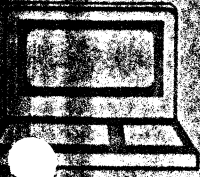


SIVEAS A...



522.70.66

LES PAS

UN COURS PROGRAMME

DE BASIC
EN FRANÇAIS

apple
INC.

apple est une marque déposée d'Apple Computer Inc.

PAS-A-PAS est un cours qui vous apprendra à programmer en BASIC. Il est étudié pour les personnes qui n'ont aucune expérience en programmation. Il a fallu une année pour développer son usage sur cinq types différents de petits ordinateurs, le testant avec des jeunes étudiants, des amateurs, et des débutants en programmation.

Les professionnels de l'ordinateur pensent souvent que le BASIC est un langage de programmation simple que tout le monde peut apprendre en quelques heures de pratique. Le fait est que, bien que le BASIC soit plus facile à apprendre que l'assembleur ou le FORTRAN, il n'est pas si facile que cela à apprendre. Une personne doit apprendre à penser en termes de programmation. Pour programmer dans des langages plus complexes, une personne a besoin de beaucoup d'habileté, mais elle en aura aussi besoin en BASIC.

PAS-A-PAS enseigne les caractéristiques de programmation qui sont communes à tous les petits ordinateurs parlant le BASIC. Il n'est pas prévu pour enseigner les caractéristiques spéciales de l'un d'entre eux bien que quelques spécialités sont décrites. Nous pensons qu'une personne doit d'abord apprendre les bases puis ensuite apprendre les caractéristiques spéciales de son propre ordinateur. Les caractéristiques spéciales sont les graphiques, l'édition de textes et les manipulations de chaînes.

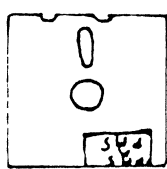
Comment utiliser PAS-A-PAS :

Les leçons du cours doivent être suivies dans l'ordre. Aucune leçon ne doit être sautée même si vous sentez que vous connaissez déjà tout le contenu de la leçon. Vous pouvez faire autant que vous voulez en une séance. Mais, il est conseillé de ne pas trop espacer les séances. Rappelez-vous aussi que les séances de pratique sont extrêmement importantes pour l'apprentissage.

Votre cours PAS à PAS se présente sur diskette.

- La face 1 comporte les leçons 1 à 5 avec les questionnaires correspondants
- La face 2 comporte les leçons 6 à 10

1) Mettre la diskette 1 dans le drive avec l'étiquette face au dessus



INTRODUIRE DANS CE SENS

2) Mettre votre moniteur ou téléviseur sous tension.

3) Mettre l'APPLE II sous tension.

Le lecteur de diskette va se mettre en route seul.

La leçon n° 1 va commencer.

Les leçons sont divisées en 2 parties avec un questionnaire correspondant à chaque leçon, à la fin de la leçon 1 première partie si vous désirez continuer ...

TAPER

RUN [] LECON [] 1 [] PARTIE [] 2 [] SIGNIFIE ESPACE.

ETC...

A la mise sous tension, si on met la diskette PAS à PAS dans le drive. La première leçon démarre automatiquement. (ceci pour vous faciliter les premières manipulations).

Si vous voulez passer à une autre leçon taper RESET.

Vous obtenez alors le curseur []

Taper : RUN LECON...

ATTENTION : POUR LES UTILISATEURS APPLE LIRE RETURN À LA PLACE DE "ENTER"

←CATALOG

DISK VOLUME 254

CONTENU DE LA FACE N° 1

A 002 HELLO
 *A 017 LECON 1 PARTIE 1
 *A 015 LECON 1 PARTIE 2
 *A 011 QUESTIONS SUR LECON 1
 *A 019 LECON 2 PARTIE 1
 *A 020 LECON 2 PARTIE 2
 *A 011 QUESTIONS SUR LECON 2
 *A 019 LECON 3 PARTIE 1
 *A 018 LECON 3 PARTIE 2
 *A 014 QUESTIONS SUR LECON 3
 *A 018 LECON 4 PARTIE 1
 *A 016 LECON 4 PARTIE 2
 *A 012 QUESTIONS SUR LECON 4
 *A 016 LECON 5 PARTIE 1
 *A 018 LECON 5 PARTIE 2
 *A 012 QUESTIONS SUR LECON 5

←CATALOG

DISK VOLUME 254

CONTENU DE LA FACE N° 2

A 002 HELLO
 *A 018 LECON 6 PARTIE 1
 *A 015 LECON 6 PARTIE 2
 *A 012 QUESTIONS SUR LECON 6
 *A 018 LECON 7 PARTIE 1
 *A 018 LECON 7 PARTIE 2
 *A 014 QUESTIONS SUR LECON 7
 *A 018 LECON 8 PARTIE 1
 *A 019 LECON 8 PARTIE 2
 *A 012 QUESTIONS SUR LECON 8
 *A 017 LECON 9 PARTIE 1
 *A 022 LECON 9 PARTIE 2
 *A 011 QUESTIONS SUR LECON 9
 *A 019 LECON 10 PARTIE 1
 *A 018 LECON 10 PARTIE 2
 *A 009 QUESTIONS SUR LECON 10
 *A 017 REVISION LECONS 1 A 5
 *A 012 REVISION LECONS 6 A 10

LECON 1, PARTIE 1

● Paragraphe 1 :

Regardez le clavier de votre ordinateur. Quand vous tapez quelque chose à l'ordinateur, ce n'est pas enregistré tant que vous n'avez pas appuyé sur la touche "ENTER", sur le côté droit du clavier .

Quand vous tapez un zéro, n'utilisez pas la lettre O. Le zéro de l'ordinateur a une barre en travers, comme ceci : Ø

Quand vous tapez un 1, n'utilisez pas la lettre L. L'ordinateur a une touche pour le nombre 1.

Si vous avez fait une faute et voulez changer votre réponse, appuyez sur la touche de retour-arrière, celle avec une flèche dirigée vers la gauche.

● Paragraphe 2 :

Ces nombres sont ce que vous trouvez dans la mémoire de votre ordinateur. Notez que les lettres A,B,C,D,E et F sont des nombres pour l'ordinateur.

```

ØFØØ 21 ØØ 45 4D 11 ØØ ØE CD 1F ØF B9 CA Ø8 ØF 77 ØØ
ØF1Ø 8Ø 47 23 1B 7A B3 CA 29 ØF CD 1F ØF C3 ØF ØF DB
ØF2Ø ØØ E6 40 CA 1F

```

● Paragraphe 3 :

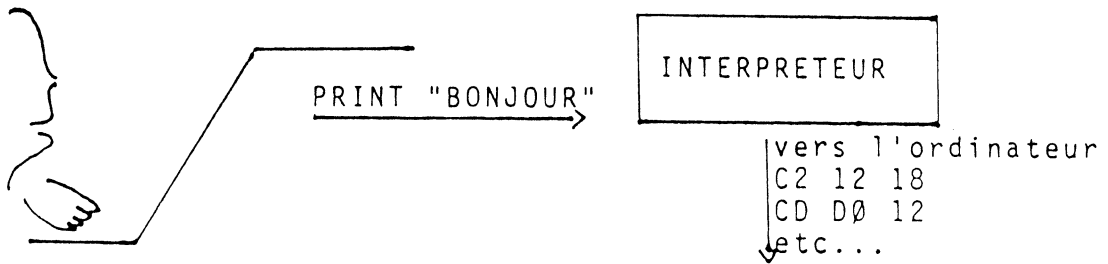
Programme en langage machine :

```

FFØØ D8
FFØ1 58
FFØ2 AØ 7F
FFØ4 8C 12 DØ
FFØ7 A9 A7
FFØ9 8D 11 DØ
FFØC 8D 13 DØ
FFØF C9 DF

```

- Paragraphe 4 :



- Paragraphe 5 :

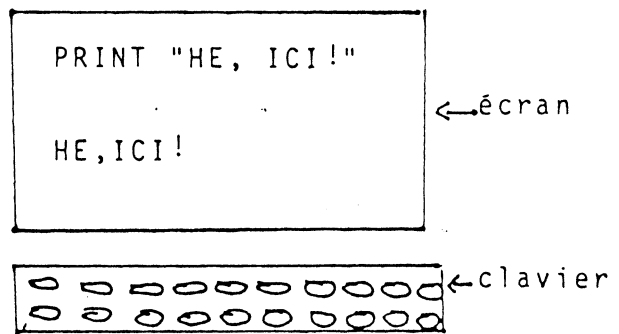
Quelques commandes en BASIC :

`PRINT` (imprimer); `STOP` ; `RUN` (exécuter); `NEW` (nouveau);
`SAVE` (sauver); `LOAD` (charger); `INPUT` (ENTRER); `GOTO` (aller à);
`READ` (lire); `DATA` (données); `LET` (soit); `FOR` (pour);
`NEXT` (suivant); `RETURN` (retour); `IF... THEN` (si...alors);

- Paragraphe 6 :

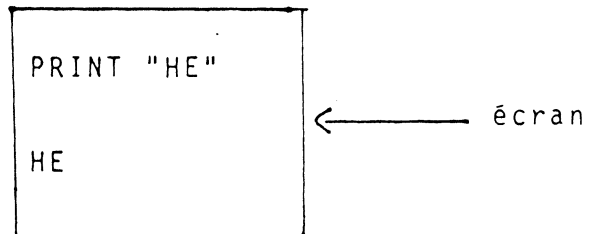
Utilisation de la commande `PRINT`

vous tapez →
 appuyez sur ENTER
 l'ordinateur répond



- Paragraphe 7 :

votre instruction →
 l'ordinateur imprime ceci →

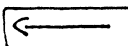
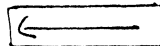



- Paragraphe 8 :


Voici quelques choses pour pratiquer sur votre ordinateur. Regardez ce qui arrive quand vous tapez les commandes suivantes à votre ordinateur. Notez bien que les guillemets sont au-dessus du 2. Vous devez appuyer sur la touche marquée SHIFT pour taper les guillemets. Appuyez sur la touche ENTER après chaque commande.

```
PRINT "BONJOUR" ← appuyez sur la touche ENTER après chaque ligne
PRINT 'BONJOUR'
PRINT BONJOUR
PRINT "BONJOUR
PRINT "BONJOUR ← Note : sur certains ordinateurs, ceci ne produira pas de message d'erreur. Mais tapez toujours le deuxième guillemet ou vous aurez des surprises avec des commandes plus complexes !
```

- Paragraphe 9 :

Si vous avez fait une faute en tapant une ligne, vous pouvez la corriger en utilisant la touche . Appuyez sur la touche  pour revenir en arrière dans la ligne d'autant d'espaces que nécessaire puis tapez les caractères corrects. Essayez maintenant.

```
tapez : PRINT "BONJOUR ICI"
        puis appuyez 8 fois sur la touche  puis de tapez la ligne
        PRINT "BONJOUR ICI"
```

Essayez ces commandes, en utilisant la touche  pour corriger toutes les erreurs que vous faites :

```
PRINT "VOTRE NOM"
PRINT "'VOTRE NOM'"
PRINT "MON NOM EST"JEAN " "
PRINT "MON NOM EST'JEAN'"
```

Maintenant, à votre tour, utilisez la commande PRINT. N'utilisez pas de nombres cette fois. Quand vous serez prêt, chargez la partie 2 de la leçon 1 dans l'ordinateur.

