ambitieux concepteur, Hien Vohoang de la société Sixgraph Computing, être révolutionnaire. Monsieur Vohoang veut tout simplement devenir numéro un mondial! Pour réussir, il devra faire face à Texas Instrument, Analog Device, Creative Lab (C.A. 700M\$US, pour cette seule compagnie), ou d'autres encore... Mais la Vsound a déjà pas mal d'atouts. Construite autour de deux processeurs, un pour l'interface MIDI, l'autre pour le son 3D, vous pourrez écouter le galop d'un cheval et le situer dans un endroit précis de votre salon. Le 10 novembre prochain, cette carte sera présentée au Comdex de Las Vegas, plus grand salon audio au monde. Toujours d'après monsieur Vohoang, les performances techniques de la Vsound devraient surpasser les cartes les plus perfectionnées déjà existantes. Mieux que la "Surround sound", cette carte devrait être extraordinaire! Notes: Monsieur Vohoang détient deux doctorats en physique électronique et en génie électronique de l'université de Lyon. Il s'est établi à Montréal

après avoir enseigné le génie électrique à Trois-Rivières pendant quatre ans. Il a fondé Sixgraph en 1987.

Yves MARTIN

L'informatique selon le professeur (2e partie)

Bonjour à vous tous. Ce mois dans ce module;

## UNITE 2: CONCEPTS ET ÉLÉMENTS DU TRAITEMENT CENTRAL

- 2.1 Emplacement en mémoire et adresses.
  - 2.2 Définition des unités de données.
  - 2.3 Le code binaire
  - 2.4 Le système hexadécimal
  - 2.5 Le système octal
  - 2.6 Le système décimal
- 

### INTRO

Contrairement à ce que certains peuvent penser, il n'existe pas vraiment de similarité entre la mémoire humaine et la mémoire d'un ordinateur. Les deux servent à garder des données pour traitement mais là s'arrête la comparaison. Nous verrons dans le prochain cours, des éléments physiques de la mémoire mais ce mois-ci nous toucherons surtout l'adressage de celle-ci. Comme la capacité d'adressage de la mémoire se fait en mode binaire pour l'ordinateur et que pour bien comprendre ceci nous devons connaître les différents formats d'emmagasinement disponibles, nous avons pensé traiter ces sujets simultanément. Il y a un peu de mathématiques dans ce module. Certains d'entre vous seront sûrement tenter de passer par dessus mais je vous le déconseille vivement. Lorsque viendra le temps de traiter l'assembleur sur unité centrale et la manipulation d'adresse vous serez très heureux de déjà maîtriser ce domaine.

- 2.1 Emplacement en mémoire
- 2.2 Définition des unités de données
- 2.3 Le code binaire.

Lorsque l'on parle de la mémoire d'un ordinateur, on parle de bit. En effet, le bit représente la plus petite quantité d'information que l'on peut manipuler et l'unité de base avec laquelle l'ordinateur travaille. Quand on se rend compte que les micro de nos jours peuvent facilement utiliser des millions de bits pour un seul logiciel, on commence à prendre conscience de la quantité d'informations qui est manipulé. La mémoire d'un ordinateur est composée de milliers de cellules, chacune ayant une charge ou non. On dit d'une cellule chargée qu'elle a la valeur 1 et d'une cellule non chargée qu'elle a la valeur 0, une cellule équivaut à un bit. Si votre PC a une mémoire de 1 Meg, vous avez donc 8,388,608 bits de disponible sur lequel votre ordinateur peut travailler. Nous reviendrons sur ce sujet un peu plus tard.

Il est important de se rappeler qu'il n'existe aucune valeur intermédiaire entre 1 et 0 pour votre ordinateur, pas de fractions. En fait, ce ne sont pas des valeurs mais bien des charges électriques et celles-ci sont INTERPRÉTÉES comme étant des valeurs grâce à des circuits spécialisés. Ainsi, aucune valeur autre que 1 ou 0 ne peut être exprimée. On dit de ce système d'opération qu'il est binaire (Bi=deux) car il traite 2 valeurs uniquement. Comme comparaison, rappelons-nous que le système que nous utilisons couramment est le système décimal soit 10 valeurs (0 à 9). Tous les ordinateurs travaillent avec le système binaire et ce, quelque soit leur origine ou manufacturier.

Comme vous vous en doutez sûrement, la programmation bit par bit (programmation binaire) n'est certes pas facile et est extrêmement sujette à l'erreur. Lors des débuts de la programmation, on utilisait ce langage machine. Il fallait alors prendre des semaines voir des mois pour trouver une erreur qui se glissait dans une fonction simple. Il fallait trouver d'autres méthodes. Grâce au langage de 3 et 4<sup>ème</sup> génération, on peut maintenant manipuler de plus grandes quantités d'informations si on le désire, tout en gardant la possibilité de manipuler les bits.

