CN0101103

REPUBLIQUE DU SENEGAL

MINISTERE DE LA RECHERCHE SCIENT. ET TECHN.
INSTITUT SENEGALAIS DE RECHERCHES AGRICOLES
DEPARTEMENT SYSTEMES
CNRA / BAMBEY

MINISTERE DE L'EQUIPEMENT
DIRECTION DE LA METEO NATIONALE
DIVISION RECHERCHE ET DEVELOPPEMENT
DAKAR

RAPPORT DE FIN DE STAGE

AU CENTRE DE CALCUL

"DE LA METEO. NATIONALE"

PRESENTE PAR : NDONGO **NGOM**
TECHNICIEN METEO.
ISRA - CNRA / BAMBEY

MAITRE DE STAGE
OMAR **LO**
DIRECTION METEO NATIONALE

D E D I C A C E

Ce rapport est dédié à :

"TERRE DES HOMMES - FRANCE*/ SENEGAL".

Et à ma famille : pour son soutien moral constant,

Que tous mes collaborateurs ISRA/CNRA/BAMBEY y trouvent leur joie, et leur satisfaction,

* Terre des Hommes/France (THF) --> ONG (Organisme non Gouvernemental d'appui au Développement).

REMERCIEMENTS

- Nos remerciements et nos gratitudes à MM les Directeurs de la Météorologie Nationale et du CNRA de Bambey.

- Nous remercions également tout le personnel du Centre de Calcul de la Direction de la Météorologie Nationale, en particulier :

. le Chef de Division : Mr, Sory DIALLO
. le Chef de Bureau par intérim : Mr. Alpha K. TRAORE
. les Opérateurs - systèmes MM. O.G - B.K et O.B.
. principalement à Mr Omar LO qui fût mon encadreur pendant le stage.

Enfin, nous remercions l'ensemble du personnel de la Direction de la Météorologie Nationale par leur collaboration et leur ouverture d'esprit,

Nous souhaitons de tout coeur que ces genres de collaboration entre deux Directions de Tutelles différentes se multiplient en un échange d'expérience au grand profit de la nation SENEGALAISE sur le plan météorologique et agrométéorologique,

Que chacun veuille trouver ici notre profonde gratitude et nos sincères remerciements.-

S O M M A I R E

I - COURS THEORIQUES FONDAMENTAUX

1/- Introduction

, L'informatique est une discipline scientifique, s'il est vrai qu'elle s'est développée grâce au cours des mathématiciens et des physiciens, elle se veut aujourd'hui une science autonome et un outil à la disposition de l'homme dans ses multiples activités de recherches scientifiques ou autres,

, Mais il faut bien admettre que l'ordinateur est une machine, rien de plus et nous savons que les machines sont dépourvues d'intelligence .

Donc à l'homme de décomposer les problèmes les plus complexes en une suite d'opération de base.

, L'ordinateur est un outil idéal pour l'homme, car il lui apporte ce qui :Lui fait défaut c'est-à-dire la rapidité et la sureté d'exécution ; Citons à ce propos, J.J. SERVAN SCHREIBER qui disait : "L'ordinateur n'est pas une forme de raison", au contraire, il n'est qu'un produit de la raison, il n'est qu'un instrument pour une meilleure application de la raison, Un ordinateur ne peut pas se substituer au jugement, pas plus qu'un stylo ne peut se substituer au talent.

"L'homme pense, l'ordinateur exécute".

, Théoriquement, l'utilisateur n'a pas besoin dos notions qui vont suivre. On peut très bien analyser et programmer en les admettant comme des principes, De même rares sont les personnes qui ont besoin de connaître les principes de l'algèbre de Boole ou le fonctionnement des circuits,

, La programmation dépend essentiellement d'une organisation correcte des travaux à effectuer, Mais au fond des choses tout est interaction entre éléments constitutifs d'un ordinateur, C'est ce "pourquoi" qu'il est satisfaisant de comprendre : les leçons suivantes en particulier deviendront alors beaucoup plus claires,

. Durant le stage nous avons étudié comme langage de travail le BASIC* beaucoup plus facile et accessible à t

Le BASIC est un langage de programmation dont les instructions ressemblent à des formules mathématiques auxquelles s'ajoutent dos mots anglais - (Lot - Go - to - Read - Print - If - Then - End....etc)

En raison de sa parenté avec l'algèbre élémentaire ; BASIC se prête bien à la résolution do problèmes mathématiques statistiques ot aux calculs de bureaux d'études. BASIC résoud des problèmes courant de gestion d'économie de psychologie, de médecine, de techniques documentaires, et virtuellement tous domaines qui demandent la manipulation de données numériques ou alpha-numériques (c'est à-dire lettres - mots et symboles).

Structure d'un programmo BASIC

Les régles qui s'appliquent à toutes les instructions de BASIC sont :

1 - chaque instruction doit apparaître sur une ligne séparée : 1 ligne = 72 ou 80 caractéros (132 caractères possibles)

2 - uno instruction no peut dépasser la longueur d'une ligne

3 - Chaque instruction doit commencer par un nombre entier positif appelé numéro de ligne ou d'instruction : deux instructions ne peuvent avoir la même numéro

4 - les numéros d'instruction successives doivent aller au croisant

5 - Chaque numéro d'instruction doit être suivi d'un mot clé qui indique le type d'instruction à effectuer

6 - Des blancs peuvent insérer partout afin d'améliorer la lisibilité du programme

Exécution d'un proqrammo BASIC

Une fois écrit, un programme peut être exécuté sur les ordinateurs de toutes marques et de toutes garnmes ou presque : à part quelques différences minourcs entre certaines versions BASIC : le langage est indépendant do la machine - ce point a largemont contribué à la diffusion do BASIC. L'exécution d'un programme BASIC est une opération à 2 passes,

- Le 1er est appelé compilation ; c'est la traduction des instructions de BASIC on instruction de langage machine propre propre à l'ordinateur employé. Chaque ordinateur a son propre langage machine - Le transfert du programme écrit au langage machine est impossible entre deux ordinateurs différents.

- Le programme BASIC ost appelé programme source

- Le programme en langage machine est appelé programme objet et est exécuté provoquant le traitement des données introduites selon les instructions données,

- Le programme objet est exécuté immédiatement après compilation,
- Le programme Source : les données sont entrées et les résultats imprimés sont seuls visibles,

QUELQUES AVANTAGES DU BASIC : OU BIEN LES CARACTERISTIQUES

(1) - le BASIC est simple à apprendre et agréable à utilise:
(2) - le langage est simple et la modification du programme est possible et sans difficulté
(3) - le BASIC est adapte aux systèmes de temps partagé
(4) - les différences sont mineures d'une version do BASIC à l'autre, Le langage est indépendant de la machine,

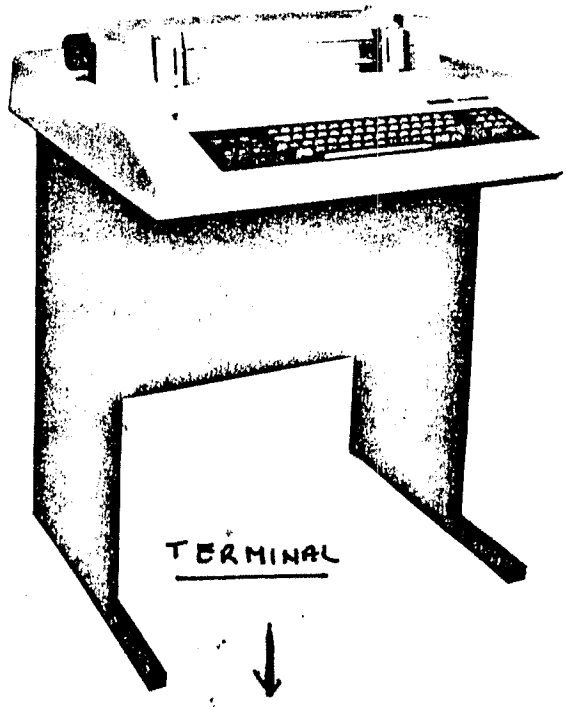
NB : Les cours théoriques comme pratique ont pour but l'acquisition d'une connaissance de la programmation nécessaire à la saisie des données. La salle informatique de la D.M.N/* est équipée d'ordinateur de marque DEG - système 10 do Digital Equipement CORPORATION avec comme composants·

- 1Magtape ou (Déroulour de bande magnétique)
- 1 terminal LA 36 ou (console imprimante) avec 0 comme N° logique
- 1 Terminal DiabloXEROX T740 ou (imprimante) on-line/off-line avec 3 comme N logique
- 1 VT 55(Vidéo Terminal) avec écran de visualisation avec 2 comme N° logique.