LECON 1, PARTIE 2

● Paragraphe 10 :

vous tapez _____ → PRINT 1 + 5
 appuyez sur ENTER
 l'ordinateur répond _____ → 6

vous tapez _____ → PRINT 6 - 1
 appuyez sur ENTER
 l'ordinateur répond _____ → 5

● Paragraphe 11 :

vous tapez _____ → PRINT " 3 + 5 "
 appuyez sur ENTER
 l'ordinateur répond _____ → 3 + 5

● Paragraphe 12 :

*	: multiplication	2 * 5 = 10
/	: division	10 / 2 = 5
+	: addition	3 + 3 = 6
-	: soustraction	10 - 6 = 4

● Paragraphe 13 :

* et / sont faites en premier

Problème : $3 + 8 * 2 / 4$

$3 + 16 / 4$ } * et / sont faites

$3 + 4$ } puis + est faite

7 ← l'ordinateur imprime 7

Problème : $10 - 50 / 2 + 8 * 3$

$10 - 25 + 24$

9 ← l'ordinateur imprime 9

● Paragraphe 14 :

Problème : $(25 + 8 - 32) / (9-8)$

1 / 1 les calculs entre () sont faits en premier

1 ← l'ordinateur imprime 1

● Probleme : $16 * (4 + 3 - 2) / 16 / (10 - 9)$

$16 * 5 / 16 / 1$

$80 / 16 / 1$

5 ← l'ordinateur imprime 5

- Paragraphe I5 :

voici comment l'ordinateur calcule :

- 1) multiplications et divisions à l'intérieur des ()
- 2) additions et soustractions à l'intérieur des ()
- 3) multiplications et divisions à l'extérieur des ()
- 4) additions et soustractions à l'extérieur des ()

voici un exemple :

Problème : $(3 * 4 + 1) + 2 * (3 + 3)$
 $(12 + 1) + 2 * (3 + 3)$ ← *et / à l'intérieur des ()
 $13 + 2 * 6$ ← +et - à l'intérieur des ()
 $13 + 12$ ← *et / à l'exterieur des ()
 25 ← + et -

- Paragraphe I6 :

Passez à la pratique de ces commandes. Tapez chacune de ces commandes et appuyez sur ENTER. Regardez ce que l'ordinateur répond :

```
PRINT 2 + 6
PRINT 3 * 9
PRINT 48 / 6
9000 / 2
PRINT 5 + 6 - 3 + 7 + 19
PRINT " 3 + 3 "
```

essayez en d'autres. Puis allez au paragraphe I7

Note : pour taper *et + , il faut appuyez sur la touche SHIFT en même temps que les touches *et +

- Paragraphe I7 :

Regardez chaque exemple et essayez de le résoudre en utilisant les règles de l'ordinateur. Puis tapez la commande à l'ordinateur, appuyez sur ENTER, et regardez comment l'ordinateur le résoud :

Si vous pensez que vous comprenez très bien les règles, essayez de résoudre les derniers exemples. Si vous le pouvez passer le reste:

```

PRINT 4 / 2 + 6
PRINT 3 * 2 + 1
PRINT 10 / 5 + 4 - 5
PRINT 8 * 2 - 6 * 2
PRINT (3 + 6) / (3 + 3)
print ( 10*4 -4) / (2*8 -4)
print 2*(6*2) - 20
PRINT 25 + (5 *5) / 5
PRINT (6 *4 + 4) / (3 + 4) -4
PRINT 100 / (10 + 10)*(2*4) - 1

```

LECON 2, PARTIE 1

- Paragraphe I8 :

Voici un programme. Chaque instruction commence par un nombre. Ici ces nombres sont 1,2 et 3 :

```

1 PRINT "DEBUT"
2 PRINT
3 PRINT "FIN"

```

- Paragraphe I9 :

Les commandes que vous avez utilisées auparavant sont exécutées dès que vous appuyez sur la touche "ENTER". Une instruction d'un programme est exécutée seulement quand la totalité du programme est exécutée.

- Paragraphe 20 :

```
1 PRINT " 2 + 2 = "
2 PRINT 2 + 2
```

RUN ← tapez RUN et appuyez sur ENTER

2 + 2 = ← l'ordinateur répond avec ces deux lignes
4

- Paragraphe 21 :

Regardez ce programme :

```
3 PRINT "BONJOUR "
4 PRINT
5 PRINT "AU REVOIR"
```

- Paragraphe 22 :

```
12 PRINT "C'EST ACTUELLEMENT LE MILIEU"
35 PRINT "C'EST ACTUELLEMENT LA FIN"
5 PRINT "C'EST LE DEBUT"
```

- Paragraphe 23 :

```
5 PRINT
10 PRINT "8 + 5 ="
15 PRINT 8 + 5
```

RUN ← première exécution

8 + 5 } réponse de l'ordinateur

13

RUN ← seconde exécution

8 + 5 =

13

} réponse de l'ordinateur

- Paragraphe 24 :

```

I5 PRINT I0 * 2
  5 PRINT
I0 PRINT "I0 * 2 = "
LIST ← commande LIST
  5 PRINT
I0 PRINT "I0 * 2 = "
I5 PRINT I0 * 2

```

} réponse de l'ordinateur

- Paragraphe 25 :

```

I0 PRINT "DEBUT"
30 PRINT
20 PRINT "STOP "

```

Vous devez connaître une commande qui exécute ce programme et une commande qui vous permet de voir le programme tapé dans l'ordre correct.

LECON 2, PARTIE 2

- Paragraphe 26 :

Un "message d'erreur" est imprimé par l'ordinateur pour vous dire qu'il a trouvé une erreur et qu'il ne peut continuer l'exécution du programme jusqu'à ce que l'erreur soit trouvée et corrigée.

```

TRS-80, LEVEL I : WHAT ?
TRS-80, LEVEL II : ? SN ERROR IN I0
SN ERROR veut dire : erreur de syntaxe, ou "mot incorrect".

```

- Paragraphe 27 :

Il y a une faute de frappe à la ligne 30 de ce programme. L'ordinateur imprimera un message d'erreur.

```

10 PRINT "DEBUT"
20 PRINT
30 PRINT
40 PRINT "FIN"
RUN

```

```
DEBUT
```

```
WHAT ?
```

```
30 P ? IRNT
```

```
ou
```

```
DEBUT
```

```
? SN ERROR IN 30
```

} réponse de l'ordinateur (LEVEL I)

} réponse de l'ordinateur (LEVEL II)

• Paragraphe 28

Voici un moyen de corriger une erreur :

Tapez le numéro de ligne et appuyez sur ENTER

```
10 PRINT "DEBUT"
```

```
20 PRINT
```

```
30 PRINT
```

```
40 PRINT "FIN"
```

```
30 ← tapez 30 et appuyez sur ENTER
```

```
LIST ← cette commande fait que l'ordinateur imprime à nouveau le programme.
```

```
10 PRINT "DEBUT"
```

```
20 PRINT
```

```
40 PRINT "FIN"
```

} réponse de l'ordinateur (la ligne est partie)

• Paragraphe 29 :

Un autre moyen de corriger une erreur : retaper la ligne

```
10 PRINT "DEBUT"
```

```
20 PRINT
```

```
30 PRINT
```

```
40 PRINT "FIN"
```

```

30 PRINT ← vous retapez la ligne
LIST
10 PRINT "DEBUT"
20 PRINT
30 PRINT ← nouvelle ligne
40 PRINT "FIN"
RUN
DEBUT
FIN

```

réponse de l'ordinateur
(l'erreur est corrigée)

le programme tourne

- Paragraphe 30 :

Regardez ce programme :

```

10 PRINT "DEBUT"
20 PRINT "2 + 2 ="
30 PRINT 2 + 2

```

- Paragraphe 31 :

Tout programme doit se terminer quelque part. Pour dire à l'ordinateur où est la fin, vous tapez un numéro de ligne et END.

Regardez cet exemple :

```

10 PRINT "PROBLEME DE MATHS"
20 PRINT "2 + 2 ="
30 PRINT 2 + 2
40 END

```

- Paragraphe 32 :

Une autre instruction qui arrête un programme est STOP.

Regardez les résultats utilisant STOP et les résultats utilisant END.

```

10 PRINT "1"
20 PRINT "2"
30 STOP
RUN
1
2
BREAK AT 30

```

réponse de l'ordinateur

```

10 PRINT "1"
20 PRINT "2"
30 END
RUN
1
2
READY

```

● Paragraphe 33 :

Entrez ces programmes et exécutez-les pour voir ce qui arrive.

```
10 PRINT "DEBUT"
20 PRINT "MILIEU"
30 PRINT "FIN"
40 END
```

exécutez ce programme.

```
10 PRINT "8 x 5 ="
20 PRINT 8 * 5
30 STOP
```

exécutez-le.

```
5 PRINT "MATHS AVEC 2"
10 PRINT "2 + 2 ="
20 PRINT 2 + 2
30 PRINT "2 - 2 ="
40 PRINT 2 - 2
50 PRINT "2 x 2 ="
60 PRINT 2 * 2
70 PRINT "2 / 2 ="
80 PRINT 2 / 2
90 PRINT "C'EST CELA"
100 STOP
```

exécutez-le.

ajoutez ce qui suit au dernier programme sans retaper les instructions ci-dessus.

```
8 PRINT "ADDITION"
25 PRINT "SOUSTRACTION"
45 PRINT "MULTIPLICATION"
65 PRINT "DIVISION"
```

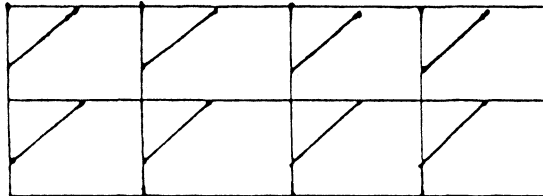
exécutez-le.

Note : les numéros de lignes sont tapés de 10 en 10 pour laisser de la place pour rajouter des instructions entre celles déjà tapées.

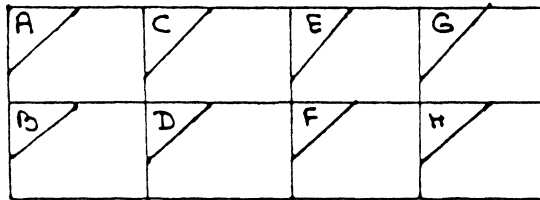
LECON 3, PARTIE 1 :

● Paragraphe 34 :

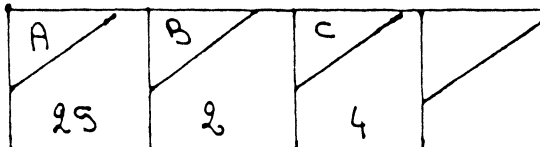
Boites aux lettres



● Paragraphe 35 :



● Paragraphe 36



● Paragraphe 37 :

Voici quelques exemples sur la façon d'utiliser les "boites mémoires" dans différents programmes :

A est l'age moyen d'un groupe : 14.2 A est une aire : 15.73

K est une distance en kilomètres : 200 L est une longueur : 2

D est le nombre d'erreurs : 6

La seule limitation à l'usage des boites est la possibilité de pouvoir mettre un nombre dans ces boites.