On utilise couramment de nos jours l'octet (byte) comme unité de mesure sur le PC et la plupart des ordinateurs courants. Celui-ci constitue le "mot" et est composé de 8 bits, pour donner une idée de la quantité d'informations qu'un octet peut contenir, disons que 1 caractère demande l'utilisation de 1 octet, c'est beaucoup mieux qu'auparavant mais pas pour l'emmagasinement permanent de grande envergure, c'est plutôt réduit. Aussi peut-t-on utiliser diverses méthodes qui combinent plusieurs octets sous divers formats tel que le champ, composé de 1 ou plusieurs octets. Si vous désirez combiner ensemble plusieurs champs, vous obtenez un enregistrement. La combinaison de 1 ou plusieurs enregistrements permettent la création d'un fichier (de données) et plusieurs fichiers combinés forment une base de données. Les capacités d'emmagasinement sont donc maintenant adéquates pour traiter pratiquement toute information de manière relativement simple et efficace que ce soit sur un support rigide ou en mémoire volatile.

#### NOTE:

Ce ne sont pas tous les ordinateurs qui fonctionnent avec des mots de cette dimension. Certains peuvent en utiliser de plus gros d'autres de plus petits. Certains grands ordinateurs par exemple utilisent des mots de 32 bit plutôt que 8. Ceux-ci peuvent donc représenter une plus grande quantité d'informations et manipuler celles-ci plus facilement mais cela au prix de beaucoup plus de mémoire.

Pour bien calculer les capacités en informatique, il nous faut d'abord faire une excursion sur les vieux bancs d'écoles et se rappeler les exposants. En effet, les exposants et, en particulier, les puissances de deux sont utilisés très fréquemment en informatique, en particulier lorsqu'il s'agit de l'adressage de la capacité mémoire. Il est important sinon indispensable de bien les connaître.

L'exposant, pour ceux qui auraient oublié, sert à indiquer qu'un nombre doit être multiplié par lui-même X nombre de fois. Ainsi,  $2^2$  pour nos besoins

d'édition donnera comme résultat 4,  $2^3$  donnera comme résultat 8. Ne pas confondre exposant et multiplication ou encore la notation scientifique  $2E3$ . En effet,  $2*3 = 6$  et  $2E3$  donnera 2000, pas du tout le résultat attendu.

Petit détail ou plusieurs semblent faire des erreurs lorsqu'ils calculent avec les exposants. Que croyez-vous que  $2^0$  donnera? 0 ? ... Non, le résultat sera 1 ! En effet, rappelons-nous que ce n'est pas 2 multiplié par 0. Tout nombre ayant comme exposant 0 donnera 1 comme résultat. Que ce soit  $8^0$  ou  $16^0$ , le résultat sera 1 (essayez avec votre calculatrice, vous verrez).

Comme mentionné précédemment, l'adressage de la capacité de mémoire d'un ordinateur se fait en octet (parfois mais plutôt rarement de nos jours en bit). On dira d'un ordinateur qu'il a 640 "kilo-octets" ou d'un disque qu'il peut contenir 1 "Meg" par exemple. Nous avons vu qu'un octet est composé de 8 bits. Mais qu'est-ce qu'un "kilo-octet" ou un "mega-octet"? Combien d'informations cela représente-t-il?

Comme l'ordinateur ne peut que traiter en binaire, la capacité de mémoire sera forcément exprimé en puissance de 2. L'octet, le kilo-octets ou le mega-bytes (byte = octet en anglais), ceux-ci expriment des capacités d'emmagasinement en puissance de 2.

Comme nous avons vu plutôt, l'octet représente 8 bits, soit suffisamment pour écrire un caractère (lettre, caractère de contrôle, caractère graphique, etc). En puissance de deux, on exprimera cette valeur par  $2^3$  bits. Celui-ci (l'octet) est fréquemment utilisé en programmation aussi. Il est bien d'être familier avec celui-ci. Ex.: lorsque vous utilisez la plupart des fonctions de votre D.O.S., les valeurs sont exprimées en octets, bien que la plupart des logiciels utilise l'expression anglaise "byte" plutôt qu'octet.

Le kilo-octets est exprimé par  $2^{10}$  octets soit 1024 octets. Il faut faire attention de ne pas pas confondre le KILO du système décimal et celui du système binaire, en décimal  $10^3$  (1000) représente le "kilo" mais comme nous avons indiqué à plusieurs reprises, l'ordinateur utilise toujours le système binaire pour ses calculs internes. Ainsi un mega-octets sera  $2^{20}$  Octets soit 1024 kilo-octets et non  $10^6$  (1000 000).