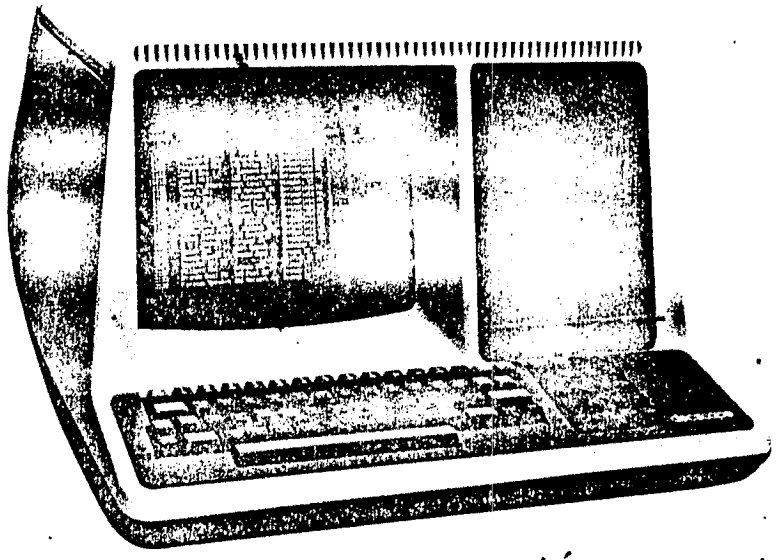
Le terminal est du terminal C R T (Cathode - Ray - Tube) et on fin 1 X E 0 2 (Lecteur de diskotte) souple,

---

* BASIC : Beginners All Purposo Symbolic Instructions Codes
* DMN : Direction de la Météorologie Nationale,



LA36



VT52

Figure l-3 Terminal Devices

Generally, an RT-il computer system has only one terminal' through. which all system/user interaction takes place. This is , called the console terminal. If the system has more than one terminal, one of them is still designated the **console terminal; others simply provide auxiliary message-printing capabilities.**

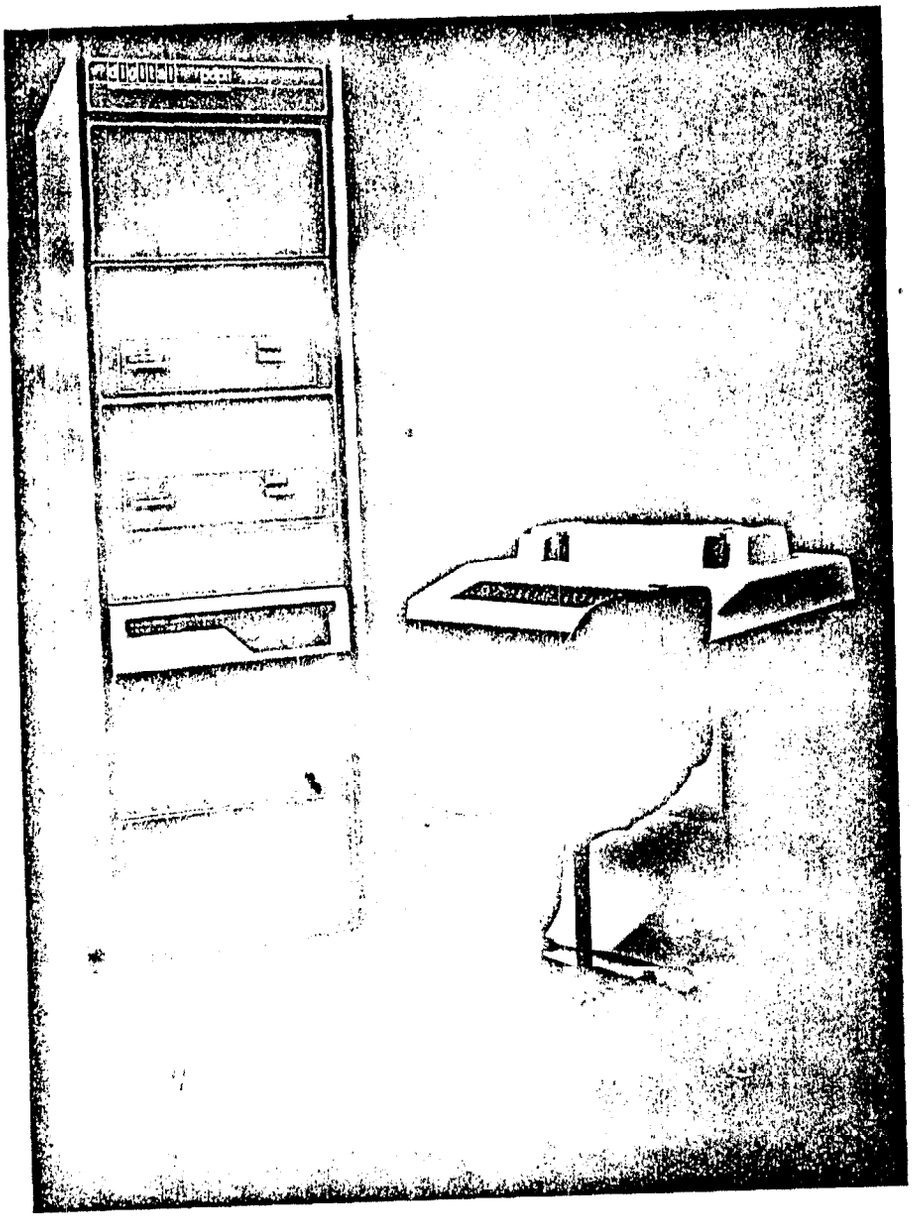
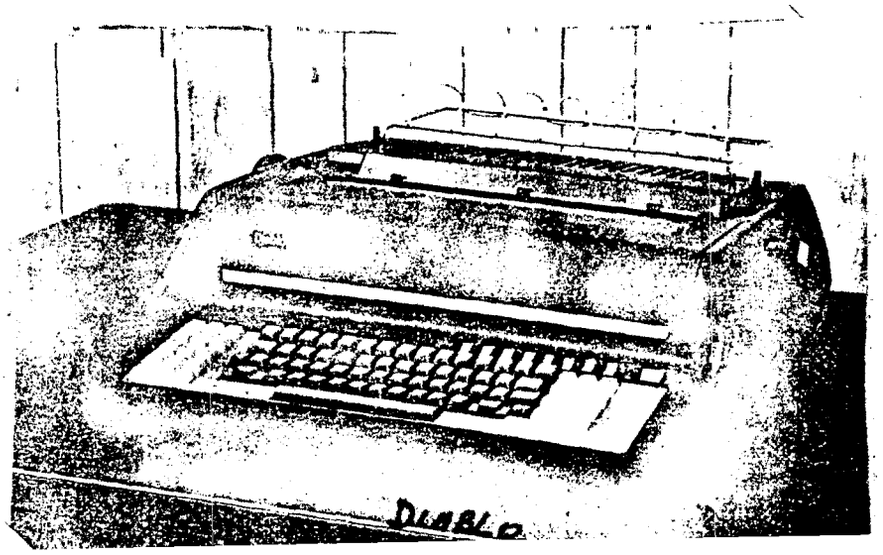


Figure l-l RT-11 Computer System



Les premiers pas en BASIC

"Etude de quelques concepts fondamentaux du langage BASIC : comme les nombres, les variables et formules, les instructions les plus courantes qui servent aux entrées/sorties des données ; aux traitemonts et pormcttont de sauter à un autre point do programme quand on le désire,,

(ou constantos) : sont appelés quantites numériques on BASIC ; ils sont exprimes on entiers ou en décimaux.

NB : (1) Pas de virgule (,) dans un nombre : mais un point (.)

(2) Un nombre admet un exposant : base 10 symbolysé par E.

Exemple : 1.2×10^{-3} s'écrira en BASIC 1.2E-3

L'exposant peut être positif ou négatif, mais pas de point décimal.

(3) Le signe (+ ou) peut précéder un nombre,

(4) les versions BASIC acceptent dos nombres de 8 ou 9 chiffres significatifs

(5) Les nombres sont compris entre 10^{-38} le plus petit et 10^{38} le plus grand,

- Zéro est permis, Ces limites varient d'une version BASIC à l'autre.

2/- Les chaînes :

Une chaîne do caractércs ou chaîne est une suite de caractères, c'est-à-diro des lettres, chiffres, caractères spéciaux comme +, -, /, *, =, $.....etc

Le nombre de caractères pour une chaîne varie selon les versions BASIC de 15 à 4095 C sur certains systèmes. On se sort de chaînes pour représenter des informations non numériques comme des adresses, dos noms.....etc, mais aussi pour commenter des résultats imrpimés, ou' imprimer des messages ainsi qu'il sera vu plus loin,

Exemple : papa Noël y $19.95, Peugeot 504 D

3/- Les variables :

Une variable est un nom qui représente un nombre ou une chaîne. Une variable numérique doit être composée d'une lettre, éventuellement suivie d'un entier. Une variable do chaîne est écrite

Exemple : A, K, X, C1, X5 (1 variable = 1 quantité numérique)

A$, K$, X$, C$, T$(1 variable = une chaine)

LES INSTRUCTIONS D'ENTREE ET DL SORTIE

a) L'instruction INPUT

L'instruction INPUT sert à introduire les données(numériques ou caractères) au cours de l'exécution d'un programme : instruction N° de ligne + INPUT t des variables.