- Paragraphe 38 :

```

1Ø LET X = 6
2Ø PRINT X ← instructions
3Ø END
RUN
6 ← écriture de l'ordinateur

```

- Paragraphe 39 :

```

1Ø LET Y = 12
2Ø END
PRINT Y ← commande (sans numéro de ligne)
12 ← écritures de l'ordinateur

```

- Paragraphe 40 :

```

1Ø LET A = 25
2Ø PRINT A
3Ø END

```

- PARAGRAPHÉ, 41 :

```

1Ø LET N = - 8
2Ø PRINT N
3Ø END

```

- Paragraphe 42 :

```

1Ø LET Z = . 5
2Ø PRINT Z
3Ø END

```

- PARAGRAPHÉ 43 :

```

1Ø LET A = 1Ø
2Ø PRINT A + 5
3Ø END

```

- Paragraphe 44 :

```

1Ø LET A = 5
2Ø LET B = 8
3Ø PRINT A + B
4Ø END

```

- Paragraphe 45 :

```

10 LET P = I0
20 LET S = P
30 PRINT S * S
40 END

```

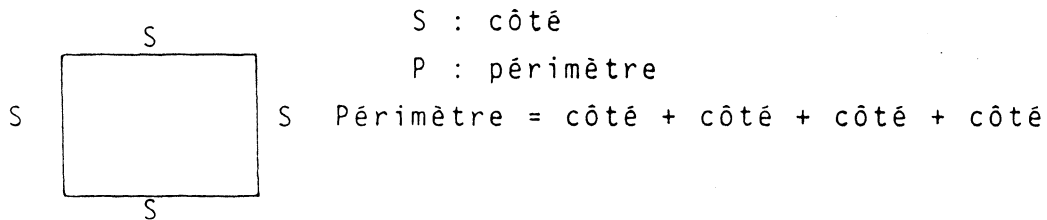
- Paragraphe 46 :

Les paragraphes 46 à 49 contiennent des programmes simples utilisant des variables. Entrez chaque programme et exécutez-le. Faites les changements suggérés et exécutez à nouveau. Tapez NEW avant de taper un programme.

Faites autant de ces paragraphes qu'il faut pour que vous vous sentiez à l'aise avec les variables.

Programme pour trouver le périmètre d'un carré :

Tapez NEW avant de commencer.



```

10 LET S = 5
20 LET P = S + S + S + S
70 PRINT P
80 END

```

Ajoutez les instructions avec une flèche pour faire un programme avec des réponses :

```

10 LET S = 5
20 LET P = S + S + S + S
—> 30 PRINT "PERIMETRE D'UN CARRE"
—> 40 PRINT "COTE = "
—> 50 PRINT S
—> 60 PRINT "PERIMETRE ="
70 PRINT P
80 END

```

Dans le programme ci-dessus, changez la ligne 10 et trouvez le périmètre d'un carré de côté 12; 15; un autre nombre....

- Paragraphe 47 :

Programme trouvant l'aire d'un carré



```

10 LET S = 14
20 LET A = S * S
30 PRINT "AIRE D'UN CARRE"
40 PRINT "COTE = "
50 PRINT S
60 PRINT "AIRE ="
70 PRINT A
80 END

```

Changez la ligne 10 et trouvez l'aire d'autres carrés.

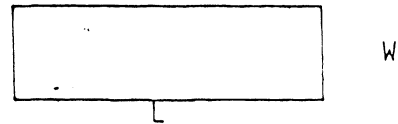
- Paragraphe 48 :

Programme trouvant l'aire d'un rectangle.

```

10 LET L = 3
20 LET W = 2
30 LET A = L * W
40 PRINT "AIRE D'UN RECTANGLE"
50 PRINT
60 PRINT "LONGUEUR ="
70 PRINT L
80 PRINT "LARGEUR ="
90 PRINT W
100 PRINT "AIRE ="
110 PRINT A
120 END

```



Changez les lignes 10 et 20 et trouvez d'autres aires.

- Paragraphe 49 :

Programme convertissant les kilomètres en miles

1 Kilomètre = 0.621 miles

```

10 LET K = 100
20 LET M = .621 * K
30 PRINT "KILOMETRES ="
40 PRINT K
50 PRINT "MILES"
60 PRINT M
70 END

```

Changez la ligne 10 pour convertir 60 kilomètres en miles.

LECON 3, PARTIE 2

- Paragraphe 50 :

```

10 PRINT "PROGRAMME TROUVANT L'AIRE D'UN CARRE"
20 LET S = 10
30 LET A = S * S
40 PRINT A
50 END

```

- Paragraphe 51 :

```

10 PRINT "PROGRAMME TROUVANT L'AIRE D'UN CARRE"
20 INPUT S ← vous demande de taper quelque chose
30 LET A = S * S
40 PRINT A
50 END

```

RUN

PROGRAMME TROUVANT L'AIRE D'UN CARRE

? 12 ← vous tapez 12 , S = 12

144 ← l'ordinateur imprime S * S pour A

- Paragraphe 52 :

```

10 PRINT "PROGRAMME POUR TROUVER L'INTERET DU PRINCIPE"
20 INPUT P
30 LET I = .05
40 PRINT I * P
50 END
RUN

```

```

PROGRAMME POUR TROUVER L'INTERET DU PRINCIPE } écritures de l'ordinateur
?

```

- Paragraphe 53 :

Programme 1 :	10 INPUT S	Programme 2 :	10 LET S = 15.8
	20 LET P = 4 * S		20 LET P = 4 * S
	30 PRINT P		30 PRINT P
	40 END		40 END

- Paragraphe 54 :

```

10 INPUT A,B
20 PRINT A + B
30 END

```

- Paragraphe 55 :

```

10 INPUT A,B
20 Print A + B
30 END

```

RUN

? 5,3 ← tapez deux nombres, séparés par une virgule puis appuyez sur ENTER.

8

- Paragraphe 56 :

```

5 PRINT "PROGRAMME TROUVANT L'AIRE D'UN CARRE"
8 PRINT "ENTREZ LA LONGUEUR DU COTE"
10 INPUT S
20 LET A = S * S
25 PRINT "AIRE ="
30 PRINT A
40 END

```

- Paragraphe 57 :

Les paragraphes 57 à 58 contiennent des programmes à essayer. Entrez ces programmes pour voir comment travaille l'instruction INPUT. Quand vous pensez avoir compris, chargez le questionnaire sur la leçon 3.

```
5 PRINT
10 PRINT "PROGRAMME ADDITIONNANT DEUX NOMBRES"
20 PRINT "ENTREZ DEUX NOMBRES"
30 INPUT A,B
40 PRINT "LA REPONSE EST"
50 PRINT A + B
60 END
```

Comme pratique, faites vos propres programmes pour :

- . soustraire deux nombres
- . multiplier deux nombres
- . diviser deux nombres
- . additionner trois nombres
- . multiplier trois nombres

- Paragraphe 58 :

```
10 PRINT "PROGRAMME CONVERTISSANT GALLONS EN LITRES"
20 PRINT "ENTREZ LE NOMBRE DE GALLONS"
30 INPUT G
40 LET L = 16.907 * G
50 PRINT "NOMBRE DE LITRES :".
60 PRINT L
70 END
```

- Paragraphe 59

```
10 PRINT
20 PRINT "PROGRAMME CONVERTISSANT LES PIEDS EN METRES"
30 PRINT "COMBIEN DE PIEDS ?"
40 INPUT F
50 LET M = .3048 * F
60 PRINT "NOMBRE DE METRES"
70 PRINT M
00 END
```

LECON 4, PARTIE 1 :

● Paragraphe 60 :

```

10 PRINT "CECI EST LE DEBUT"
20 PRINT "CECI EST LE MILIEU"
30 PRINT "CECI EST LA FIN"
40 END
GOTO 30 ← commande GOTO
CECI EST LA FIN ← écritures de l'ordinateur

```

● Paragraphe 61 :

```

10 LET A = 30
20 GOTO 50
30 PRINT "LE RESULTAT EST"
40 PRINT A * A
50 END

```

● Paragraphe 62 :

L'instruction GOTO peut faire boucler un programme

```

10 LET A = 5
20 PRINT A
30 GOTO 10

```

boucle

● Paragraphe 63 :

Le but de la boucle GOTO est de faire qu'un programme se répète.
Ce programme additionne les deux nombres qui sont entrés :

```

10 PRINT "EXEMPLE SIMPLE"
20 PRINT "ENTREZ DEUX NOMBRES"
30 INPUT A,B
40 PRINT "LA SOMME EST :"
50 PRINT A + B
60 GOTO 20

```

Boucle

● Paragraphe 64 :

L'instruction IF... THEN fait prendre des décisions à l'ordinateur

```

IF A = 30 THEN STOP

```

condition ce qui arrive c'est qu'il va l'ordinateur si la condition est vraie

- Paragraphe 65 :

```

10 INPUT A
20 PRINT A
30 IF A = 0 THEN STOP
40 GOTO 10

```

RUN

?3

3

?8

8

?3000

3000

?0

BREAK AT 30

quand ces nombres sont entrés, l'ordinateur les imprime

entrée 0

- Paragraphe 66 :

L'instruction IF-THEN peut stopper un programme répétitif. Ce programme s'arrêtera quand on entre - 1.

```

10 PRINT "PROGRAMME ADDITIONNANT DEUX NOMBRES"
20 PRINT "ENTREZ DEUX NOMBRES "
30 INPUT A,B
40 PRINT A + B
50 IF A = - 1 THEN END
60 GOTO 20

```

- Paragraphe 67 :

On peut utiliser d'autres instructions avec l'instruction IF-THEN

```

10 INPUT A
20 IF A = 1 THEN PRINT "OUI"
30 IF A = 2 THEN PRINT "NON"
40 IF A = 3 THEN PRINT "PEUT-ETRE"
50 END

```

- Paragraphe 68 :

Dans ce programme, si l'entrée est tout nombre sauf 1, l'ordinateur va à la ligne 30 et n'imprime rien.

```
10 INPUT X
20 IF X = 1 THEN PRINT "BONJOUR"
30 END
```

- Paragraphe 69 :

Exécutez ce programme et essayez de l'arrêter. Si vous avez oublié comment stopper une boucle, vous pouvez éteindre l'ordinateur!

```
10 LET A = 10
20 PRINT A
30 GOTO 10
40 END
```

Regardez ce qui arrive quand ce programme tourne.

```
10 GOTO 20
20 GOTO 10
30 END
```

Essayez ce programme

```
5 PRINT "TAPEZ UN NOMBRE"
10 INPUT A
20 IF A = 1 THEN PRINT "OUI"
30 IF A = 2 THEN PRINT "NON"
40 IF A = 3 THEN PRINT "PEUT-ETRE"
50 IF A = 4 THEN STOP
60 GOTO 5
```

LECON 4, PARTIE 2

- Paragraphe 70 :

```
IF A = 3 THEN GOTO 30
```

- Paragraphe 71 :

```

10 INPUT X
20 IF X = 1 THEN GOTO 50
30 IF X = 2 THEN GOTO 60
40 STOP
50 PRINT X
60 PRINT "FIN"
70 END

```

- Paragraphe 72 :

```

10 PRINT "ENTREZ 1 POUR LA PREMIERE ROUTINE"
20 PRINT "ENTREZ 2 POUR LA DEUXIEME ROUTINE"
30 INPUT A
40 IF A = 1 THEN 100
50 IF A = 2 THEN 200
60 ...
70 ...
100 ...
110 ...
200 ...
210 ...

```

- Paragraphe 73 :

```

10 INPUT A
20 IF A = 1 THEN 200
30 IF A = 2 THEN 400
40 IF A = 3 THEN 600
50 IF A = 4 THEN 900

```

```

10 INPUT A
20 ON A GOTO 200,400,600,900

```

- Paragraphe 74 :

Quand A est égal à 1 l'ordinateur va au premier nombre → 100. Quand A est égal à 2, il va à 200. Quand A est égal à 3, il va à 300.

```

10 INPUT A
20 ON A GOTO 100,200,300
100 PRINT "A= 1"
110 GOTO 10
200 PRINT "A =2"

```

.../...

```

210 GOTO I0
300 PRINT "A =3"
310 GOTO I0
RUN
?1
A =1
?3
A =3
etc...

```

écritures de l'ordinateur

- Paragraphe 75 :

```

I0 INPUT R
20 ON R GOTO I00,220,300,450,600

```

- Paragraphe 76 :

Jusqu'à maintenant, nous n'avons utilisé que le signe égal (=).
Voici d'autres signes que nous pouvons utiliser.