Combien de bits en puissance de 2 peuvent être stockés dans une mémoire de 1 Kbits (1 kilo-bits) ...?  $2^{10}$  soit 1024 bits. Dans 8 Kbits d'après vous?  $2^3 * 2^{10}$  soit  $2^{13}$  bits. Si nous avons voulu connaître le nombre d'octets dans une mémoire de 8 Kbits, quel aurait été le résultat? ... Nous aurions dû diviser le résultat précédent par  $2^3$  car, comme nous l'avons vu tout à l'heure, il y a  $2^3$  (8) bits dans un octet. Le résultat aurait alors été  $(2^3 * 2^{10}) / 2^3$  soit  $2^{10}$  octets ou 1024 octets. Nous avons inclus quelques petits exercices à la fin de la section sur le système décimal. Les solutions se trouvent en fin d'article. Essayez de vous familiariser autant que possible avec le système de puissance de 2 et la manipulation de celui-ci.

L'addition binaire.

Puisque l'ordinateur traite en binaire, il semble naturel que nous ayons aussi à l'apprendre, à tout le moins nous familiariser un peu avec celui-ci afin d'avoir une vue générale.

En décimal, lorsque nous additionnons une unité à la valeur maximum autoriser par ce système (soit 9), nous obtenons 10. En fait, ce que nous obtenons est un déplacement vers la gauche, une "retenue" d'une unité "dizaine" et la remise à 0 de l'unité ayant fait le déplacement vers la gauche. De même, en binaire, si on prend une unité et qu'on l'additionne à la valeur maximale autorisée par le système binaire (soit 1), on obtient un déplacement vers la gauche, une "retenue", une "deuzaine" si vous préférez et la remise à zéro de l'unité ayant fait le déplacement vers la gauche.

Exemple;

$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$	$\begin{array}{r} \overline{1} \quad   \\ 1 \quad   \\ + 1 \quad   \text{ Retenue} \\ \hline 10 \quad   \lll\text{- Dire UN ZERO et non DIX} \\ \underline{\quad}   \end{array}$
---	---	--

Autres exemples;

$\begin{array}{r} \overline{1} \quad   \\ 0011 \quad   \\ + 1010 \quad   \text{ Retenue} \\ \hline 1101 \quad   \\ \underline{\quad}   \end{array}$	$\begin{array}{r} \overline{1} \quad   \quad   \\ 1 \quad 1 \quad   \quad   \\ 1010 \quad   \quad   \\ + 1010 \quad   \quad   \text{ Retenue} \\ \hline 10100 \quad   \quad   \\ \quad   \quad   \quad   \\ \underline{\quad}   \quad   \end{array}$	$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \\ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \end{array}$
---	--	---

L'important est de TOUJOURS faire l'addition de la droite vers la gauche. Il faut un peu de temps pour s'habituer mais le calcul lui-même est d'une simplicité enfantine. Les opérations de soustraction, multiplication / division sont quelque peu plus complexes aussi nous laisserons cela de côté pour un autre cours.

■ Le système hexadécimal.

Il s'agit d'un système basé sur 16 chiffres comme son nom l'indique. Les valeurs seront donc respectivement étalées de 0 à 15, de même qu'avec le système binaire et décimal, lorsqu'une unité est ajoutée à la valeur maximum autorisée par ce système (15) il y a retenue et remise à zéro. Comme vous l'avez constaté, il est impossible d'exprimer en un seul caractère les valeurs 10,11,12,13,14 et 15 aussi on utilise des lettres. Voici donc la table des valeurs du système hexadécimal et un exemple d'addition;

$\begin{array}{r} \overline{1} \quad   \\ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F \\ \quad \quad \quad (10) (11) (12) (13) (14) (15) \end{array}$	$\begin{array}{r} \overline{1} \quad   \end{array}$
--	---

F		15B	
+ 1		Retenue	
-----		999	
10		AF4	

Évidemment, il faut un peut plus de temps pour manipuler ce système que le binaire mais avec un peut d'attention vous devriez y arriver sans problèmes. La plus grande erreur se retrouve dans l'habitude que nous avons acquis de porter la retenue lors de l'addition de 9 avec 1 (i.e.  $9+1 = 10$ ), en hexa,  $9 + 1$  font A et non 10.

L'utilité de ce système pour nous vient surtout du fait qu'il nous permet de manipuler des valeurs binaires avec plus de faciliter en particulier les octets. Rappelons-nous que les octets (des groupes de 8 bits) sont exprimés en binaire. Il est cependant plutôt fastidieux de se rappeler les valeurs de chacun des bits. En effet, dire UN UN UN UN ZERO UN ZERO UN + ZERO UN ZERO UN UN UN ZERO n'est pas pratique et invite à l'erreur. C'est là que le système hexadécimal vient à notre secours.