Exemple : 5 INPUT A, B, C
10 INPUT N$, M$, XO, F5
15 INPUT P(I), Q (I), T$ (1)= variables indicées

- L'instruction INPUT est utile paur les programmes conversationnels.

Les régles paur l'i:troductiondes données :

(1) - Les données introduitos doivent correspondre en nombre et on typas aux variables figurant dans INPUT

(2) - Les données doivent être séparées par des virgules

(3) - Elles doivent êtro dos nombres ou des chaines. Les formules rie sont pas autoriséos,

(4) - Les chaines avec virgules, et commonçont par des blancs doivent être "* * * * "les guillemets sont optionnels pour des autres chaînes.

Exemple : X, y, et C9 = 5,-1.2 X 10-3,et 7 Déc 1947,
La ligne d'instruction des données sera ? 5, - 1.2E - 3, "7 DEC 1947"et après on frappe un retour chariot, provoquant la transmission dos données à 1' ordinateur, L'instruction INPUT est utile pour les programmes élémentaires avec peu do données,

B/- IMPRESSION DE RE:SULTATS ET TEXTES AVEC : L'INSTRUCTION PRINT

L'instruction PRINT sert à transmettre des données numériques au terminal ou à imprimer dos caractères quelconques (caractères ASCII) qu'il faut toujours (délimiter par des " ". Elle comporte un n° de ligne le mot clé PRINT, et une listo d'éléments en sortie qui peuvent être des nombres, dos formules, ou des chaines : les points virgules ou les virgules séparent les éléments successifs,

Exemple : '100 PAINT A, B, C
110 PRINT " X = " ; X , " y = " ; y
120 PRINT " NOM ; " ; N $, "ADRESSE : " A $
1 30 PRINT

Les variables C $ (K), U (I) et V (1) sont indicées

Si la variable stockée en mémoire à une valeur numérique quelconque, mais dépassant 999.999, la machine J:'imprime sur le terminol en mettant un SP devant et un SP derrière,

Exemple : W7 = 23.55 (en mémoire)

(1) : PRINT W7 donnera

423.55 Δ

(2) : PRINT 53 donnera

Δ53Δ

. Si la valeur dépasse 6 chiffres, la machine l'imprime sous la forme (exponentielle) en mettant toujours un SP devant, et un Sp derrière

. Si le contenu de la variable est une chaîne de caractères.

Exemple : A9 $ = "Il est midi"

Alors la machine l'imprime intégralement sur le terminal sans mettre un Sp ni devant ni derrière.

C/- L'INSTRUCTION END

L'instruction END marque la fin d'un programme BASIC, elle comporte un N° d'instruction suivi du mot END. Ce doit être la dernière instruction d'un programme : Elle doit avoir le N° de ligne le plus élevé .

III - QUELQUES INSTRUCTIONS ELEMENTAIRES

a) L'instruction R E M

La manière la plus simple d'introduire des commentaires dans un programme est d'employer l'instruction REM (Remarque). Elle comporte un N° de ligne, le mot clé REM et un message. On peut insérer une instruction REM a tout endroit d'un programme.

Exemple : 15 REM ce programme sert à calculer les racines d'une équation,

b) Transfert de contrôle : l'instruction GO TO

- L'instruction GO TO sert dans un même programme à faire un branchement inconditionnel sur une autre instruction, Donc on peut l'utiliser pour modifier l'exécution normale d'un programme.

Exemple :
```
5 R E M GO TO
10 surface du rectangle
20 PRINT "longueur - largeur"
30 INPUT L1, L2
40 s = L1 * L2
50 PRINT "la surface" ; S
60 PRINT
70 GO TO 10
80 E N D
```

- L'instruction GO TO sert à faire des programmes répétitifs ; la machine exécute un programme per N° de ligne croissant qui va de 1 à 32 767.

- L'instruction GO TO doit être l'avant dernière instruction avant END.

* On peut altérer l'ordre normale d'un programme.

- Les branchements inconditionnels (Par exemple GO TO) servent à altérer l'ordre normale d'un programme.

- L'instruction GO TO permet de transférer le contrôle à n'importe qu'elle autre instruction d'un programme y compris l'instruction REM.

c) L'instruction IF..... THEN.....

- Contrairement aux instructions GO TO, les instructions IF..... THEN..... exécutent un branchement conditionnel selon qu' une condition est vraie ou fausse. Il existe doux types d'instruction conditionnelles IF..... THEN.....

a) IF (cond) THEN (n° do ligne), Cet instruction sert à tester une condition donnée et exécute un branchement sur une instruction dont le N° de ligne est spécifié.

Exemple (1) IF A = 1 THEN 300
(2) IF B$=/ "NGOM" THEN 315

b) IF (cond) THEN instruction

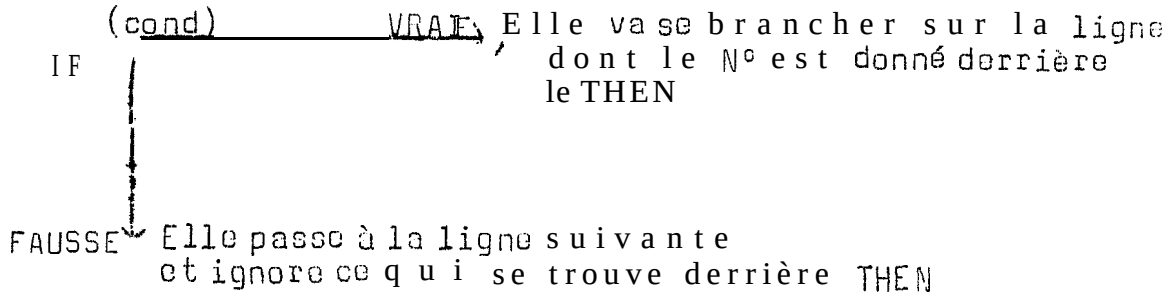
Elle permet d'exécuter une instruction si une certaine condition est vraie,

Exemple : IF$ < 7 0 THEN PRINT "TOTAL" = ; $

Si la condition testée par le IF est fausse, l'exécution du programe continue sur la ligne suivûntc, et l'instruction qui suit le THEN est purement et simplement ignorée.

Fonctionnement : IF (cond) THEN N° de ligne

On peut schématiser son fonctionnomcnt par 2 flèches perpendiculaires



d) L'instruction STOP : Elle sert à mettre fin à l'exécution on n'importe quel point du programme, Elle remplace un GO TO qui renverrait à l'instruction END.

* La différence entre Stop et END.

- STOP : peut apparaître à n'importe quel point du programme, sauf en toute dernière place, et il peut y avoir plus d'un stop

- END : Ne peut se trouver strictement qu'une seule fois en dernière instruction uniquement,

A - PROGRAMME REPETITIF BRANCHEMENTS ET BOUCLES

La boucle consiste à répéter une portion de programme un nombre déterminé de fois, ou jusqu'à ce qu'une certaine condition soit satisfaite. La partie du programme qui est répétée peut elle même contenir un branchement conditionnel qui. sort à déterminer si la boucle est terminée, ou si elle doit être effectuée encore au moins une fois, Si elle doit être recommencée, le contrôle ost transféré à la 1ère instruction de la boucle et les instructions de cotte dernière n'ont pas besoin d'être dupliquées,

PAS D'UNE BOUCLE

On peut ajouter une valeur fixe au compteur d'une boucle (valeur négative positive) pour modifier le nombre de répétitions. Cette valeur fixe ajoutée au compteur avant chaque répétition est appelée le pas du compteur, Si cette valeur n'est pas donnée, la machine assure qu'elle est égale à + 1.

B - L'INSTRUCTION FOR TO : CONSTRUCTION D'UNE BOUCLE

- L'instruction FOR TO indique combien de fois la boucle doit être effectuée ; ça doit être obligatoirement la 1ère instruction de la boucle,

FIN DE BOUCLE : L'INSTRUCTION NEXT

- Une boucle commence par une instruction FOR TO, et se termine par une instruction NEXT, La boucle est donc comprise entre ces deux instructions,

Exemple :

```
50 FOR I = 1 TO 10
60   -"-     -"-
70   -"-     -"-
80 NEXT I
```

la boucle sera exécutée 10 fois.