$X > Y$ veut dire que le nombre X est plus grand que le nombre Y
 $X < Y$ veut dire que le nombre X est plus petit que le nombre Y

- Paragraphe 77 :

```

I0 INPUT N
20 INPUT X
30 IF X > N THEN STOP
40 GOTO I0

```

- Paragraphe 78 :

Nous pouvons mélanger les signes $<$, $=$ et $>$:

$X <= Y$: X est plus petit ou égal à Y
 $X >= Y$: X est plus grand ou égal à Y
 $X <> Y$: X n'est pas égal à Y

- Paragraphe 79 :

```

I0 INPUT N
20 IF N <= I0 THEN STOP
30 GOTO I0

```

- Paragraphe 80 :

Entrez et exécutez ces programmes en essayant diverses entrées :

```

5 PRINT
10 PRINT " OU VA CE PROGRAMME ? ENTREZ UN NOMBRE"
20 INPUT A
30 ON A GOTO 100,200,300,400
40 PRINT "LE NOMBRE ETAIT TROP GRAND"
50 GOTO 5
100 PRINT "LIGNE 100"
110 GOTO 5
200 PRINT "LIGNE 200"
210 GOTO 5
300 PRINT "LIGNE 300"
310 GOTO 5
400 PRINT "LIGNE 400"
410 GOTO 5
500 END

```

Supprimez les lignes 40 et 50 et regardez ce qui arrive puis supprimez les lignes 110, 210 et 310

```

5 LET N = 5
10 PRINT "VOICI UN JEU DE DECOUVERTE DE NOMBRE"
20 PRINT "ESSAYER DE TROUVER LE NOMBRE ENTRE 1 et 10"
30 PRINT "VOTRE ESSAI ?"
40 INPUT G
50 IF G = N THEN 100
60 IF G <> N THEN PRINT "NON, ESSAYEZ ENCORE"
70 GOTO 30
100 PRINT "C'est CELA !"
110 END

```

```

10 PRINT "VOICI UN JEU DE DECOUVERTE D'UN NOMBRE POUR"
20 PRINT "DEUX PERSONNES. UNE PERSONNE ENTRE UN"
30 PRINT "NOMBRE SECRET ENTRE 1 ET 100 ET"
40 PRINT "EFFACE L'ECRAN .L'AUTRE PERSONNE."
50 PRINT "ESSAYE DE DEVINEZ LE NOMBRE ."
60 PRINT "ENTREZ LE NOMBRE SECRET"
65 INPUT N
70 PRINT "BEBUT DES ESSAIS..."
80 INPUT G
90 IF G = N THEN 200
100 IF G > N THEN PRINT "TROP GRAND, ESSAYEZ PLUS PETIT"
110 IF G < N THEN PRINT "TROP PETIT, ESSAYEZ PLUS GRAND"
120 GOTO 80
200 PRINT "C'EST CELA!"
210 END

```

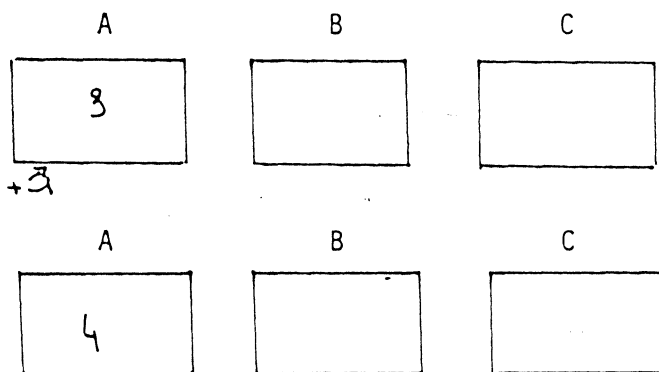
LECON 5, PARTIE 1

● Paragraphe 81 :

ancienne valeur de A

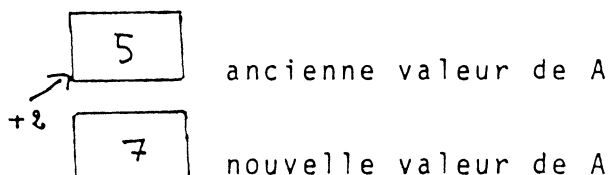
ajoutez 1 à A

résultat : nouvelle
valeur de A



● Paragraphe 82 :

Dans cette exemple , A était égal à 5. Puis nous y avons ajouté 2.
COMMANDE : LET A = A + 2



● Paragraphe 83 :

```

10 LET A = 4
20 LET A = A + 2
30 PRINT A
40 END

```

- Paragraphe 84 :

```

10 LET A = 4
20 LET A = (A + 3) / 7
30 PRINT A
40 END

```

- Paragraphe 85

```

5 LET A = 0
20 LET A = A + 1
25 PRINT A
30 GOTO 20

```

- Paragraphe 86 :

Quand le programme débute, A est mis égal à 0. Puis le programme ajoute 1 à A. La ligne 25 teste la valeur de A pour voir si A égal 5. Tant que ce n'est pas fait, le programme boucle à la ligne 10 et ajoute à nouveau 1 à A.

Il continue jusqu'à A = 5

```

5 LET A = 0
10 LET A = A + 1    ← compteur
20 PRINT A
25 IF A = 5 THEN END ← nouvelle ligne
30 GOTO 10

```

- Paragraphe 87 :

```

10 LET C = C + 1
20 PRINT C * C
30 IF C > 19 THEN END
40 GOTO 20

```

- Paragraphe 88 :

Chaque faute que fait l'utilisateur, augmente E de 1

```

10 PRINT "TROUVER UN NOMBRE ENTRE 1 ET 10"
20 LET N = 5
30 PRINT "DEBUT DES ESSAIS"
35 INPUT G
40 IF G <> N THEN LET E = E + 1
45 IF G <> N THEN PRINT "ESSAYEZ ENCORE"
50 IF G <> N THEN GOTO 35
60 IF G = N THEN PRINT "C'est CELA"
70 PRINT "VOUS AVEZ FAIT CE NOMBRE D'ERREURS"
80 PRINT E
90 END

```

- Paragraphe 89 :

La ligne 10 est une remarque ou instruction REM

```

10 REM CECI EST UN PROGRAMME SIMPLE
dit à l'interpreteur BASIC 20 PRINT 2 + 2
d'ignorer cette ligne 30 END
RUN
4 ← écriture de l'ordinateur

```

- Paragraphe 90 :

```

10 REM C'EST UNE INSTRUCTION REM
20 REM L'ORDINATEUR L'IGNORERA
30 PRINT "L'ORDINATEUR N'INGNORERA PAS CETTE INSTRUCTION"
40 END
RUN
L'ORDINATEUR N'IGNORERA PAS CETTE INSTRUCTION ← écriture de l'ordinateur
LIST
10 REM C'EST UNE INSTRUCTION REM
20 REM L'ORDINATEUR L'IGNORERA
30 PRINT "L'ORDINATEUR N'IGNORERA PAS CETTE INSTRUCTION"
40 END.

```


- Paragraphe 91 :

```

5 REM LA SUITE EST UN COMPTEUR
8 LET A = 0
10 LET A = A + 1
20 PRINT A
30 IF A = 10 THEN END
40 GOTO 10

```

- Paragraphe 92 :

```

5 REM PROGRAMME IMPRIMANT I A 10
10 B = 0
20 B = B + 1
30 IF B = 10 THEN END
40 PRINT B
50 GOTO 10

```

- Paragraphe 93 :

Qu'arrive-t-il quand ce programme tourne ?

```

10 LET A = A + 1
20 PRINT A
30 GOTO 10

```

Ce programme s'arrêtera de lui même :

```

10 LET A = A + 1
20 PRINT A
30 IF A = 20 THEN PRINT "NOUS SOMMES A LA MOITIE"
40 IF A = 40 THEN PRINT "C'EST FAIT"
50 IF A = 40 THEN END
60 GOTO 10

```

● Paragraphe 94 :

```
5 REM JEU DE DECOUVERTE D'UN NOMBRE AVEC COMPTEUR D'ERREURS
6 LET E = 0
10 PRINT "CECI EST UN JEU DE DECOUVERTE D'UN NOMBRE POUR DEUX"
20 PRINT "UNE PERSONNE ENTRE UN NOMBRE SECRET ENTRE"
30 PRINT " 1 ET 100 ET EFFACE L'ECRAN. L'AUTRE PERSONNE"
40 PRINT "ESSAYE DE LE DECOUVRIR. LE PROGRAMME"
50 PRINT "DONNE DES INDICATIONS ET COMPTE LES ERREURS"
60 PRINT "ENTREZ LE NOMBRE SECRET ET EFFACEZ L'ECRAN"
70 INPUT N
80 PRINT "ESSAYEZ DE TROUVER"
85 INPUT G
90 REM ESSAI JUSTE
100 IF G = N THEN 200
110 REM ESSAI FAUX
120 IF G > N THEN PRINT "TROP GRAND ESSAYEZ PLUS PETIT"
130 IF G < N THEN PRINT "TROP PETIT ESSAYEZ PLUS GRAND"
140 REM COMPTEUR D'ERREURS
150 IF G <> N THEN LET E = E + 1
160 GOTO 85
200 REM ESSAI CORRECT
210 PRINT "CORRECT !"
220 PRINT "NOMBRE D'ERREURS : "
230 PRINT E
235 LET E = 0
240 PRINT "VOULEZ VOUS JOUEZ DE NOUVEAU ? (1 = OUI, 0 = NON)"
250 INPUT A
260 IF A = 1 THEN 60
270 END
```

Paragraphe 95 :

```

10 REM PROGRAMME CALCULANT L'INTERET BANCAIRE
20 PRINT
30 PRINT "CECI EST UN PROGRAMME QUI CALCULE UN"
40 PRINT "INTERET COMPOSE. VOUS ENTREZ VOTRE"
50 PRINT "MONTANT LE TAUX D'INTERET ET LE NOMBRE"
60 PRINT "D'ANNEES"
70 PRINT " NOUS VOUS DONNONS LA BALANCE BANQUAIRE"
80 PRINT "A LA FIN DE LA PERIODE"
90 PRINT " ENTREZ 3 NOMBRES SEPARES PAR UNE VIRGULE"
95 REM P EST L'AVOIR, R EST LE TAUX D'INTERET"
96 REM ET Y EST LE NOMBRE D'ANNEES.
100 INPUT P,R,Y
110 REM AVOIR PLUS INTERET DONNE LE NOUVEL AVOIR
120 LET P = P + P * R
130 REM COMPTEUR DU NOMBRE D'ANNEES
135 LET A = A + 1
140 IF A = Y THEN 200
150 GOTO 120
200 PRINT "VOTRE AVOIR PLUS INTERETS A LA FIN DE"
210 PRINT "LA PERIODE EST:"
220 PRINT P
230 END

```

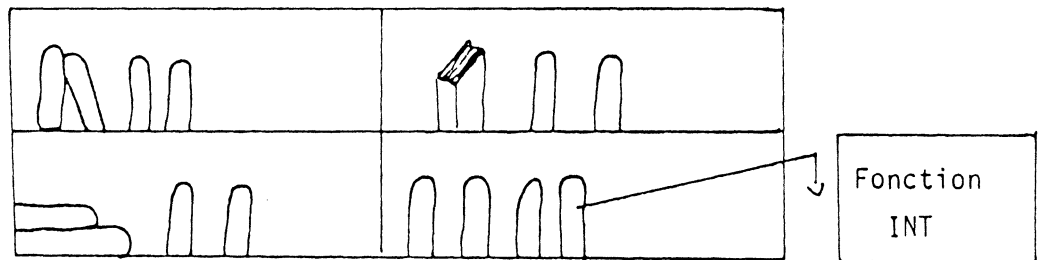
LECON 5, PARTIE 2

Paragraphe 96 :

INT	: Fonction INTeger (ENTIER)	: Convertit n'importe quel nombre en nombre entier.
ABS	: Fonction valeur ABSolue	: convertit n'importe quel nombre en un nombre positif
RND	: Fonction RaNDom (ALEATOIRE)	: choisit au hasard un nombre.