F (15) la valeur maximum du système hexadécimal peut très bien exprimer la valeur maximum d'un groupe de 4 bits binaire si on se rappelle ses tables de 2, 1111 est en binaire l'équivalent de F, en effet  $2^3 + 2^2 + 2^1 + 2^0$  (soit  $8 + 4 + 2 + 1$ ) font 15. Exprimer la valeur binaire 1111 1111 (1 octet) est alors un jeu d'enfant FF! Ce qui est beaucoup plus facile à retenir et prête bien moins à confusion.

#### ■ Le système octal

Une des raisons qui amena la naissance du système octal est que plusieurs trouvaient le système hexadécimal quelque peu difficile à manier. Grâce à 3 nombres on peut quand même exprimer les valeurs d'un octet (un peu plus même). Mais comme on manipule des nombres de 0 à 7 avec lesquels nous sommes beaucoup plus familiers, les erreurs sont moins fréquentes.

Il s'agit d'un système basé sur 8 chiffres comme son nom l'indique. Les valeurs seront donc respectivement étalé de 0 à 7, de même qu'avec le système binaire et décimal, lorsqu'une unité est ajoutée à la valeur maximum autorisée par ce système (7), il y a retenue et remise à zéro. Voici donc la table des valeurs du système octal et un exemple d'addition;

0, 1, 2, 3, 4, 5, 6, 7

		1	
3		4	
+3		+4	
Retenue			
---		---	
6		10	
		<<<<<<-	
		Dire UN ZERO et non DIX.	

Ceci s'exprime bien en binaire sur 3 bit, 111 en binaire équivaut à 7 en Octal en effet  $1^2 + 1^1 + 1^0$  ( $4 + 2 + 1$ ) = 7. Pour exprimer un octet on n'a qu'à utiliser 3 nombres octaux. Exemple, 1111 1111 (1 octet) peut être séparé en groupe de 3 bits soit 011 111 111 et la valeur obtenue serait 377 en octal. En passant, veiller à toujours préciser le système que vous utilisez si ce n'est pas le système décimal. Par exemple;

11	11	11	ou	11	11
2	8			10	16

Donnerait comme équivalent en décimal;

3	9	11	11	17
---	---	----	----	----

Comme vous voyez, l'ajout de la base utilisée est des plus important, si aucune base n'est signalée, la base décimale est celle utilisée par défaut.

#### ■ Le système décimal

Le système décimal nous étant familiers puisque nous l'utilisons constamment je vais plutôt axer cette partie sur la transformation d'un système à l'autre. Vous avez déjà vu comment faire la conversion hexadécimal-binaire et octal-binaire, cette partie vous donnera quelques exercices pour pratiquer cette transformation ainsi que les méthodes propres à chacune pour transférer en décimal.

Tout d'abord la liste complète des nombres de 1 à 16 dans chaque système.

Décimal	Octal	Hexadécimal	Binaire
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111
16	20	10	10000

#### Transformation octal-hexadécimal

Comme nous avons vu plutôt, le système octal et le système hexadécimal

peuvent facilement être exprimés directement sans calcul de transformation en binaire si on se rappelle sa table de 2. Aussi, rien de plus normal que d'utiliser cette qualité pour transformer d'un système à l'autre (octal-hexa). Il suffit de se rappeler qu'il faut toujours diviser le nombre binaire intermédiaire obtenu lors de cette transformation selon des blocs compatibles avec l'unité du système choisi et le faire de DROITE à GAUCHE. Rappelons-nous que le système octal s'exprime sur TROIS bits et le système hexadécimal, sur QUATRE bits.

Prenons un exemple afin de mieux illustrer.

111010101:

en octal	en hexadécimal	
111 010 101	0001 1101 0101	Vous noterez que nous avons
7 2 5	D 5	ajoutez les zéros afin de
	_____	garder des blocs de 4 bits.

Bien que certains n'ajoutent pas les zéros précédents comme nous l'avons fait, nous préconisons cette approche car elle est moins sujette à l'erreur et s'avère bien pratique lorsque l'on traite avec des soustractions et des multiplications binaires comme nous le verrons plus tard. En effet, lorsqu'on travaille sur papier, les nombres ont tendance à se déplacer d'un côté ou de l'autre lorsque mis sur papier, utiliser un nombre fixe de bloc qu'on aligne ensuite lorsque couché sur papier, réduit les chances d'erreur. Il en tient à vous de choisir l'approche que vous préférez. D'une façon ou d'une autre, le résultat demeure le même.

Transformation décimal -> tous

Pour passer du système décimal à autre un système, que ce soit hexa, octal ou binaire, vous n'avez qu'à diviser la valeur par le type de système désiré. Ainsi supposons que vous désiriez transformer la valeur 37 du système décimal vers le système hexadécimal, vous effectuerez le calcul suivant;

$$\begin{array}{r} 37 / 16 = 2 \text{ ---} \\ 32 \quad \quad \quad | \\ \text{---} \quad \quad \quad 25 \\ 5 \text{ ---} \quad \quad \quad | \end{array}$$

Votre résultat final après lecture de droite à gauche serait 25 (base 16).