C - BOUCLES IMBRIQUEES

Une boucle peut être imbriquée dans une autre si besoins est ; il peut même y avoir plusieurs niveaux d'imbrication,

2 - Une boucle imbriquée dans une autre, ne peut avoir la même variable de contrôle ,

3 - une boucle intérieure (imbriquée) doit être entièrement contenue dans la boucle extérieure (c'est-à-dire que 2 boucles ne peuvent se chevaucher),

4 - On peut effectuer un saut depuis une boucle intérieure vers une instruction d'une boucle extérieure, ou même située en dehors de toutes les boucles, mais on ne peut pas effectuer un saut de l'ext vers l'int

D/- LES INSTRUCTIONS FOR.....NEXT

La paire d'instruction FOR.....NEXT permet do simplifier l'écriture des programmes répétitifs,

On peut résumer son fonctionnement en disant qu'elle asscie un ampleur à un branchement inconditionnel.

- Le compteur peut être n'importe quelle variable BASIC à laquelle on assigne une valeur initiale et une valeur finale,

- Le branchomcnt inconditionnel est symbolysé ici par l'instruction NEXT. Il permet de dira à la machine de recommencer le même travail.

Exemple :
```
10  TV = 0
20  FOR 1 = 1 TO 100
30  TV = T9 + 1
40  NEXT 1
```

Remarques sur les instructions FOR.....NEXT

1°/- On appelle Rang d'une instruction FOR.....NEXT....., le nombre de lignes qui se trouvent entre l'instruction FOR et l'instruction NEXT.

STRUCTURE DES INSTRUCTIONS FOR.....NEXT

- Lorsque la machine exécute toutes les instructions comprises entre l'instruction FOR et l'instruction NEXT, elle exécute immédiatement l'instruction qui uuit lc NEXT, on dit qu'elle fait une sortie normale de la bouclo.

- Nais lorsqu'un branchement inconditionnel intervient avant l' instruction NEXT, l'exécution du programme peut continuer en dehors de la bouclo y dans ce cas, on dit qu'il y a un saut, Lo saut peut être considéré alors comme une sortie anormale de la boucle.

Exemple :
```
10  DIM D$ (50)
20  FOR 1 = 1 TO 50
30  INPUT D $ (1)
40   IF D $(I) = "504" THEN 200
50  PRINT D $ (1)
60  NEXT 1
200 T4 = T4 + val (D $ (I) )
280 E N D
```

- La fonction VAL utilisée dans l'instruction 200 est une fonction BASIC qui permet d'utiliser la valeur d'un nombre mis entre " " o u d'une variable de chaînes dont la valeur est mise entre " ".

L'argument de la fonction val peut être :

- soit un nombre quelconque mis entre " ".

Exemple : 410 X 1 = VAL (" 10.5")

- Lorsque la machine exécute l'instruction 410, la valeur de X1 devient égale 10.5

- soit une variable do chaînes dont la valeur est égale à un nombre mis entre " ".

Exemple : 415 D$ = "142.8"
420 X1 = VAL (D$)

Lorsque la machine exécute CE programma, la valeur de X1 devient égale à 'i42.8.

Remarques : la fonction VAL peut donc être utilisée pour stocker en mémoire une liste dont les éléments sont des nombres réels, et des chaines de caractères mélangés, la fonction servant uniquement à connaître la valeur des nombres mis entre " ".

BOUCLES FOR.....NEXT IMBRIQUEES

Introduction: on peut mettre une boucle FOR.....NEXT à l'intérieur d'une autre boucla FOR.....NEXT. Cependant, cette technique do programmation demando quelques précautions. Les réglas suivantes sont à connaître :

I - REGLE N°1 : On no doit jamais utiliser le même compteur pour les 2 boucles.

EXEMPLES

Mauvais exemple

```
10 FOR K  = 11 TO 30
20 T6     = A1 (K)
30 FOR K  = 1 To 15
50 T7     = AZ (K)
60 NEXT K
70 NEXT K
```

Bon exemple

```
10 FOR K = 11 TO 30
20 T6    = A1 (K)
30 FOR J = 1 To 15
40 T7    = A2 (J)
SO NEXT J
60 NE XT K
```

II - REGLE N° 2

Quand dos boucles FOR.....NEXT sont Imbriquées, la bouclo ayant le plus petit rang doit se trouver complètement à l'intérieur de la boucle FOR.....NEXT, dont le rang est immédiatement supérieur, ut ainsi de suite,

EXEMPLES

Bon exemple

```
FOR 1 = 20 To 30
FOR K = 9 To 22
FOR J = 1 To 15
NEXT J
NEXT K
NEXT 1
```

Mauvais exemple

```
FUR H1 = 1 To 15
FOR J9 = 10 To 20
NEXT H1
NEXT J9
```

III - REGLE N° 3 :

Les régles qui guident l'emploi des "sauts" (Rang étendu d'une boucle) dans une boucle FOR...NEXT simple sont las mêmes que pour les boucles FOR NEXT imbriquées, On peut **sauter** d'une boucle intérieure vers une boucle extérieure vers lc b a s , , Mais o n ne peut pas sauter d'une boucle Extérieure vers une instruction appartenant à une boucle intérieure autre quo la 1ère instruction de la boucle : c'est-à-dire l'instruction FOR.

EXEMPLES

Bon exemple

```
FOR 1 = 10 To 32
FOR J = 1 To 20
GO TO 110
NEXT J
110 T6 = VAL (B$)
130 NEXT I
```

Mauvais exemple

```
FOR I = 10 To 32
GO TO 130
FOR J = 110 TO 120
130 T7 = 8.5* A9 (1,J)
NEXT J
NEXT 1
```

- On peut utiliser autant do boucles imbriquées qu'on veut dans un programme, en respectant les régles définies au paragraphe précédent (saut incrémcnt) - Cependant le niveau d'imbrication des boucles ne pout être supérisur à 18

Exemple

4 niveaux d'imbrication,

Il faut étudier soigneusement la valeur prise par un compteur avant d'exécuter un saut, surtout lorsque la valeur initiale et la valeur finale d'un compteur sont des expressions algébriques.

Exemple :

Ecrire un programme qui calcule le produit de 2 tableaux A1 (6,6) et B1 (6,6). On créera un tableau C1 (6,6) dont chaque élément est le produit de l'élément de A1 et de l'élément de B1 situé sur la même ligne et la même colonne.

SOLUTION

```
 5 REM  Produit de 2 tableaux
10 DIM A1 (6,6), B1 (6,6), C1 (6,6)
15 FOR 1 = 1 TO 6
20 FOR 3 = 1 TO 6
25 READ A1 (I,J), B1 (I,J)
30 C1 (I,J) = A1 (I,J)* .1 (I,J)
35  NE:XT 3
40  NEXT 1
45  DATA A1,B1, A1,B1, A1,B1, A1,B1, A1,B1.....etc
50 DATA A1,B1, A1,B1
60  E N D
```

NB. Ici on fait rentrer un élément de A1, sur lequel suivra un élément de B1.

Remarques : Générales sur les boucles FOR... NEXT imbriquées

I - Les valeurs initiale et finale d'un compteur peuvent être des expressions algébriques ; quelque soit l'expression algébrique utilisée, la valeur initiale doit être un entier positif ou nul. Le BASIC convertit automatiquement les expressions algébriques utilisées en nombres entiers,

- Si la valeur de l'expression algébrique utilisée est un nombre fractionnaire, la machine ignore tout simplement la partie décimale du nombre, (Troncature).

II - On ne doit pas modifier la valeur d'un compteur par des assignations de valeurs à l'intérieur d'une boucle.

III - Les indices des listes et des tableaux (lignes et colonnes) peuvent être des expressions algébriques linéaires de la forme $Y = A X + b$.

LES LISTES

Introduction

En informatique, l'ensemble E est appelé une liste qu'on désignera par E (X).
Par exemple : le 1er élément de la liste sera noté E (1) le 2ème E (2). et le Xe Elément de la liste (le dernier) sera noté E (X) appelé "E de X".

LES DIFFERENTS TYPES DE TABLEAUX

Il y a 2 types possibles de tableaux en BASIC :

- les tableaux dont les Eléments ont des valeurs numériques
- les tableaux dont les Eléments sont des chaînes.

Nom symbolique dos tableaux,
Pour choisir lc nom d'une liste, on obéit aux mêmes réglas quo pour choisir les noms des autres variables,

1°/ Exemple : Si la liste contient dos valeurs numériques, on lui donne 10 nom d'une variable numérique.

2°/ Exemple : Si la liste contient des chaînes, on lui donne, le nom d'une variable de chaînes,

Exemple (1) : X, X1, XC!, W, W9
(2) : X$, X1$. W$. W0$.

A - L'INSTRUCTION DIMENSION ou DIM

Pour permettre 3 la machine de prévoir au niveau do la mémoire un nombre dc cases mémoires suffisants pour y loger tous les éléments d'une liste, on utilise l'instruction DIM qui permet do dimentionner la liste,

Exemple : 25 DIM X1 (20)

Le chiffre ( ) indique le nombre d'éléments de la liste, et la machine s'aura qu'il. y a une liste de 20 éléments ayant tous des valeurs numériques à caser dans la mémoire centrale.