- Paragraphe 97 :

Ces instructions sont parfois nommées fonction de la librairie. Ces fonctions peuvent toutes être effectuées par le programme que vous écrivez. Au lieu d'écrire un programme, la librairie de l'interpreteur BASIC le fait à votre place.



- Paragraphe 98 :

La fonction INT convertit un nombre en entier.

INT (3.7) → 3 ← (3.7 devient un nombre entier et l'ordinateur l'imprime)

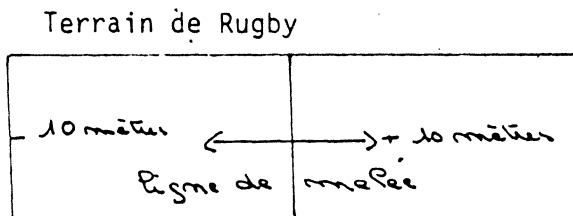
- Paragraphe 99 :

INT convertit un nombre comme 4.56 en un nombre entier en supprimant la partie décimale .56 et en gardant 4

PRINT INT (4.56)

4 ← l'ordinateur imprime le nombre sans le .56.

Paragraphe 100 :



Une flèche montre un gain de 10 mètres : + 10 l'autre flèche montre une perte de 10 mètres : - 10. Le signe d'un nombre montre dans quel sens on va. Quelques fois vous ne désirerez connaître que la taille du nombre, sans son signe. C'est la valeur absolue du nombre La valeur absolue de - 10 ou de + 10 est 10

- Paragraphe I01 :

La commande ou l'instruction ABS enlève le signe

```
PRINT ABS ( - 7 ) = 7 valeur absolue
      le nombre doit être entre ( ).
```

- Paragraphe I02 :

```
I0 LET A = ABS ( - I0 )
20 PRINT A
30 END
```

- Paragraphe I03 :

```
I0 LET A = ABS ( - 2.58 )
20 PRINT A
30 END
```

- Paragraphe I04 :

RND veut dire nombre aléatoire (en anglais: Random). Un nombre aléatoire est un nombre obtenu par le lancement d'une pièce, en battant un jeu de cartes, en tirant un nombre d'un chapeau, etc ... Ils sont tirés complètement au hasard.

- Paragraphe I05 :

Sur les différents interpréteurs BASIC, le générateur de nombres aléatoires travaille différemment. Voici 3 méthodes différentes utilisées pour imprimer un nombre aléatoire entre I et I00 :

Commodore PET

TRS-80

APPLE II

```
I0 LET N = RND (0)
```

```
I0 PRINT RND (I00)
```

```
I0 PRINT RND (I00) + 1
```

```
20 LET N = I00 * N
```

```
30 PRINT INT (N) + 1
```

- Paragraphe I06 :

Entrez et exécutez les programmes des paragraphes I06 à I08

```

I0 REM PROGRAMME DETERMINANT SI UN NOMBRE EST
20 REM PAIR OU IMPAIR
30 PRINT "CE PROGRAMME VOUS DIT SI UN NOMBRE"
40 PRINT "EST PAIR OU IMPAIR"
50 PRINT "ENTREZ VOTRE NOMBRE (UN NOMBRE ENTIER)"
60 INPUT N
70 REM SI N EST IMPAIR, N/2 SERA UNE FRACTION
80 IF INT (N/2) = N/2 THEN PRINT "PAIR"
90 IF INT (N/2) <> N/2 THEN PRINT "IMPAIR "
I00 END

```

- Paragraphe I07 :

Exécutez ce programme quelques fois pour voir différents résultats.

```

I0 REM PROGRAMME DE PILE OU FACE
20 LET P = 0
30 LET F = 0
40 PRINT "L'ORDINATEUR JETTE UNE PIECE 10 FOIS"
50 LET N = RND (2)
50 LET N = RND (0)
55 LET N = INT (2 * N) + 1
60 IF N = 1 THEN PRINT "PILE"
70 IF N = 1 THEN P = P + 1
80 IF N = 2 THEN PRINT "FACE"
90 IF N = 2 THEN F = F + 1
I00 IF P + F = I0 THEN I20
I10 GOTO 50
I20 PRINT "NOMBRE DE PILE"
I30 PRINT P
I40 PRINT "NOMBRE DE FACE"
I50 PRINT F
I60 END

```

- Paragraphe I08 :

```

I0 REM JEU DE DECOUVERTE D'UN NOMBRE AVEC NOMBRES ALEATOIRES
20 PRINT "L'ORDINATEUR VA PENSER A UN NOMBRE ENTRE"
30 PRINT "0 et 99. VOUS DEVEZ ESSAYER DE LE TROUVER"
40 REM L'ORDINATEUR PENSE A UN NOMBRE
50 LET N = RND (99)      { OU      50 LET N = RND (0)
                        {             60 LET N = INT (100 * N)
70 PRINT "OK, ESSAYEZ DE TROUVER"
80 INPUT G
90 REM ESSAI JUSTE
I00 IF G = N THEN 200
I10 REM ESSAI FAUX
I20 IF G > N THEN PRINT " TROP GRAND, ESSAYEZ PLUS PETIT"
I30 IF G < N THEN PRINT " TROP PETIT, ESSAYEZ PLUS GRAND"
I40 REM COMPTEUR D'ERREURS
I50 LET E = E + 1
I60 GOTO 70
200 REM ESSAI CORRECT
210 PRINT "CORRECT"
220 PRINT "NOMBRE D'ERREURS:"
230 PRINT E
235 LET E = 0
240 PRINT "VOULEZ-VOUS JOUER ENCORE ? (I = OUI, 0 = NON)"
250 INPUT A
260 IF A = 1 THEN 50
270 END

```

LECON 6, PARTIE 1 :

- Paragraphe I09 :

```

I0 PRINT "BONJOUR"
20 PRINT
30 PRINT "AU REVOIR"
40 END

```

- Paragraphe II0 :

```

10 PRINT "QUEL AGE AVEZ-VOUS?"
20 INPUT A
30 PRINT "DITES LA VERITE"
40 PRINT
50 PRINT
60 PRINT
70 GOTO 10
80 END

```

- Paragraphe III :

Quand l'interpreteur Basic rencontre une virgule dans une instruction PRINT, il place ce qui suit la virgule dans la "colonne" suivante de l'écran. Bien évidemment la virgule ne doit pas être entre guillemets. Car l'ordinateur imprimerait alors la virgule.

```

10 A = 1
20 B = 2
30 C = 3
40 PRINT A,B,C
50 END

```

RUN

```

1           2           3 ← écritures de l'ordinateur

```

- Paragraphe II2 :

```

10 A = 1
20 B = 2
30 PRINT "A" ; "B" ← imprime les lettres
40 PRINT A ; B ← imprime les valeurs de A et B

```

RUN

AB

```

1           2

```


- Paragraphe II3 :

10 PRINT "PORTE"	10 PRINT "PORTE" ;	← point virgule
20 PRINT "FEUILLE"	20 PRINT "Feuille"	
30 END	30 END	
RUN	RUN	
PORTE	} ← écriture de l'ordinateur →	
FEUILLE	PORTEFEUILLE	

- Paragraphe II4 :

```
10 PRINT 6,
20 PRINT 2
30 END
```

- Paragraphe II5 :

La commande ou l'instruction TAB travaille de la même façon que le TABulateur d'une machine à écrire. TAB dit à l'ordinateur combien d'espaces blancs il faut laisser sur une ligne.

```
PRINT TAB (15); "DEBUT" ← dit à l'ordinateur de laisser 15 espaces blancs
DEBUT sur la ligne
```

- Paragraphe II6 :

TAB est utilisé avec un point-virgule

```
10 PRINT TAB (20) ; "DEBUT"
20 END
RUN
```

```
DEBUT ← écritures de l'ordinateur
} espaces laissés par TAB
```

- Paragraphe II7 :

Exécutez ces commandes PRINT et regardez ce qui arrive

```
PRINT 5,6,7
PRINT 5;6;7
PRINT TAB (20) ; B
PRINT TAB (20) ; "8 + 5 ="
PRINT TAB (20) ; 8 + 5
PRINT "PREMIER"; TAB (20) ; "DERNIER"
```

- Paragraphe II8 :

Programmes simples

```
10 PRINT "LIGNE";
20 PRINT "ESPACE"
30 PRINT
40 PRINT "LIGNE",
50 PRINT "ESPACE"
60 PRINT
70 PRINT "LIGNE"
80 PRINT "ESPACE"
90 END

10 LET A = 5
20 LET B = 8
30 LET C = A + B
40 PRINT A; "+" ; B ; "=" ; C
50 END
```

- Paragraphe II9 :

```
10 PROGRAMME D'EXERCICES D'ADDITIONS
20 PRINT "FAITES LES ADDITIONS SUIVANTES"
30 PRINT
40 LET A = INT (10 * RND (0)) } OU { 40 LET A = RND (10)
50 LET B = INT (10 * RND (0)) }     { 50 LET B = RND (10)
60 PRINT A ; "+" ; B ; "="
70 INPUT C
80 IF C = A + B THEN PRINT "BIEN !"
90 IF C <> A + B THEN PRINT "FAUX"
110 PRINT "TAPEZ 0 POUR CONTINUER"
120 INPUT D
130 IF D=0 THEN 40
140 END
```

```

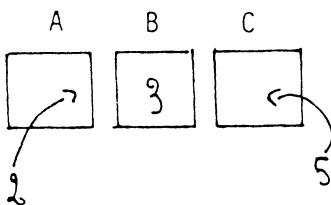
10 REM PROGRAMME D'ADDITIONS PLUS EVOLUE
20 PRINT "FAITES LES ADDITIONS SUIVANTES"
30 PRINT
40 LET A = INT (10*RND (0)) } ou { 40 LET A = RND (10)
50 LET B = INT (10*RND (0)) }      { 50 LET B = RND (10)
60 PRINT A ; "+" ; B ; "="
70 INPUT C
80 IF C = A + B THEN 100
90 PRINT "FAUX, ESSAYEZ ENCORE"
95 LET E = E + 1
98 GOTO 60
100 PRINT "EXACT !"
110 LET R = R + 1
120 REM VERIFIER QUE L'ETUDIANT A FAIT 10 PROBLEMES
130 IF E + R = 10 THEN 200
140 GOTO 40
200 PRINT "VOTRE SCORE EST" ; R ; " REPONSE JUSTES SUR 10."
210 END

```

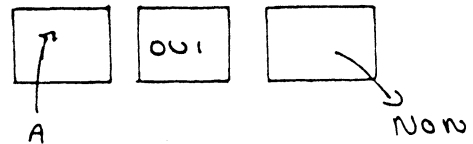
LECON 6, PARTIE 2

• Paragraphe 120 :

• nombres



mots



Paragraphe 121 :

A § contient une chaîne de caractères. Aussi on l'appelle une variable de chaîne.