Autre exemple 640 décimal en base 16 (hexa).

640 / 16 = 40	/ 16 = 2	Comme la valeur est plus grande que la base utilisée (ici base 16) utilisez on re-divise et ce jusqu'à ce que le résultat soit PLUS PETIT que la valeur maximum de la base recherchée (i.e. 16 pour la base 16,
640	32	
----	----	
0	8	

8 pour la base 8.

Votre résultat final 280 (base 16). Il faut toujours lire la valeur de droite à gauche.

Autre exemple, transformez 67 (décimal) en octal;

167	/	8	=	23	/	7	=	3	Ici, les valeurs sont plus grandes que 7. Donc on continue la division jusqu'à ce qu'on arrive avec un nombre inférieur à 8.
161				21					
---				----					
6				2					

La transformation inverse, d'un système vers le décimal est aussi simple. Il suffit de se rappeler la base et utiliser la position du nombre comme exposant.  
Ex: 67 octal en décimal;

1	6	7	=	1*	8^2	+	6*	8^1	+	7*	8^0	soit	1*64	+	6*8	+	7*1	=	119	(décimal)

Un autre exemple mais hexadécimal cette fois;

2	2	4	=	2*	16^2	+	2*	16^1	+	4*	16^0	soit	2*256	+	2*16	+	4	=	548	(décimal)

Vous n'avez qu'à faire de même pour une transformation binaire-décimal mais évidemment cette fois l'exposant sera la puissance de 2 puisqu'il s'agit du système binaire ...

Si vous désirez quelques travaux pratiques vous pouvez faire ceux-ci. Vous trouverez la solution à la fin de cet article.

- 1- 1 MByte représente;
  - a) 2^10 Bytes
  - b) 2^20 bytes
  - c) 10^2 Bytes
  - d) 10^10 Bytes
  
- 2- Le traitement des données à l'intérieur de l'ordinateur est fait en;
  - a) décimal
  - b) hexadécimal
  - c) binaire
  - d) octal

- 3- La plus petite unité d'information manipulée par l'ordinateur est:  
a) le programme b) le bit c) le fichier d) le byte (octet)
- 4- Un octet est formé de;  
a) majuscule seulement b) minuscule seulement c) les deux  
d) un caractère e) aucune de ces réponses
- 5- Effectuer les conversions suivantes;  
En binaire:  
a) 96<sub>10</sub> b) 127<sub>10</sub> c) 129<sub>10</sub>
- 6- En octal:  
a) 80<sub>10</sub> b) 192<sub>10</sub> c) 128<sub>10</sub> d) 63<sub>10</sub>
- 7- En hexadécimal  
a) 88<sub>10</sub> b) 496<sub>10</sub> c) 701<sub>10</sub> d) 1616<sub>10</sub>
- 8- En octal:  
a) 10011000111<sub>2</sub> b) 001111111010<sub>2</sub> c) 111100001010<sub>2</sub> d) A12CF89<sub>16</sub>
- 9- En hexadécimal  
a) 100011110111<sub>2</sub> b) 001001110010<sub>2</sub> c) 111101101010<sub>2</sub> d) 101010101011<sub>2</sub>
- 10- Effectuer les additions:  
a) 10101111<sub>2</sub> + 11110001<sub>2</sub> = X<sub>2</sub> b) 01110111<sub>2</sub> + 11111101<sub>2</sub> = X<sub>2</sub>  
c) 234577<sub>8</sub> + 171711<sub>8</sub> = X<sub>2</sub> d) 60621455<sub>8</sub> + 76556734<sub>8</sub> = X<sub>8</sub>  
e) 981A<sub>16</sub> + 12FE<sub>16</sub> = X<sub>16</sub>

Le mois prochain, nous compléterons le module  
Concepts et éléments du traitement central en plus des sujets suivants:

- 2.7 Le code ASCII
- 2.8 Le code EBCDIC
- 2.9 Le code BCD (Binaire Codé Décimal)
- 2.10 Les éléments de l'unité centrale.
- 2.11 Saisie - unité d'entrée.
- 2.12 Mémoire auxiliaire.

-----  
\*\* Toutes questions SUR LE SUJET TRAITÉ DANS L'ARTICLE COURANT peuvent nous

être adressées, comme je suis seul à faire ces articles il m'est impossible de répondre aux questions qui seront traitées dans les modules suivant, attendez que nous soyons rendus là pour obtenir des explications détaillées. Les questions sur les modules déjà passés seront traitées avec une priorité d'arrivée (i.e. FIFO First In First Out). Si vous avez une question complexe pour laquelle nous ignorons la réponse, elle sera traitée en ordre prioritaire, comme nous avons accès à une large banque d'experts chez Herzing nous vous répondrons sans délai.