Remarques : 1- le nombre d'éléments d'une liste ne pout dépasser 32 767
2- Il ne faut pas confondre le nom symbolique de la liste qui se trouve dans l'instruction DIM et le Nième Eléments do la liste,

Exemple :
```
10 DIM N8 (50)
20 N8 (50) = N8 (50) + X1
```

3- On peut mettre plusieurs instructions DIM sur une même ligne,

Par exemple : 5 DIM N 8 (50), ii 9 (25),X 4 $ (100)

Cependant, il faut séparer les éléments Par une virgule (délimiteur)

4- Les instructions DIM doivent être obligatoirement placés au début du programme avant toute instructior exécutable - (Sauf REM)

5- Dans un programme BASIC, une liste est dimentionnée une seule fois ;

6- L'Instruction DIM n'est pas une instruction exécutable. Par conséquent, on ne peut Pas effectuer un branchement (CO TO, IF - THEN,... etc) sur une instruction DIM.

B - LES INSTRUCTIONS READ - DATA

Instruction : Contrairement à l'instruction INPUT qui permet d'entrer dos nombres directemont à travers un terminal (temps réel), les instructions READ - DATA. . . permettent d'entrer des nombres dans la mémoire à travers le programme lui même (temps DIFFERE).

1/- L'Instruction READ

Comporte un N° de ligne suivi du mot clef READ et d'une liste d'éléments qui peuvent être soit des variables, soit des éléments d'une liste, dont on aura au préalable défini les dimensions,

Exemple : 120 REAC' A$, CC, K1, K2, F5$.

NB : les éléments de la liste doivent être séparés par dos virgules.

2 /- L ' instruction DATA : comporte un N° de ligne suivi du mot clef DATA et d'une liste de constantes séparées par des virgules,

Exemple : 55 DATA "juillet", 31.8, 22, 25, "BAC"

NC : unu chaîne de caractères placée dans une instruction DATA doit toujours être mise " "

Exemple : 160 DATA "Evaporation", "Température", "vent"

Remarques générales sur les instructions READ - DATA

1 - On peut mettre autant d'instructions READ....DATA dans un programme. Cependant une liste d'éléments ne peut contenir plus du 32.767 éléments.

Il faut se rappeler aussi qu'une chaîne do caractères, ne peut comporter plus de 255 caractères.

II- L'instruction READ (liste) peut se mettre dans une boucle FOR.....NEXT.pour faire une lecture répétitive,

III- L'ordre dans lequel on met l'instruction READ et l'instruction DATA n'est pas important, c'est-à-dire que dans un même programme,
L'instruction DATA peut précéder l'instruction READ et vis versa.

IV - Le nombre de variables dans une instruction READ doit correspondre, Terme à terme aux nombres de constantes dans une instruction DATA - c'est pourquoi il est important de les séparer par une virgule.

LES TABLEAUX A DEUX DIMENSIONS

I - Définition

Un tableau à deux dimensions est une matrice contenant N lignes et p colonnes - Donc il contient N X p Eléments ; chaque Elément du tableau peut être reprérer par deux indices :

- un indice I représentant le N° de sa ligne
- un indice I représentant le N° de sa colonne

Donc un élément quelconque du tableau peut s'écrire.

A (I,J)

II - Utilisation dos tableaux à 2 Dimonsions

Un tableau à doux dimensions peut être utilisés en Météo pour ranger les observations quotidiennes, mensuelles ou annuelles..etc Dans le tableau climatologique mensuel (TCM) par exemple, on peut utiliser un tableau à 2 dimensions dont l'indice de la ligne représentera le jour de l'observation, et l'indice de la colonne représentera l'heure de l'observation. Dans un tableau, on peut ranger ou (stocker) dos valeurs numériques oc! bien dos chaînes de caractères. Ce pondant il ost déconseillé do mélanger dans un même tableau des valeur3 numériques et des variables de chaînes, sauf si on est obligé (cas des observations marquantes).

NOMS SYMBOLIQUES DES TABLEAUX

Los noms symboliques à attribuer aux tableaux obéissent aux mêmes régles quo pour les autres variables BASIC,

- Tableau de variables ou de constantes numériques

On forme le nom symbolique do ces tableaux, en prenant d'abord une lettre alphabétique majusculo suivie par les indices (I,J) mis entra parenthèses.

Exemple : A (3,4) X(100,200) W9(40,12) sont des tableaux pouvant contenir des constantes ou dos variables numériques.

- Tableau des chaines de caractères : même lorsque les variables BASIC t le signe $

Exemple : A id (3,4) B5 $ (100,100) W9 $ (50,1)-

III - Nature des Eléments à stocker dans le tableau

- Eléments numériques

La valeur d'un élément numérique ne peut dépasser la valeur admise pour une variable BASIC quelconque : cette valeur dépend dos caractéristiques de la mémoire et do la version BASIC utilisée (donc du constructeur) - Cette valeur peut être :

Exemple : 32 767 1.4789 5 4.09E+02 3455E-08

- Chaînes de caractères

Les chaines de caractères à stocker dans un tableau peuvent être alphanumériques (mélange de lettres et chiffres), mais la longueur de chaque élément ne saurait dépasser 255 caractères quelque soit la version de BASIC utilisé et doivunt être mises en côtes" ".

IV - Les indices

- Les 2 indices doivent toujours être mis entre parenthèses et être séparés par une virgule. Los valeurs dos indices I et J doivent être comprises entre 1 et 32 767. Ils ne doivent pas être nuls.

Remarques : Saisie des données

1 - Lorsque au cours do la saisie des données on constate des observations manquantes, il faut s'adresse:: au chef do service Exploitation ou à l'Ingénieur responsables qui se chargera de fournir une estimation dus poromètres qui manquent.

II- La saisie dus données dans un contra de calcul pout se faire de plusieurs manières qui dépendent de la configuration du système (support - terminaux,. .etc. . .) et des logiciels disponibles (langage - editeur etc. . .)
pour une configuration donnée, la suivie des données en langage BASIC se fait généralement avec les instructions READ (vis à un programme) ou Ul. fichier) et INPUT (vis à **un** terminal ou un fichier),

Il est **donc** important que le programmeur qui a écrit un programme de saisie puisse spécifier à l'opérateur responsable de l a saisie le mode d'organisation dos données à saisir,

ENNONCE N° 16

(1) Ecrire un Programme de saisie qui range les observations mensuelles et crée un fichier concernant les sept paramètres suivants TS - TM - HR - E - D.D FF

N T C M des 30 BS par jour Expl station de Bouaké, sept 84

(2) Calculer les totaux et moyennes mensuels (sauf pour la direction du vent).

Problème

Solution Programme N° 16

```
LIST

REM          01:57:57
5 A=0 : B=0 : C=0 : D=0 : E=0 : F=0 : G=0
10 DIM A1(31,7)
11 READ R$(1),R$(2),R$(3),R$(4),R$(5),R$(6),R$(7)
15 FOR I=1 TO 31
20 INPUT A1(I,1),A1(I,2),A1(I,3),A1(I,4),A1(I,5),A1(I,6),A1(I,7)
25 NEXT I
30 DATA TS , TM ,HR ,E ,DD ,FF ,N
35 DATA
40 DATA
45 REM Pour (DD)Directionduvent,pasdetotalnidemoyenne.
50 FOR I=1 TO 31
55 A=A+A1(I,1) : REM THERMO SEC TS
60 B=B+A1(I,2) : REM THERMO MOUILLE (TW)
70   =C+A1(I,3) : REM HUMIDITE RELATIVE (HR)
80 D=D+A1(I,4) : REM TENSION DE VAPEUR (E)
90 F=F+A1(I,6) : REM VITESSE DU VENT (FF)
100 G=G+A1(I,7) : REM NEBULOSITE (N)
110 NEXT I
120 PRINT "TOTAL TS=";A;"MOY TS=";A/I
130 PRINT "TOTAL TW=";B;"MOY TW=";B/I
140 PRINT "TOTAL HR=";C;"MOY HR=";C/I
150 PRINT "TOTAL E=";D;"MOY E=";D/I
160 PRINT "TOTAL FF=";F;"MOY FF=";F/I
170 PRINT "TOTAL NEB=";G;"MOY N=";G/I
180 STOP

READY
```

```
120 TOTAL TS= 269 MOY TS= 8.67742
130 TOTAL TW= 310 MOY TW= 10
140 TOTAL HR= 198 MOY HR= 6.3871
150 TOTAL E= 175 MOY E= 5.64516
160 TOTAL FF= 183 MOY FF= 5.90323
170 TOTAL NEB= 318 MOY N= 10.2581
```