A §

- Paragraphe I22 :

```

10 LET A$ = "CHIEN"
20 PRINT A$
30 END

```

- Paragraphe I23 :

```

10 LET A$ = "ON NE PASSE PAS"
20 PRINT A$
30 END
RUN
ON NE PASSE PAS ← écritures de l'ordinateur

```

- Paragraphe I24 :

```

10 INPUT A$
20 PRINT A$
30 END
RUN
? HAUT ET BAS ← entrée de l'utilisateur
HAUT ET BAS ← écriture de l'ordinateur

```

- Paragraphe I25 :

```

10 LET A$ = "VRAI"
20 LET B$ = "FAUX"
30 PRINT "QUE FAIT 3 X 3 ?"
40 INPUT N
50 IF N = 9 THEN PRINT A$
60 IF N <> 9 THEN PRINT B$
70 END

```

Paragraphe 126 :

Ces instructions comparent deux variables-chaines lettres par lettres pour voir si elles sont identiques. Dans la première instruction, l'ordinateur imprime VRAI si elles le sont.

Dans la deuxième, l'ordinateur va à I00 si elles ne le sont pas. Dans la troisième, si A\$ est OUI l'ordinateur imprime VRAI

- 1) IF A\$ = W\$ THEN PRINT "VRAI"
- 2) IF C\$ = T\$ GOTO I00
- 3) IF A\$ = "OUI" PRINT "VRAI"

Paragraphe 127 :

```

10 LET A$ = "CHIEN"
20 PRINT "QUEL EST VOTRE ANIMAL FAVORIT ?"
30 INPUT F$
40 IF A$ = F$ PRINT "LE MIEN AUSSI!"
50 IF A$ = F$ PRINT "VRAIMENT ?"
60 END

```

Comment vient-il ?

Paragraphe 128 :

Dans le programme suivant, expérimentez les choses qui peuvent être tapées dans une variable de chaîne depuis le clavier :

```

10 INPUT A$
20 PRINT
30 PRINT "VOICI CE QUE VOUS AVEZ TAPE DANS LA VARIABLE DE CHAINE:"
40 PRINT A$
50 END

```

Paragraphe 129 :

Essayez ces deux programmes :

```
10 REM REPONSES STUPIDES A DES ENTREES.  
20 PRINT "QUEL EST VOTRE NOM ?"  
30 INPUT A$  
40 PRINT  
50 PRINT A$ ;"! QUEL JOLI NOM!"  
55 PRINT  
60 PRINT "OU HABITEZ-VOUS ?"  
70 INPUT B$  
80 PRINT B$ ;"? J'AI UN AMI QUI VIT A " ; B$ ;"."  
90 PRINT "OK," ; A$ ;" RETOURNEZ A CE QUE VOUS FAISIEZ"  
95 PRINT "A" ; B$ ; "."  
100 END
```

LECON 7, PARTIE 1 :

● Paragraphe 130 :

Notez que l'on utilise une routine similaire à celle-là dans PAS-A-PAS pour arrêter la programme pendant que vous lisez quelque chose

```
I000 PRINT
I010 PRINT "TAPEZ Ø ET APPUYEZ SUR LA TOUCHE RETURN"
I020 INPUT A
I030 PRINT
```

● Paragraphe 131 :

Dans ce programme, on utilise la même routine plusieurs fois, aussi on la met à la fin. Maintenant c'est une sous-routine (en anglais : subroutines)

```
I0 ....
20 ....
30 IF A = 1 THEN PRINT "OUI"
40 ....
50 ....
60 IF A = 1 THEN PRINT "OUI"
70 ....
80 END
I000 IF A = 1 THEN PRINT "OUI"
```

← routine mise hors du programme

← sous routine écrite une seule fois

● Paragraphe 132 :

Quand l'ordinateur rencontre l'instruction GOSUB à la ligne 40, il va à la ligne I000

```
I0 ...
20 ...
30 ...
40 ... GOSUB I000
45 ...
50 ...
```

```

55 ...
60 ...
70 ...
80 END
I000 PRINT
I010 PRINT "TAPEZ 0 ET APPUYEZ SUR ENTER"
I020 INPUT A
I030 PRINT

```

programme régulier

sous-routine

● Paragraphe 133 :

Bien sûr , une fois que l'ordinateur est allé dans la sous-routine, vous avez besoin d'un moyen de le faire revenir dans le programme et le continuer là où vous l'aviez laissé.

Une instruction RETURN fait retourner l'ordinateur à l'instruction qui suit l'instruction GOSUB

Programme

sous-routine

```

10 ...
20 ...
30 ...
40 GOSUB 1000
50 ...
60 ...

```

```

I000 PRINT
I010 PRINT "TAPEZ 0 ET APPUYEZ SUR ENTER"
I020 INPUT A
I030 PRINT
I040 RETURN ← revoir l'ordinateur au
                programme principal

```

● Paragraphe 134 :

Voici comment fonctionne l'instruction RETURN

```

10 ...
20 ...
30 ...
40 GOSUB I000
45 ...
50 ...

```

```

I000 PRINT
I010 PRINT "TAPEZ 0 ET APPUYEZ SUR ENTER"
I020 INPUT A
I030 PRINT
I040 RETURN

```

The diagram shows two columns of code. The left column contains lines 10 through 50. The right column contains lines I000 through I040. A curved arrow starts from the right side of line 40 (GOSUB I000) and points to the right side of line I000 (PRINT). Another curved arrow starts from the right side of line I040 (RETURN) and points to the left side of line 45 (...), indicating the return path.

● Paragraphe 135 :

Regardez ce programme pour répondre aux quelques questions suivantes

```
10 REM SIMPLE PROGRAMME DE SOUS-ROUTINE
20 PRINT "PREMIER PASSAGE VERS LA SOUS-ROUTINE"
30 GOSUB 1000
40 PRINT "SECOND PASSAGE"
50 GOSUB 1000
60 PRINT "TROISIEME PASSAGE"
70 GOSUB 1000
80 STOP
1000 PRINT
1010 PRINT I ; 2 ; 3 ; 4 ; 5
1020 PRINT
1030 RETURN
1040 END
```

● Paragraphe 136 :

Voici la fin d'un programme d'exercice d'addition

```
110 PRINT "ADD 2 + 2"
120 INPUT S
130 LET A = 4
140 GOSUB 180
150 PRINT "ADD 3 + 3"
160 INPUT S
170 LET A = 6
175 GOSUB 180
180 IF S = A PRINT "VRAI"
190 IF S <> A PRINT "FAUX"
200 RETURN
210 PRINT " ALLER A LA LECON SUIVANTE".
```

- Paragraphe 137 :

```

10 REM DEMONSTRATION DE GOSUB
20 PRINT "ALLER A LA SOUS-ROUTINE"
30 GOSUB 1000
40 PRINT "ENCORE"
50 GOSUB 1000
60 PRINT "TROISIEME FOIS"
70 GOSUB 1000
80 STOP
1000 PRINT "VOICI LA SOUS-ROUTINE"
1010 PRINT "
1020 PRINT
1030 RETURN

```

exécutez le. Puis supprimez la ligne 180 et exécutez le.

- Paragraphe 138 :

```

10 REM PROGRAMME D'ARITHMETIQUE AVEC SOUS-ROUTINE
20 PRINT "QUEL EST VOTRE NOM ?"
30 INPUT A$
40 PRINT
50 PRINT A$ ; ", FAITES LES MULTIPLICATIONS SUIVANTES"
60 LET A = INT (10 * RND (0))    OU    60 A = RND (10)
70 LET B = INT (10 * RND (0))    70 B = RND (10)
80 PRINT A ; "X" ; B ; "=" ;
90 INPUT C
100 IF C = A * B THEN 140
110 PRINT "FAUX. ESSAYEZ ENCORE"
120 LET E = E + 1
130 GOTO 80
140 GOSUB 1000
150 REM VERIFIER SI L'ETUDIANT A FAIT 10 PROBLEMES
160 IF E + R = 10 THEN 200
170 GOTO 60
200 PRINT A$ ; ", VOTRE SCORE EST DE " ; R ; "JUSTES SUR 10"
210 END
1000 REM SOUS-ROUTINE
1010 LET Q = INT (3 * RND (0))    OU    Q = RND (3)
1020 IF Q = 0 THEN PRINT "CORRECT"

```

```

I030 IF Q = 1 THEN PRINT "C'EST CELA,"; A8
I040 IF Q = 2 THEN PRINT "EXCELLENT"
I050 LET R = R + 1
I060 PRINT
I070 RETURN

```

LECON 7, PARTIE 2 :

- Paragraphe 139 :

Ce programme a une instruction READ et une instruction DATA

```

I0 READ X
20 PRINT X
30 GOTO I0
40 DATA 1,2,3,4,5,6

```

- Paragraphe 140 :

Voici ce qui arrive : l'ordinateur prend le premier nombre dans l'instruction DATA et le met dans la variable de l'instruction READ

```

I0 READ X
20 PRINT X
30 GOTO I0
40 DATA 1,2,3,4,5,6
RUN
1 ← première fois, voici X

```

- Paragraphe 141 :

```

I0 READ X
20 PRINT X
30 GOTO I0
40 DATA 8,5,7,I0

```

● Paragraphe 142 :

```

10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,2,3,4,5,6

```

```

RUN

```

```

1
2
3
4
5
6

```

} l'ordinateur imprime X

OUT OF DATA ← l'ordinateur imprime un message d'erreur

● Paragraphe 143 :

```

10 READ A
20 PRINT A * A
30 GOTO 10
40 DATA 3,2,8

```

● Paragraphe 144 :

Ici RESTORE fait que l'ordinateur utilise les données (DATA) à nouveau

```

10 READ Y
20 PRINT Y
30 IF Y = - 1 THEN RESTORE
40 GOTO 10
50 DATA 1,2,3,- 1

```

```

RUN

```

```

1
2
3
- 1
1
2
3

```

```

etc...

```

← données reprises ici

} écritures de l'ordinateur

- Paragraphe 145 :

```
5 DATA 8,3,6,4,0
10 READ X
20 PRINT X
30
40 GOTO 10
```

- Paragraphe 146 :

Les nombres placés dans DATA doivent être séparés par une virgule.

```
10 DATA 6,8,3
20 READ A
30 IF A = 3 THEN RESTORE
```

- Paragraphe 147 :

Ici l'ordinateur lit un nombre pour X et le suivant pour Y à chaque fois

```
10 READ X,Y
20 PRINT X,Y
30 GOTO 10
40 DATA 1,2,3,4,5,6
RUN
```

1	2	} l'ordinateur imprime X,Y chaque fois
3	4	
5	6	

OUT OF DATA ERROR ← l'ordinateur n'a plus de données

- Paragraphe 148 :

```
5 REM EXEMPLES D'ADDITIONS
10 DATA 1,3,2,3,2,1,1,2,3,1
20 READ A,B,C
30 PRINT A;"+"; B ;" +" ; C ; "="
40 PRINT A + B + C
50 GOTO 20
```

- Paragraphe 149 :

```

10 PRATIQUE DE READ ... DATA
20 READ A
30 PRINT A
40 GOTO 10
50 DATA 5,6,7,8

```

exécutez le programme précédent, puis ajoutez la ligne suivante :

```
35 IF A = 8 THEN RESTORE
```

Puis ajoutez les lignes suivantes :

```

5 LET B = 0
36 LET B = B + 1
37 IF B = 8 THEN STOP

```

Ajoutez la ligne suivante :

```
6 DATA 1,2,4,5
```

Ajoutez les lignes suivantes :

```

20 READ X,Y
30 PRINT X,Y

```

- Et effacez les lignes 35,36 et 37

- Paragraphe 150 :

```

10 REM PROGRAMME D'EXERCICES
20 PRINT "FAITES LES DIVISIONS SUIVANTES :"
30 PRINT
40 READ A,B,C
50 PRINT A ; "/" ; B ; "=" ;
60 INPUT D
70 IF D<>C THEN PRINT "NON. LA REPONSE EST "; C
80 DATA .1,.2,.5,.8,.2,.4,.9,1.2,.75