#### COMMENT NOUS JOINDRE:

Pour ceux qui auraient des questions ou voudraient des détails supplémentaires sur les articles publiés, veuillez nous les adresser (Yvon Decelles) dans FM-PROGRAMMATION, FM-MATERIEL, FM-C ou les envoyer à Constantin Koutrias, l'éditeur de cette feuille de chou qui me les enverra à son tour. Si vous affichez directement dans FrancoMédia, veuillez S.V.P. respecter le sujet de conférence, i.e. les questions sur le matériel dans FM-MATERIEL, les questions sur les logiciels dans FM-LOGICIELS.

Note de l'éditeur: Faites-nous le parvenir sous forme de courrier du lecteur et on le publiera le mois prochain.

On peut aussi nous joindre sur FIDONET: 1:167/123, MTL-NET: 17:514/420 ou sur internet: yvon.decelles@odn.illusion.tpg.org

#### CONTENUS TRAITES DURANT L'ANNEE:

#### SIGLES

Mathématiques Appliquées  
 Statistiques et probabilités  
 Communications et interrelations  
 Administration  
 Programmation I  
 Fonctionnement interne des ordinateurs  
 Structure de données  
 Systèmes d'exploitation  
 Technique de développement de systèmes  
 Base de données  
 Développement d'applications en COBOL  
 Développement et mise en oeuvre I  
 Développement et mise en oeuvre II  
 Langage de gestion  
 Approfondissement d'un langage de gestion  
 Développement de systèmes à l'aide d'outils informatisés (CASE)  
 Eléments de micro-informatique.....T  
 Approfondissement de système d'exploitation  
 Approfondissement des télécommunications  
 Anglais de l'informatique

T=Article courant, F=Déjà vu.

-----  
 ELEMENTS D'INFORMATIQUE (introduction)







\*\*\*\*\* Catalogue des logiciels fournis par le RAPQ \*\*\*\*\*  
 \*\*\*\*\* sur le réseau FrancoMédia \*\*\*\*\*

Voici les fichiers distribués dans les bases du Regroupement d'Aide à la Programmation du Québec via FrancoMédia. Vous trouverez ci-joint les quatre bases présentement ouvertes. D'autres s'ajouteront bien sûr selon les besoins.

Tous ces fichiers sont du domaine publique ou sont distribués sous forme de PARTAGICIEL.

Si vous êtes opérateurs de système de babillard électronique et que vous voulez recevoir ces bases de fichiers, il vous suffit de devenir membre de FrancoMédia, et sur votre demande, les fichiers vous seront acheminés directement sur votre système.

Pour les autres, les fichiers sont disponibles sur presque tous les babillards membres de FrancoMédia près de chez vous. Sinon, ils sont disponible directement sur le babillard du R.A.P.Q.:

R.A.P.Q. BBS  
 (514) 671-4464  
 FrancoMédia: 101:190/508  
 FidoNet: 1:167/506

\*\*\*\*\* FM-RAPQCOMM \*\*\*\*\*  
 Cette base contient tous les fichiers ayant rapport à la télécommunication. Exemple: les programmes de babillards, les jeux en ligne, les utilitaires de babillards, les lecteurs hors-ligne et les programmes de communication.

BLOKD101.ARJ <IBM> Block Door 1.01 - Porte genre Tetris  
 FAX100F.ARJ <IBM> FAX v1.0 - Generateur de "news" pour les BBS  
 INFQWK10.ARJ <IBM>  QWK v1.00  
 The ULTIMATE ANSI/ASCII Generator  
 for your QWK Packets!! Use With  
 --- RemoteAccess 1.11 ONLY! ---  
 LHL\_V2.ARH <IBM> LHL v2.00 - Lecteur Hors-Ligne en Francais pour les QWK  
 et BlueWave  
 MAX2TRM2.ARJ <IBM> Transforme la liste MAXBBS en format:Telix, TM, Qmodem  
 et ProComm+  
 MMUSRT1D.EXE <IBM> MMUSORT v1.0d - User sort pour YaleBBS