Impression

TOTAL TS = 269

Mon TS 8.6

TS = 269

SOLUTION N° 18

```
USER 00 LOGGED OFF -- GOOD BYE

MU BASIC-11/RT-11 IS OPERATIONAL.
PLEASE TYPE IN "HELLO".


HEE\E\LLO
WELCOME TO MU BASIC-11/RT-11

READY


OLD DY0:12HEUR.B00

READY

LIST

12HEUR                02:54:15
5  K=0 \ A=0 \ B=0 \ C=0 \ D=0 \ E=0 \ F=0 \ G=0
10 DIM A1(31,7)
11 PRINT 'QUEL EST LE NOM DE LA STATION ' ; \ INPUT A$
12 PRINT "ENTREZ LE MOIS , L'ANNEE "; \ INPUT B$,C3
13 IF C3=INT(C3/4)*4 THEN K=1
14 GOSUB 1710
15 FOR I=1 TO P
20 READ A1(I,1),A1(I,2),A1(I,3),A1(I,4),A1(I,5),A1(I,6),A1(I,7)
25 NEXT I
30 DATA ,26,1,25.0,91,30.8,0,0,7
31 DATA 26.8,25.4,89,31.4,0,0,0
32 DATA 26.0,24.1,85,28.6,0,0,7
33 DATA 26.3,24.6,86,29.6,0,0,6
34 DATA25,6925,0995Y31,2009094
35 DATA 25.0,23.7,90,28.3,0,0,4
36 DATA 24 .5,23.3,90,27.8,0,0,5
37 DATA24 . 5 923.5,92,28.2,0,0,7
38 DATA 25.0,24.0,92,29.1,0,0,7
39 DATA 24.6,23.8,93,28.9,0,0,4
40 DATA26,9925 . 8 , 9 1, 32 . 5 Y28 Y6 Y7
41 DATA 26.0,25.5,96,32.3,0,0,0
42 DATA 23.2,22.5,94,26.7,0,0,6
43 DATA 23.5,22.6,94,26.8,0,0,5
44 DATA 24 .3,23.6,94,28.6,9,4,8
45 DATA 25.1,24.0,91,29.0,4,4,6
46 DATA 26.0,25.6,97,32.5,0,0,0
47 DATA 26.8,25.3,88,31.2,0,0,0
48 DATA 26 .5,26.1,97,33.6,0,0,6
49 DATA 25.0,24.0,92,29.1,0,0,6
50 DATA 24.0,23.8,99,29.3,0,0,3
51 DATA 26.4,25.2,90,31.1,0,0,1
52 DATA 25.0,24.2,94,29.6,0,0,2
53 DATA 25.5,25.1,97,31.7,0,0,0
54 DATA 26.9,26.0,93,32.9,0,0, 1
55 DATA 23.9,23.4,96,28.4,0,0,0
56 DATA 25.4,25.1,98,31.7,0,0,0
57 DATA 26.0,25.4,95,32.0,0,0,1
58 DATA 26.6,26.1,96,33.5,0,0,0
59 DATA 23.6,21.5,82,24.0,0,0,5
300 FOR I=1 TO P
400 A=A+A1(I ,1) \ REM THERMO SEC TS
500 B=B+A1(I,2) \ REM THERMO MOUILLE TW
600 C=C+A1(I,3) \ REM HUMIDITE RELATIVE HR
700 D=D+A1(1:,4 ) \ REM TENSION DE VAPEUR E
```

ENONCE : I ECRIRE Un Programme q
de sauvegarde les obs mensuelles de 0800 co
cernant les sept Paramètres suivants : TS, TM, HR, E
DD, FF, N (TCM des 3 obs par Jour) pour le mois de sept
1984 de La station de BAMBEY -
II Calculer les TOTaux et Moyennes mensu
sauf pour la Direction du vent -/-

```
800 F=F+A1(I,6) \ REM VITESSE DU VENT FF
900 G=G+A1(I,7) \ REM NEBULOSITE N
1000 NEXT I
1100 PRINT "TOTAL TS=";A;"MOY TS=";A/I
1200 PRINT "TOTAL TW=";B;"MOY TW=";B/I
1300 PRINT "TOTAL HR=";C;"MOY HR=";C/I
1400 PRINT "TOTAL E=";D;"MOY E=";D/I
1500 PRINT "TOTAL FF=";F;"MOY FF=";F/I
1600 PRINT "TOTAL NEB=";G;"MOY NEB=";G/I
1700 GO TO 2500
1710 IF B$="JANVIER" THEN P=31
1720 IF B$="FEVRIER" THEN P=28+K
1730 IF B$="MARS" THEN P=31
1740 IF B$="AVRIL" THEN P=30
1750 IF B$="MAI" THEN P=31
1760 IF B$="JUIN" THEN P=30
1765 IF B$="JUILLET" THEN P=31
1770 IF B$="AOUT" THEN P=31
1780 IF B$="SEPTEMBRE" THEN P=30
1785 IF B$="OCTOBRE" THEN P=31
1790 IF B$="NOVEMBRE" THEN P=30
1795 IF B$="DECEMBRE" THEN P=31
1800 RETURN
2500 END

READY
RUN

12HEUR          02:55:42
QUEL EST LE NOM DE LA STATION ? BAMBEY
ENTREZ LE MOIS , L'ANNEE ? SEPTEMBRE,1984


TOTAL TS= 761 MOY TS= 25.3667
TOTAL TW= 733.2 MOY TW= 24.44
TOTAL HR= 2777 MOY HR= 92.5667
TOTAL E= 900.4 MOY E= 30.0133
TOTAL FF= 14 MOY FF= .466667
TOTAL NEB= 108 MOY NEB= 3.6

READY
```

B - LES SOUS PROGRAMMES

Introduction : Un sous programme est constitué par un ensemble d'instruction dont la première peut-être n'importe qu'elle instruction BASIC (REM - LET - FOR TO - INPUT...) y compris l'instruction DIM) et la dernière est l'instruction RETURN.

I - Structure des sous programmes

- Un sous programme ost un sous ensemble d'un programme principal que l'on appelle souvent programme source (Main-program) en Anglais.

- A partir d'un programme principal, on peut transférer l 'exécution à un sous programme à l'aide de l'instruction GOSUB. Ce précédé est équivalent à un saut. Dans le langage courant, on dit "Appeler un sous programme". On peut mettre autant de sous programmes qu'on veut dans un programme principal.

II - Fonctionnement et Exemple

1°/- Syntaxe : La régle de syntaxe (Technique pour former les mots clefs et les instructions) à appliquer dans un sous programme est la même que pour les autres instructions du BASIC, Chaque in:;-truction du sous programme est composée d'un N° de ligne et d'une instruction BASIC complets,

Exemple :

```
800 DIM$ (35,50), L1 (20)
810 FOR T = 1 TO 100
840 FOR I = 1 TO 35
860 FOR J = 1 TO 50
870 I F A $ = F $ (I,J) THEN 900
880 NE X T J
885 NEXT I
900 PRINT A $
910 NEXT T
940 RETURN.
```

L'instruction d'APPEL GOSUB

L'instruction GOSUB appartient au programme principal, elle est composée d'un N° de ligne, du mot clef GOSUB et d'une adresse (N° de la ligne Ou commence le sous programme). Lorsque la machine exécute l'instruction GOSUB, elle transfert l'exécution à la ligne dont le N° est indiquée après le mot clef GOSUB. Le N° de ligne qui se trouve après le GOSUB doit correspondre au 1er N° de ligne du sous

Quand la machine exécute la dernière instruction du sous programme (RETURN), elle transfert le contrôle à 1 'instruction du programme principal qui vient après le GOSUB.

Exemple :

```
10  A = 15 / B = 18 / C = X1 (20)
20  GOSUB 50
30  GO TO  90
50  PRINT A * B/C
60  IF C = (B/C) THEN F1 = 0
70  PRINT A * C
80  RETURN
90  REM
100 STOP
```

SOUS PROGR.

- Un sous programme de travail peut-être divisé en plusieurs sous programmes. Chaque sous programme se charge de traiter un problème particulier. Cette technique est appelée Programmation structurez,

TABLEAU DES CARACTERES ASCII RECONNUS PAR LE BASIC

Les 26 lettres de l'alphabet (Majuscules et Minuscules)
Les 10 unités du système décimal de (0 à 9)

LES CARACTERES

SP ou A = barre de spacement

! = Point d'exclamation

; = Point virgule

, = virgule

" = double côte

« » = différent

= Numéro

" " = double guillemets

$ = dollar

= pourcentage

& = Ampersand

' ' = guillemets simples

(= parenthèse gauche

) = -"- droite

* = Astérix

+ = Plus

- = Moins

~ =

. = Point

/ = Slash

: = double point

< = inférieur

= : égal

> = supérieur

@ = sign

[= bracket gauche

\ = back slash

] = bracket droit

^ = Accent circonflexe

_ = tiré horizontal

= Accent grave

{ = Guillet gauche

| = tiré vertical

} = Guillemet droit

NB : Il y a aussi des caractères spéciaux de contrôle.