```

```

90 DATA .7,.2,3.5
100 IF D = C THEN PRINT "CORRECT!"
105 IF C = 3.5 THEN END
110 GOTO 40
120 END

```

LECON 8, PARTIE 1 :

● Paragraphe 151 :

```

10 REM BOUCLE DE COMPTAGE
15 LET A = 0
20 PRINT A
30 LET A = A + 1
40 IF A = 11 THEN END
50 GOTO 20

```

• Programme 152 :

```

10 FOR A = 0 TO 10

```

● Paragraphe 153

Donne une valeur à A
et démarre

```

10 FOR A = 0 TO 10
20 NEXT A

```

retourne pour la valeur suivante
de A

● Paragraphe 154 :

Ce programme a une instruction PRINT placée entre les instructions FOR et NEXT. Chaque fois que le programme boucle, l'instruction PRINT imprime la nouvelle valeur de A.

```

10 FOR A = 0 TO 10
20 PRINT A ;
30 NEXT A

```

(; fait que l'impression est sur une seule ligne

- Paragraphe 155 :

```

10 FOR X = 1 TO 7
20 PRINT " * " ;
30 NEXT X
40 END

```

- Paragraphe 156 :

```

10 FOR Y = 10 TO 50 STEP 10
20 PRINT Y
30 NEXT Y
40 END
RUN

```

```

10 }- pas de 10
20 }
30
40
50

```

écritures de l'ordinateur

- Paragraphe 157 :

```

10 FOR Z = 2 TO 12 STEP 2
20 PRINT Z ;
30 NEXT Z
40 END

```

- Paragraphe 158 :

```

10 FOR A : 10 TO 1 STEP - 1
20 PRINT A
30 NEXT A
40 END

```


● Paragraphe 159 :

```

10 REM SIMPLE PROGRAMME FOR ... NEXT
20 FOR A = 1 TO 10
30 PRINT A
40 NEXT A
50 END

```

Ajoutez cette ligne au programme précédent :

```
35 PRINT A * A
```

Ajoutez cette ligne :

```
45 PRINT "FIN"
```

● Paragraphe 160 :

```

10 REM PROGRAMME COMPTE A-RECOURS
20 FOR X = 10 TO 0 STEP - 1
30 PRINT X
40 NEXT X
50 PRINT "***** BOUM*****"

```

← note : ajoutez une instruction effaçant l'écran (CLS sur le TRS 80)

Ajoutez cela au programme précédent,

```

35 FOR Y = 1 TO 500 } boucle de temporisation
36 NEXT Y

```

LECON 8, PARTIE 2 :

● Paragraphe 161 :

UN TABLEAU

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)

Paragraphe 162 :

Voici un tableau construit pour 30 jours. On a mis 50 dans chaque variable pour représenter 50 tickets :

A(1) / 50	A(2) / 50	A(3) / 50	A(4) / 50	A(5) / 50	A(6) / 50	A(7) / 50
A(8) / 50	A(9) / 50	A(10) / 50	A(11) / 50	A(12) / 50	A(13) / 50	A(14) / 50
A(15) / 50	A(16) / 50	A(17) / 50	A(18) / 50	A(19) / 50	A(20) / 50	A(21) / 50
A(22) / 50	A(23) / 50	A(24) / 50	A(25) / 50	A(26) / 50	A(27) / 50	A(28) / 50
A(29) / 50	A(30) / 50					

Paragraphe 163 :

A(1) / 3	A(2) / 20	A(3) / 15	A(4) / 6	A(5) / 0	... etc.
-------------	--------------	--------------	-------------	-------------	----------

- Paragraphe 164 :

```

5 LET X = 0
10 LET X = X + 1
20 LET A (X) = 5
30 IF X = 10 THEN STOP
40 GOTO 10

```

- Paragraphe 165 :

```

10 FOR X = 1 TO 30
20 LET Z (X) = 50
30 NEXT X
40 END

```

- Paragraphe 166 :

```

10 FOR X = 1 TO 30
20 LET A (X) = 50
30 NEXT X

```

5 DIM A(30) à ajoutez si vous utilisez le basic
LEVEL II

construit un tableau avec 50 dans chaque variable

```

40 PRINT "POUR QUEL JOUR VOULEZ-VOUS UN TICKET ?"
50 INPUT X
60 IF A(X) = 0 PRINT "DESOLE. PLUS DE TICKET. ESSAYEZ UN AUTRE JOUR"
70 IF A(X) <> 0 GOTO 90
80 GOTO 40
90 PRINT "IL Y A" ; A(X) ; "TICKETS"
100 PRINT "VENDEZ UN TICKET POUR CE JOUR"
110 LET A(X) = A(X) -1
120 GOTO 40

```

- Paragraphe 167 :

Simple programme :

```

10 PRINT "VOICI UN SYSTEME DE RESERVATION POUR"
20 PRINT "LA L.W.S .AIRWAYS. NOUS AVIONS ONT"

```

```

30 PRINT "4 PLACES. VOUS POUVEZ RESERVER SUR LES VOLS"
40 PRINT "1 A 7. DESOLE : TOUS LES VOLS SECONDE CLASSE"
50 REM SIEGES PAR AVIONS
60 FOR X = 1 TO 7
70 LET A(X) = 4
80 NEXT X
90 PRINT "QUEL VOL VOULEZ-VOUS ? (1 A 7)"
100 INPUT X
110 IF A(X) = 0 print "DESOLE, LE VOL EST COMPLET. ESSAYEZ UN AUTRE..."
120 IF A(X) <> 0 THEN I40
130 GOTO 90
140 REM LOCATION D'UN SIEGE
150 LET A(X) = A(X) - 1
160 PRINT "OK .. VOICI UN TIKET POUR CE VOL!"
170 GOTO 90

```

LECON 9, PARTIE 1 :

- Paragraphe 168 :

```

10 FOR A = 1 TO 3
20 PRINT "*****"
30 NEXT A
40 END

```

- Paragraphe 169 :

```

10 FOR A = 1 TO 3
20 PRINT "1*****"
30 PRINT "2*****"
40 PRINT "3*****"
50 PRINT "4*****"
60 PRINT "5*****"
70 NEXT A
80 END
RUN

```

```

1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *

```

à chaque boucle , voici ce que l'ordinateur imprime

● Paragraphe 170 :

```

10 FOR A = 1 TO 3
20 FOR Z = 1 TO 5
30 PRINT Z, " * * * * *"
40 NEXT Z
50 NEXT A
60 END
RUN

```

```

1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *

```

l'ordinateur imprimé cela 3 fois

● Paragraphe 171 :

● Paragraphe 172 :

```

10 FOR A = 1 TO 10
20 FOR B = 1 TO 5
30 NEXT B
40 NEXT A

```

boucle interne

boucle externe

- Paragraphe 173

Il y a une erreur dans la boucle 2 :

```

10 FOR A = 1 TO 100
20 FOR Z = 1 TO 20
30 PRINT Z
40 NEXT A
50 NEXT Z
RUN

```

1

1

1

...

WHAT ?

50 NEXT Z ?

} !'ordinateur imprime la première valeur de Z
100 fois puis vous obtenez un message d'erreur.

- Paragraphe 174 :

Programme 1

```

10 FOR X = 1 TO 10
20 FOR Y = 3 TO 1 STEP -1
30 PRINT Y
40 NEXT Y
50 NEXT X

```

Programme 2

```

10 FOR A = 1 TO 10
20 FOR B = 1 TO 3
30 PRINT B
40 NEXT A
50 NEXT B

```

- Paragraphe 175 :

```

10 FOR P = 1 TO 10
20 PRINT "PROBLEME"
30 FOR N = 1 TO 2
40 PRINT "DONNEZ MOI UN NOMBRE"
50 INPUT A(N)
60 NEXT N
70 PRINT A(1) ; "X" ; A(2) ; "=" ; A(1)*A(2)
80 NEXT P
90 END

```

● Paragraphe 176 :

D'abord entrez et exécutez le programme que vous venez de voir au programme I75. Puis essayez celui-ci :

```

10 CLS      (efface l'écran)
20 PRINT "COMBIEN DE PERSONNES DANS VOTRE FAMILLE ?"
30 INPUT N
40 FOR C = 1 TO N
50 PRINT "QUEL AGE A LA PERSONNE"; C
60 INPUT A(C)
70 NEXT C
80 PRINT "VOICI LE GRAPHE DE VOS AGES : "
90 FOR C = 1 TO N
100 PRINT "PERSONNE"; C
110 FOR X = 1 TO A(C)
120 PRINT "x";
130 NEXT X
140 PRINT
150 NEXT C

```

Ajoutez les lignes 160 et 170 et exécutez de nouveau :

```

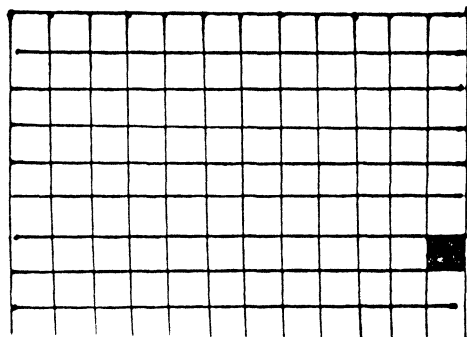
160 FOR X = 1 TO 5000 }
170 NEXT X             } boucle de temporisation maintenant l'affichage

```

LECON 9, PARTIE 2

● Paragraphe 177 :

L'écran d'un ordinateur est divisé en carré comme ce morceau de papier graphique.



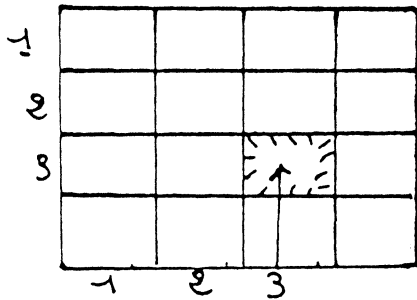
ce carré est "on"

Chaque carré peut-être allumé ("ON") ou éteint ("OFF"). Allumer ou éteindre des carrés permet de faire des dessins appelés graphiques de l'ordinateur.

●Paragraphe 178 :

Le nombre X vous dit à quelle colonne de l'écran aller
le nombre Y vous dit à quelle ligne

Voici l'adresse X = 3 (colonne) et Y = 2 (ligne) sur deux écrans



●Paragraphe 179 :

Ces deux programmes vont tous les deux allumer le même carré.

```
1 GR
5 COLOR = 6
10 X = 10
20 Y = 20
30 PLOT X,Y
```

```
1 GR
5 COLOR = 6
10 PLOT 10,20
```

Peut être utilisé avec différentes valeurs de X et de Y.

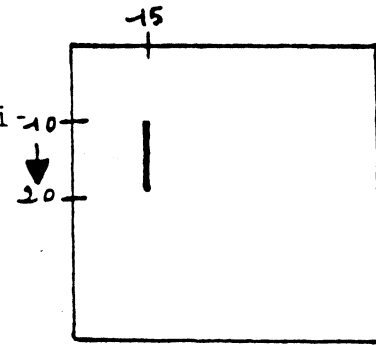
●Paragraphe 180

```
1 GR
5 FOR C = 0 TO 15
10 COLOR = C
20 PLOT 20,20
30 NEXT C
```


● Paragraphe 181 :

L'instruction VLIN dessine une ligne verticale. Les deux premiers nombres indiquent les deux extrémités de la ligne.

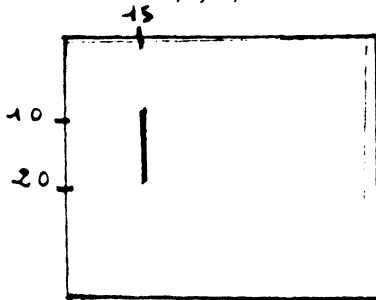
Le "AT" indique la position verticale de la ligne sur l'écran.