PSEI120.ARJ <IBM> ProTerm Special Emulator for IBM v1.20  
QCALL100.ARJ Quick Callers Log v1.0 - Liste des appels pour QuickBBS  
QFILS100.ARJ <IBM> Quick File Sort v1.0 - Pour faire un sort des FILES.BBS  
QGIFT100.ARJ <IBM> Quick Gif Trim v1.0, ecrit le format de gif dans les  
FILES.BBS  
ROBO1\_08.EXE <IBM> RoboBoard 1.08 - Babillard graphique.  
ROBO270.EXE <IBM> RoboTerm 2.70 - Terminal pour acceder les  
TW100.ARJ <IBM> TossWise v1.00 - Tosser les messages inteligenment  
(Hudson)  
YECHO112.ARJ <IBM> Yale Echo v1.12 - Tosser pour les YaleBBS

```
***** FM-RAPQDOCS *****  
Cette base est pour la distribution des fichiers d'information  
sur le R.A.P.Q. Aucun autre fichier ne devrait s'y trouver.
```

RAPQ220.EXE <IBM> Documentations et applications pour le RAPQ.  
RAPQ220.ZIP <IBM> Doc et applications format 8bits IBM.

```
***** FM-RAPQJEUX *****  
Cette base est dédiée aux jeux et à tout ce qui s'y rattache.
```

DBLOK123.ARJ <IBM> Double-Block ver 1.23, Jeu genre Tetris pour deux...  
GOBE\_NE.ARJ <IBM> Gobe-Dollar\$  
GOBMAN.ARJ <IBM> Jeu genre PacMan  
MASTERM1.ARJ <IBM> Master Mind  
QUEBECUN.ARJ <IBM> Jeu questionnaire éducatif sur le Québec.

```
***** FM-RAPQUTIL *****  
Cette base sert à la distribution d'utilitaires de toutes sortes  
sauf les utilitaires pour babillard.
```

ABD110.ARJ <IBM> ArcByDate v1.10 - Archive un fichier avec la date  
CARTES.ARJ <IBM> Pour gerer vos cartes de sports.  
DELANY10.ARJ <IBM> Delete Anywhere v1.00 - Efface fichiers n'importe ou  
DTF100.ARJ <IBM> Date/Time Format v1.0 - Partir programme par date et  
heure  
EC.ARJ <IBM> Extracteur de texte pour fichiers: ASCII, COM, EXE,  
etc ...  
EPI-103.ARJ <IBM> Economiseur de Papier d'Imprimante v1.03  
FIL\_1B\_T.ARJ <IBM> Filtreur de LOG pour Telemate 3.xx  
FM\_100.ARJ <IBM> File Maintenance v1.0 - Genre PCFiles  
GLOG100.ARJ <IBM> GeniLog v1.00 - Change le nom d'un log pour la date  
PAINTP10.ARJ <IBM> PaintPro v1.00 - Programme de dessin graphique

READ102.ARJ <IBM> READ 1.02 - Lecteur (Scroller) pour texte  
Z\_101.ARJ <IBM> Z v1.01 - Programme d'environnement pour ZIP, PAK et  
ARJ

-----

Plusieurs signes ont été utilisés dans ces listes de fichiers. En  
voici leur signification.

[AMI] = Fichier pour les ordinateurs de type AMIGA.  
[APL] = Fichier pour les ordinateurs de type APPLE.  
[ST] = Fichier pour les ordinateurs de type ATARI ST.  
[C=] = Fichier pour les ordinateurs Commodore 64 et 128.  
[CC] = Fichier pour les ordinateurs de type Coco.  
[IBM] = Fichier pour les ordinateurs de type IBM PC, XT, AT,  
386, 486, et Pentium.  
[MAC] = Fichier pour les ordinateurs de type MacIntoch.

-----

Le babillard électronique du MMQ

**SSE**

**INFO**

**SEFAULICES**

" L'idéal télématique "

- La source messagerie en réseaux la plus complète et  
diversifiée. Requêtes spéciales possibles!

- La source de plus de 20000 fichiers en réseaux/local et sur CD-ROM!
- Le choix des meilleurs portes de jeux sur le marché
- Service de support vocal, 24 heures par jour
- Service de support par télécopieur, 24 heures par jour
- Centre d'édition pour le MicroMensuel du Québec Upload des articles possibles ici-même!
- Deux lignes hautes-vitesses pour mieux vous servir.
- Coût d'accès très compétitif et très raisonnable.
- Service de point (sous technologie mailer) disponible.
- Vaste choix de musique .MOD et de programmation.
- Babillard commandité par Info Services S.C. qui vous offre des prix spéciaux pour les membres!

Site d'édition/montage du MMQ  
Site de distribution de la door Metal Knights  
Site de distribution du module BGFAX

Roulant sur un des logiciels le plus flexible sur le marché:  
RemoteAccess 2.00+

Support vocal 24h: (514)495-2610 demandez Constantin Koutrias  
Télécopieur: (514)278-1963 disponible 24h par jour  
Ligne d'accès n.1: (514)278-7832 1200 à 14400 bauds/v.32bis  
Ligne d'accès n.2: (514)278-9685 1200 à 14400 bauds/v.32bis

ALORS PREPAREZ VOS MODEMS !!!!  
\*\*\*\*\* CA VA CHAUFFER! \*\*\*\*\*

Le Courrier du lecteur

Cette nouvelle section est ouverte pour toute personne qui désire écrire en réponse à un article publié dans le MMQ, aussi du même coup il peut donner son opinion qui peut aider d'autres gens et ainsi de suite. Faites parvenir vos articles sur mon babillard dont les coordonnées sont publiées au début et à la fin du MMQ.