LA BIBLIOTHEQUE DU BASIC

A partir de la version standard du BASIC, d'autres versions ont été développées à partir des années 70 (BASIC +, MUBASIC, Commodore BASIC....etc). Chacune du ces versions contient la Bibliothèque standard de BASIC avec en plus des innovations apportées par les ingénieurs du logiciel, Il appartient à chaque programmeur Analyste d'étudier la version sur laquelle il travaille pour apporter des innovations dons la bibliothèque, ou pour comprendre certaines nouvelles fonctions.

I - LES FONCTIONS STANDARDS DU BASIC

- L'ensemble des algorithmes (procédé de calcul) programmés est mis dans un fichier qu'on appelle Bibliothèque. pour utiliser une fonction classée dans une Bibliothèque, il suffit de l'appeler, et de lui donner un argument,

Exemple : Ce programme calcul8 le SINUS de tous les Nombres compris entre 0 et PI (en Radians)

```
10 FOR T = 0 TO PI STEP 0.1
20 PRINT = SIN (T)
30 NEXT T
```

- La Bibliothèque du BASIC contient deux catégories de fonctions :
- Celles qui ont comme argument des constantes ou des variables numériques définies (Fonctions numériques).
- Celles qui ont comme argument dus constantes ou des variables de chaines mises entre " " (Fonctions Alphanumériques).

LES FICHIERS

- Un fichier est un ensemble ordonné d'information : a n peut y mettre n'importe quel type d'information,
- Il y a 2 sortes de fichiers : séquentiels et accés direct ou(virtuels Raindom)

* SEQUENTIELS : Les enregistrements, ou informations élémentaires sont d ispo séés les unes après les autres, chaque instruction apparais sant sur une ligne séparée.

A - Création : les fonctions d'un fichier sont, réalisées par les commandes du système

Ecriture $ New.....OLD.....SAVE.....LIST.....etc (son nom ne peut dépasser plus de 6 caractères,

OPEN Nom du Fichier FOR out PUT AS FILE #(vari 1), File SIZE, vari
(const) (taille du fichier) const

Record SIZE, const,
(taille enregistrement) vari

PRINT #1, A $

Exemple : 10 OPEN "MY FILE" FOR OUT PUT AS FILE #1
20 FOR I = 1 TO 10
30 PRINT 1, 1
40 NEXT I
57 CLOSE~ 1
60 END

Lecture : OPEN Nom du fichier INPUT ASFILE #2 vari const

INPUT #2 D$

Exemple : 10 OPEN "MY FILE FOR INPUT AS FILE #2
20 FOR S = 1 TO 10
30 INPUT # 2, S
40 NEXT S
50 CLOSE # 2
60 END

Accès direct :

Les informations ne sont pas ordonnées, chaque élément peut être lu ou écrit sur le fichier directement sans traiter le fichier séquentiellement depuis le début.

A - CREATION :

OPEN nom du fichier FOR out PUT AS FILE #(const vari),1 FILE SIZE vari const RECORD SIZE, const vari

DIM #1, S $ (100, 100) = 90
S $ (1 , 1) = "NGOM"

Exemple :

```
10 OPEN "MY F ILE " OUT PUT AS FILE £3
2 0 D IM £ 3 S$ (100,100) = 30
30 FOR I = 1 TO 31
40 INPUT  S$ (I,1)
50 NEXT I
60 CLOSE £ 3
70 END
```

(B) LLCTURE

OPE N Nom du fichier FOR INPUT AS FILE #1 const var

DIM # C/V, S $ (100,100) = C/V

```
10 OPEN "MY FILE" FOR INPUT AS FILE #1
20 DIM #1   SS (100, 100) = 9C
30 FOR I = 1 TO 31
40 PRI&T S$ (1,1)
50 NEXT I
6C CLOSE&
70 END
```

SPECIFICATION EN COURS D'EXECUTION

On peut introduire un nom de fichier à l'exécution si on le désire. Pour cela, il faut se servir non de l'instruction FILES, mais de l'instruction FILE : les 2 sont différents,

- LES COMMANDES DU BASIC

Introduction :

Les commandes et des instructions que le BASIC exécute immédiatement sans les mettre dans un programme. Donc les commandes ne nécessitent pas de N° de lignes. Il existe 2 types de commandes dans le BASIC :

- Celles qui sont directement tapées sur le clavier telles qu'elles (commandos directs)
- celles qui sont tapées en appuyant sur des touches particulières du clavier (commandes indirectes)

* Toutes les commandes se terminent on appuyant sur la touche RETURN

A - LES COMMANDES DIRECTES

1 - New : (nom du programme) : la commando New permet de créer Un nouveau programme. Il suffit d'écrire le mot clef New suivi du nom du programme à créer, et d'appuyer sur la touche RETURN. Avant d'utiliser la commando New, il faut s'assurer que les programmas antérieurement créés / ont été sauvés par les commandes SAVE ou REPLACE.

2 - OLD (nom du programme) permet d'appeler un programmo antérieurement créé

3 - SAVE : (Nom programme) permet de sauver un programmo créé sur le Terminal. La commande SAVE est utilisée lorsque le programme est créé pour la première fois

4 - REPLACE : Cette commande est équivalente à SAVE, à la seule différence qu'elle sauve toute nouvelle version postérieure à la première version. Replace efface toutes les versions antérieures du même programme et conserve la dernière version.

5 - LIST : Elle permet de lister (imprimer sur le terminal) un programme ou une partie du programme, pour lister toutes les lignes comprises entre la ligne 350 et la ligne 400 du programme taux, on peut mettre les commandes suivantos :

OLD TAUX - - - - appelé le programme Taux
LIST 350-400 List 350 jusqu'à 400

6 - RUN : Cet te commande perme t d'exécuter un programme, il suffit de mettre RUN suivi du Nom du programme. Lorsque la machine exécute cette commande elle transfert le programme (ou une copie) du périphérique extérieur (Terminal, ou diskette, ou Bande magnétique, vers le mémoire centrale. Cette copie contient des instructions supplémentaires que la machine ajoute elle même. Cette copie est appelée programme COMPILE.

7 - DEL : C'est une abréviation du mot clé DELETE. Cette commande permet d'effacer les lignes appartenant à un programme.

NB : RUBOUT - même signif ication que DELETE .

Exemple : DEL 10 ----- Efface la ligne 10

DEL 50 - 80 --- Efface toutes les lignes comprises entre 50 et 80

8 - S U B : C'est le diminitif du mot clé SUBTITUTE. Cette commande permet de changer un groupe de caractères par un autre groupe de caractères dans uni: même ligne. Pour cela on écrit le mot clef SUB suivi du N° de la ligne suivi par un Délimiteur, suivi du groupe de caractères à changer, suivi du même délimiteur , suivi du groupe de caractères à mettre en place. Eventuellement de l 'occurence du groupe du caractères à changer,

Exemple : SUB 10/A = B + 2 * F9/A = B - 2 * A5

Le Délimiteur utilisé ici est la barre oblique (/). Après exécution de cette commande. La machine a changé le B+ en B- dans l'instruction N° 10 du programme.

9 - RESEQ : C'est le diminitif du mot clef RESEQUENCE : Elle permet de changer tous les N° de lignes d'un programme, et d'assigner un incrément fixe à ces N° de lignes.

10- UNSAVE : Cette commande permet d'effacer tout un programme. Il faut faire attention dans l'utilisation de cette commande qui peut effacer un programme même lorsqu'il est situé sur un périphérique,,,.etc. Pour l 'exécuter on met UNSAVE suivi du nom du programme.

11 - RENAME : Cette commande permet de changer le nom d'un programme. On écrit RENAME en toutes lettres et on appuie sur la touche RETURN ; la machine répond : NEW FILE NAME...
On répond par le Nouveau Nom du programme après les 2 Tirets. Il ne faut pas oublier d'utiliser après cela la commande REPLACE pour sauver le nouveau nom du programme.

12 - COMPILE : Pour compiler un programme ou accélérer son temps d'accès ou d'exécution afin de faciliter la lecture.

KILL : Nom du Fichier : Peut servir à enlever un fichier,

A - LES COMMANDES INDIRECTES

Elles sont exécutées en appuyant sur la touche C T R L et en appuyant en même temps sur une lettre du clavier,

1/- CTRL/C : La commande contrôle C peut être exécutée une ou deux fois pour arrêter tout processus que l'on BASIC est entrain d'exécuter (exécution du programme, lister une page, ou entrer des données) Cette commande est l'équivalent de stop,

2/- CTRL/O = Cette commande empêche toute sortie sur le Terminal même si un processus interne est en cours.