VLIN 10,20 AT 15

● Paragraphe 182 :

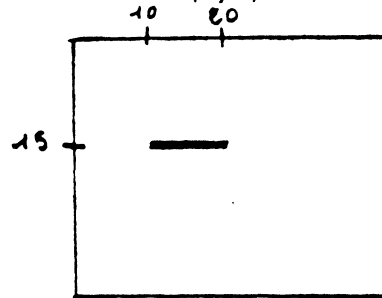
Ligne verticale
VLIN 10,20 AT 15



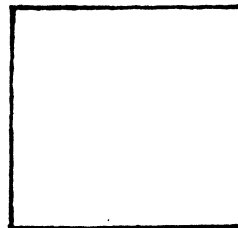
Paragraphe 183 :

```
10 GR
15 COLOR = 8
20 FOR X=10 TO 20
30 VLIN 5,25 AT X
40 NEXT X
```

Ligne horizontale
HLIN 10,20 AT 15



Ce programme dessine des lignes verticales qui forment une surface.



● Paragraphe 184 :

Programme 1

```
1 GR
5 COLOR = 8
10 FOR X=0 TO 39
20 VLIN 0,39 AT X
```

Programme 2

```
1 GR
5 COLOR = 8
10 FOR X=0 TO 39
20 HLIN 0,39 AT X
```

30 NEXT X	30 NEXT X
40 FOR T=1 TO 1000	40 FOR T=1 TO 1000 (BOUCLE D'ATTENTE)
50 NEXT T	50 NEXT T
60 TEXT	60 TEXT }*
70 HOME	70 HOME }
80 GOTO 1	80 GOTO 1

*Ces deux instructions effacent les graphiques et nettoient l'écran.

●Paragraphe 185 :

Entrer ce programme et essayez differentes valeurs de X et de Y pour voir où ce trouve l'emplacement du carré.

```

5 GR
10 COLOR=15
15 PRINT "TAPEZ DES VALEURS POUR X ET Y SEPARÉS PAR UNE VIRGULE."
20 INPUT X,Y
30 PLOT X,Y
40 PRINT"APPUYEZ SUR RETURN POUR CONTINUER"
50 INPUT A$
60 GOTO 15

```

●Paragraphe 186 :

Entrez ce programme et exécutez le pour voir ce qu'il se passe.

```

5 GR
10 COLOR=15
20 X=INT (40*RND (1))
30 Y=INT (40*RND (1))
40 PLOT X,Y
50 GOTO 20

```

Changez la ligne 10 ~~celle~~ ci-dessous et essayez de deviner ce qui va arriver.

```

10 COLOR=INT (16*RND(1))

```

Paragraphe 187 :

```

1 REM PROGRAMME POUR DESSINER DES LIGNES HORIZONTALES
5 GR
1Ø FOR C=1 TO 15
2Ø COLOR=C
3Ø FOR X=Ø TO 39 STEP 3
4Ø HLIN Ø,39 AT X
5Ø NEXT X
6Ø NEXT C
7Ø END

```

Changez le programme ci-dessus pour tracer des lignes verticales.
Puis changez le programme pour tracer des lignes et les effacer
avant d'en tracer de nouvelles.

(NOTE : COLOR=Ø FAIT DISPARAITRE UN CARRE)

Si vous ajoutez la ligne ci-dessous au programme ci-dessus, vous cons-
tituerez une grille:

```

45 VLIN Ø,39 AT X

```

Paragraphe 188,189 :

```

1 REM PROGRAMME POUR TRACER UN RECTANGLE
1Ø GR
15 COLOR=13
2Ø FOR X=Ø TO 39
3Ø HLIN Ø,39 AT X
35 NEXT X
4Ø REM MAINTENANT NOUS TRACONS LE CADRE
45 COLOR=2
5Ø HLIN Ø,39 AT Ø
5Ø HLIN Ø,39 AT 39
6Ø VLIN Ø,39 AT Ø
6Ø VLIN Ø,39 AT 39

```

Lancez le programme ci-dessus, puis ajoutez lui les instructions suivantes

```

70 REM
71 REM ALLUME OU ETEINT
75 FOR X=0 TO 39
80 COLOR=0
90 PLOT X,20
100 REM BOUCLE D'ATTENTE
105 FOR D=1 TO 50
110 NEXT D
120 COLOR=13
130 PLOT X,20
140 NEXT X
150 REM 80 ET 90 ETEIGNENT LE POINT
151 REM 120 ET 130 LE RALLUME

```

Lancez le programme ci-dessus puis lui rajoutez les instructions suivantes

```

150 REM LE POINTILLE SE REMET A REBONDIR
155 FOR X=39 TO 0 STEP -1
160 COLOR=0
170 PLOT X,0
180 FOR D=1 TO 50
190 NEXT D
200 COLOR=13
210 PLOT X,20
220 NEXT X

```

LECON 10, PARTIE 1

Paragraphe 190

Dans le second programme ; on a mis 3 instructions sur la même ligne.
Les deux programmes produisent le même effet.

Programme 1

```

10 FOR A=1 TO 3
20 PRINT " "
30 NEXT A

```

Programme 2

```

10 FOR A=1 TO 3: PRINT " " :NEXT A

```

- Paragraphe 191 :

```

10 PRINT "AVEZ-VOUS PLUS DE 18 ANS ? 1 = OUI , 0 = NON"
20 INPUT A : IF A = 1 THEN 30 : IF A = 0 THEN 40
30 PRINT "VOUS ETES ASSEZ AGE POUR VOTER ." GOTO 50
40 PRINT "VOUS NE POUVEZ PAS VOTER"
50 END

```

```

RUN
AVEZ-VOUS PLUS DE 18 ANS ? 1 = OUI , 0 = NON
? 0
VOUS NE POUVEZ PAS VOTER

```

} écritures de l'ordinateur

- Paragraphe 192 :

Dans ce programme, la flèche montre la voie que l'ordinateur suit si le test A = 1 n'est pas vérifié. Si A est égal a 1 , alors l'ordinateur exécute le reste de la ligne.

```

105 REM ROUTINE DE VERIFICATION DE REPONSE
110 INPUT A : IF A = 1 PRINT "EXACT" GOTO 130
120 PRINT "FAUX"
130 END

```

- Paragraphe 193 :

Programme 1

```

10 READ A,B : DATA 1,2,3,4
20 PRINT A*B : IF B = 4 THEN STOP
30 GOTO 10

```

Programme 2

```

10 PRINT "LISTE DE 5 NOMS"
20 C = 0 INPUT A : PRINT A
30 IF C = 5 GOTO 50
40 GOTO 20
50 END

```

- Paragraphe 194 :

Voici comment on peut utiliser des abréviations dans un programme

LEVEL I

```

10 IN. A$ (POUR : INPUT A$)
20 P. "BONJOUR" ; A$ (P. = PRINT)

```

LEVEL II

```

10 INPUT A$
20 ? "BONJOUR" ; A$ (? = PRINT)

```

● Paragraphe 195 :

RUN	R.	GOTO	G.
LIST	L.	GOSUB	GOS.
PRINT	P.	RETURN	RET.
STOP	ST.	INPUT	IN.
THEN	T.	FOR	F.
END	E.	NEXT	N.
READ	REA.	STEP	S.
DATA	D.	SET	S.
RESTORE	REST.	RESET	R.

● Paragraphe 196 :

Entrez et exécutez ce programme avec abréviations :

```

10 F.P = 1 TO 5 : P. "PROBLEME"; P
20 F.N = 1 TO 2
30 P. "DONNEZ UN NOMBRE" : IN. A(N)
40 N.N
50 P. A(1) ; "X"; A(2); "="; A(1)*A(2)
60 N.P
70 E.

```

● Paragraphe 197 :

Entrez ce programme en utilisant des abréviations et des lignes à instructions multiples.

```

5 PRINT "NOMBRES ET CARRES"
10 GOSUB I000
20 READ Y
30 PRINT Y
40 IF Y = I0 GOTO 60
50 GOTO 20
60 RESTORE
70 GOSUB B I000
80 PRINT "CARRES"

```

```

90 READY
100 PRINT Y +Y ,
110 IF Y = THEN END
120 GOTO 90
1000 PRINT "*****"
1005 RETURN
1010 DATA 1,2,3,4,5,6,7,8,9,10

```

LECON 10, PARTIE 2

● Paragraphe 198 :

Voici comment on traduit un nombre de la notation scientifique en notation régulière :

$3 E + 07$ ← (3 est le réel 3.)
 $30000000.$ ← (on déplace le point décimal de 7 places et on ajoute des 0)

● Paragraphe 199 :

$1.72 E + 06$ (ici il y a des chiffres après le point décimal)
 172 (on a déplacé le point de 2 places, il en faut encore 4)
 1720000

● Paragraphe 200 :

$3 E - 04$
 $.3$ (on s'est déplacé de 1 place vers la gauche, il en faut encore 3)
 $.0003$ (on s'est déplacé de 3 places et ajouté des 0)

● Paragraphe 201 :

Entrez ce programme. Puis essayez les nombres montrés et regardez comment répond l'ordinateur

```

10 PRINT "ENTREZ UN NOMBRE " : INPUT A
20 PRINT A

```

30 GOTO I0

Entrées :

I000000	.1234567	333333
2111000	.01234567	
.01	.001234567	
003	1.00345	

Essayez d'autres nombres

Puis changez le programme pour entrer deux nombres et les multiplier. Regardez quand vous passez à la notation scientifique pour les nombres.

Puis divisez les deux nombres. Essayez d'obtenir des réponses en notation scientifique.

Questions Ecrites Pour les leçons 1-5

23. Ecrivez un programme imprimant 1 à I0 et terminez le programme quand il a tapé I0

24. Ecrivez une instruction produisant un nombre aléatoire de I à I00.

25. Ecrivez une instruction mettant - 7.3 dans une boîte appelé A

(réponses)

23. Voici la façon la plus simple de répondre à la question :

```

10 REM PROGRAMME COMPTANT JUSQU'A 10
20 LET X = 0
30 LET X = X + 1
40 PRINT X
50 IF X = 10 THEN STOP
60 GOTO 30

```

24. 10 PRINT RND (100)

ou

```

10 PRINT INT (100 * RND (0)) + 1

```

25. 10 LET A = - 7.3 ou 10 A = - 7.3

Questions Ecrites Sur Les Leçons 6-10

11. Ecrivez une instruction mettant la chaîne suivante dans B\$:
VOUS ETIEZ LA

12. Ecrivez un programme qui demande d'entrer votre nom puis l'imprime comme cela :

```

? JEAN
BONJOUR JEAN !

```

13. Ecrivez un programme utilisant un boucle for... NEXT qui compte à rebours de 10 à 1 et qui imprime les résultats

14. Ecrivez un programme comptant de 1 à 5 utilisant les instructions

```
READ .... DATA
```

15. Ecrivez un programme mettant la valeur de 1 dans chaque variable d'un tableau de A(1) à A(5)

Réponses :

```
11. 10 LET B$ = "VOUS ETIEZ LA"
    OU
    10 B$ = "VOUS ETIEZ LA "
```

```
12. 10 INPUT N$
    20 PRINT "BONJOUR"; N$ ;"!"
```

```
13. 10 FOR X = 10 TO 1 STEP -1
    20 PRINT X
    30 NEXT X
    40 END
```

14. Il y a plusieurs façon de faire. En voilà une.

```
10 DATA 1,2,3,4,5
20 READ C
30 PRINT C
40 IF C = 5 THEN STOP
50 GOTO 20
```

```
15. 10 FOR X = 1 TO 5
    20 LET A(X) = 1
    30 NEXT X
```