Du même coup, j'aimerais vous avertir que la conférence FM-REDACTION sur le réseau FrancoMédia est bien disponible

pour qu'on s'échange des idées d'articles, des suggestions pour le MMQ, etc. Il est à noter, par contre, que VOUS NE DEVEZ EN AUCUN TEMPS PUBLIER vos articles destinés au MMQ dans cette conférence, sinon elle sera lue.

Merci de votre attention!

Constantin Koutrias

-\*\*\*> CLARION <\*\*\*-

- Beaucoup de questions me sont posées ces derniers temps sur la nouvelle conférence CLARION en Francomédia. Pas encore très active et surtout méconnue (à mon grand regret), je vais tenter de vous expliquer ce qu'est ce mystérieux "Clarion".

CLARION est un langage de programmation comme vous connaissez certainement DBase, Paradoxe, Clipper et bien d'autres, orienté gestion de bases de données. C'est aussi un générateur d'applications très simple à utiliser pour le commun des mortels. En clair, vous pouvez créer avec CLARION un agenda personnalisé, une gestion de votre stock, une gestion de comptes bancaires, ou encore créer un logiciel pour gérer votre bibliothèque de K7 vidéo, votre cave à vins, vos recettes de cuisine, votre courrier à des fins de mailing... etc. Ses possibilités sont infinies... mais le plus important, ce logiciel s'adresse à tous, professionnels (car CLARION a son propre langage, très puissant) ou néophytes d'un jour. Bien sûr, il y a beaucoup d'outils à greffer autour de ce langage. Mais le plus important, c'est que la version de base vous permet de créer vos écrans en les dessinant, créer vos tables et vos formulaires en un tour de bras, et d'avoir au total une application compilée et exécutable qui peut tourner sur n'importe quel PC mais aussi en réseau et ce, sans avoir de connaissances particulières en programmation! Pour aller plus loin, on peut y greffer des routines écrites en C, utiliser le langage SQL, travailler sur des bases de données au format DBase, les index, même perdus, seront reconstruits automatiquement! Essayez de faire la même chose avec DBase! Notre cher éditeur à même adopté ce langage... c'est pas une référence ça! -)))

Yves MARTIN.  
Modérateur FM-Clarion.

— Annonces classés —

Les intéressés à y placer une annonce quelconque devront me contacter sur mon babillard, SBE Info Services - (514)278-7832 ou en NetMail au 1:167/223 et 101:190/102 pour discuter du tarif et de l'annonce. Les annonces de babillards sont bienvenues. Tous les profits provenant de ces annonces servent, dans un but non-

lucratif, à couvrir les diverses dépenses que le MMQ entraînent pour son éditeur.

AVIS AUX COMMERCANTS - ANNONCEZ, AUX TRAVERS LE QUÉBEC A PEU DE FRAIS.

TARIFS EN COURS POUR L'ÉTÉ 1993:

Annonce particulière (usagé) .....	2\$/mois
Annonce de babillard (non-lucratif) .....	4\$/mois
Annonce de babillard (lucratif) .....	8\$/mois
Annonce commerciale (non-lucratif) .....	5\$/mois
Annonce commerciale (lucratif) .....	10\$/mois
Autre type d'annonce .....	APPELEZ

La ligne voix du MMQ, disponible 24h/24h tous les jours de la semaine.  
Constantin au (514)495-2610 voix (répondeur si indisponible)

La ligne data du MMQ, disponible 24h/24h tous les jours semaine.  
SBE Info Services (514)278-7832/9685 data (Babillard électronique)

La ligne FAX du MMQ au (514)278-1963, disponible 24h/24h.

— Info Services S.C. —

Vente et réparation d'ordinateurs IBM et Compatibles  
Service de consultation et autres

Appelez-nous, nous offrons un vaste choix de produits  
adaptés à vos besoins. Consultez nos experts conseils

Constantin au (514)495-2610 ou Simon au (514)259-9353 voix

Laissez un message si nous sommes indisponibles  
nous vous rappellerons GARANTI.

Constantin Koutrias - MMQPub

Le mois prochain dans le MMQ...

- La suite possible des articles de TurboPascal de Jean Gravel.
- L'informatique selon le professeur (3e partie)...
- Les Infos du mois
- Sondage MMQ #1 - Reporté!
- CLARION, un logiciel de développement pour tous.
- Et plus...