3/- CTRL/S = Cette commande permet de "glacer" un terminal. Le Terminal suspend temporairement tout processus qui est en cours et il n'accepte plus aucune commande à part contrôle Q ,

4/- CTRL/Q = Cette commande permet d'annuler les offets du contrôle s. Le Terminal peut continuer le travail en cours, ou accepter de nouvelles commandes.

5/- CTRL/U : Cette commande permet d'annuler une ligne entière d'un programme BASIC.

6/- CTRL/L = cette commande permet d'avancer le curseur du Terminal d'une dizaine de lignes. Elle est surtout utilisée dans l'édition de texte pour créer des paragraphes.

7/- CTRL/M = RETURN / CTRL/Z = FIN INPUT/ CTRL/K = TAB vertical (VT)

CTRL/I : TAB horizontal / CTRL/J = Line FEED (saut de ligne)

METHODE POUR CORRIGER DES ERREURS SUR UNE LIGNE

1
READY
SUB N° de ligne / .. / . -----RETURN
1000 Print : Nombre de Précip >= 0.1mm ; P1

Après avoir tout changé, tu écris REPLACE pour rectifier les erreurs précédentes.

2
SUB 1000 /../. ---- RETURN
1000 PRINT. Nbre de précipi >= 0.1mm. ; P1
} Bon

B - LES TOUCHES

- ESC) Mode pour faire des codes, de Mvts du curseur
  SEL)

- TAB = Tabulation (7 caractères même 8 c'est possible :) Espaces.

- CTRL = Contrôle

- CAPS) Majuscules - Minuscules.
- LOCK)

- SHIFT : Pour l'utilisation des caractères situés an haut d'une touche
- COPY = valable pour photocopier les données de l'écran
- REPEAT = Répéter un caractère
- RETURN) Renvoi au système, Retour du chariot
- CNTER)

- DELETE ou DEL) sert à effacer un caractère de trop, ou mal frappé-
- RUBOUT) On frappe autant de fois sur la touche RUBOUT ou
) DELETE pour offacor autant de caractères après on
) frappa la Touche RETURN,

- BLACK SPACE = Retourner un caractère un peu en arrière vers la gauche
  ou H
- Lino PEED = Sauter une ligne.

- BELL = Sonnerie ----- CTRL/G

- ALT MODE ou) Sert à effacer la totalité d'une ligne en cas d'erreur
  ESPAC:E) et puis la refrapper avec son N° de ligne d'instruction
  ESC)

. Un autre cas suffit de frapper le retour chariot RETURN et de reprendre intégralement toute la ligne.

- HOME = Point initial du curseur ou bien point do départ

- C L R = sert à effacer
- RESET = initialiser
- SCRATCH) Efface le Programme courant vient après le READ.
- S C R)

II - COURS PRATIQUES

FONCTIONNEMENT DE L'ORDINATEUR

Les programmes BASIC qui permettent de travailler sont les suivants :

A - BASIC . CAVE /

B - Les Fichiers de configuration 1 USER .CNF --1 utilisateur ou 1 terminal
2 USERi .CNF --2 utilisateurs
3 à 8 USER.CNF - 3à8 utilisateurs

Lorsque votre osdinateur est allumé, la 1ère chose que vous voyez sur l'écran représente un point (.) ou un signe qui scintille : c'est le moniteur

1ère ETAPE :

1 ♠ Date 9 - MAI - 85 -----→ RETURN

2 ♠ TIME 10 : 05 : 40 -----→ RETURN
H min sec

Si le début est bien fait, le Point (.) réapparait, et vous pouvez continuer

3 ♠ BASIC ....RETURN

Ecrit par l'ordinateur
4 - MU BASIC - 11/RT - 11 V2.1 version
5 - Configuration FILE : * Z-USER

6 - MU BASIC - 1 1/RT 11 IS OPERATIONNAL
Maintenant le BASIC - est Opérationnel

Ordinateur
7 - PLEASE TAPE IN HELLO ----- Tapez HELLO s'il vous plait

8 - HELLO

Ordinateur
9 - WELCOME TO MU BASIC - 11/RT 11
10- READY
11- NEW FILE NAME = (Nom du programme)Tu écris Exemple PLUVIO

2ème ETAPE : TU FAISENTIER Ton programme

```
10  . . ..*a.....
20 ....••••••••••••••
30 .a.....*.......
1000  END  ou  Stop
```

NB : Après chaque 1 ligne tapée on appuie sur RETURN

3ème ETAPE : Elle consiste à vérifier le programme déjà introduit On tape LIST et on appuie sur RETURN : s'il n'y a pas d'erreur, on tape RUN et on sauve le programme par SAVE.

4ème ETAPE : Pour un programme déjà existant, on écrit OLD et on frappe RETURN.

L'ordinateur : répond : OLD FILE NAME

- ot Tu tapes ----- le nom de ton ancien programme Exple : PLUVIO, puis tu appuies sur RETURN, ut
- l'ordinateur écrit READY :
- Tu RUN ou LIST ton programme.

```
000 000207 156464 000030

000 173200 165212 156464

000 173200 165212 165212

11XM (S) V04.00P

T TT:QUIET
BASIC-11/RT-11 IS OPERATIONAL.
ASE TYPE IN "HELLO".

LO
 RECT
COME TO MU BASIC-11/RT-11

DY

DY
T

T        00:07:24
REM SURFACE D'UN RECTANGLE...
PRINT "LONGUEUR ET LARGEUR";
INPUT L1,L2
S=L1*L2
PRINT " LA SURFACE =";S
PRINT
GO TO 10
END

DY


T        00:07:33
GUEUR ET LARGEUR? 45,23
 SURFACE = 1035

GUEUR E   T   LARGEUR? ^C

P AT LINE 30

DY
```

SOLUTION N° 3

ENONCE N° 3

ECRIRE Un programme qui calcule la surface d'un rectangle : le Programme doit être répétitif ./.

```
OLD
OLD FILE NAME--PART

READY

LIST

PART          CC:39:19
1( REM LISTE CE 2 ELEMENTS CCINSECUTTFS
2( DIM A2(4),A1(5)
3( PRINT "ENTREZ LES ELEMENTS DE LA LISTE";
41 INPUT A1(1),A1(2),A1(3),A1(4),A1(5)
5( REM
6( A2(1)=(A1(1)+A1(2))/2
7( A2(2)=(A1(2)+A1(3))/2
8( A2(3)=(A1(3)+A1(4))/2
9( A2(4)=(A1(4)+A1(5))/2
1(( REM
11( REM
12( PRINT "A2(1)=";A2(1)
13( PRINT "A2( 2)=";A2( 2)
14( PRINT "A2(3)=";A2(3)
15( PRINT "A2(4)=";A2(4)
16( REM
17( GO TO 3(
1E( STOP

READY
```

Solution Problème .

N° . 8

25 lignes

ENONCE 8

Considérons l'exercice suivant.

On donne une liste A1 de 5 éléments

$A1 = \{ 2, 5, 15, 20, 25.3 \}$

A partir de cette liste, Faire une 2e liste dont On donnera la dimension et dont chaque élément est la moyenne de 2 éléments consécutifs de La liste A1 -/-

SOLUTION: Nº 13

```
NDONGO     05-JUL-85     03:57:22
10 REM
20 PRINT TAB(7);
30 FOR I=1 TO 7 \ PRINT I; \ NEXT I
35 PRINT
40 FOR I=1 TO 7
45 PRINT I;
47 FOR J=1 TO 7
50 PRINT I*J;
55 NEXT J
60 PRINT
70 NEXT I
80 END

READY
RUN

NDONGO     05-JUL-85     03:57:31
          1   2   3   4   5   6   7
 1   1   2   3   4   5   6   7
 2   2   4   6   8   10  12  14
 3   3   6   9   12  15  18  21
 4   4   8   12  16  20  24  28
 5   5   10  15  20  25  30  35
 6   6   12  18  24  30  36  42
 7   7   14  21  28  35  42  49

READY
```

ENONCE Nº 13

Ecrire Un Programme qui donne comme sortie la Table de multiplication par 7 sans avoir besoin de lire des données à l'entrée.

C O N C L U S I O N

Si l'informatique en tant que science accuse un très grand retard au niveau des pays sous développés, l'ordinateur est devenu depuis quelques temps un outil précieux mis à la disposition do l'homme dans la cadre de ses diverses activités,

Ainsi la connaissance dos ressources que nous pourrions tirer de l'informatique et de l'emploi des ordinateurs peut s'avérer utile à certains d'entre nous, surtout dans le cadra de l'exploitation et le stockage des données METEO et Agrométéo.

Dommage que le délai assez court no m'a pas permis d'un tirer le maximum possible pour mes futurs travaux de recherches.

pour terminer, je réitère mes plus sincères remerciements à tous ceux qui n'ont aidé au bon déroulement du stage, en particulier à Mr Omar LO qui fût mon encadreur durant toute ma présence au Centre de calcul de la Direction do la Météorologie Nationale.