

## S O M M A I R E

TOME 1	QU'EST-CE QU'UN ORDINATEUR ? POURQUOI UN ORDINATEUR ? COMMENT FONCTIONNE UN ORDINATEUR ? LA QUINCAILLERIE MODERNE S'EST ÉQUIPÉE D'UN ORDINATEUR ECRIVONS UN PROGRAMME LES LANGAGES DE PROGRAMMATION RÉALISONS QUELQUES PETITS PROGRAMMES EN BASIC RÉSUMÉ
TOME 2	QUELQUES RAPPELS DE LA THÉORIE À LA PRATIQUE
TOME 3	SAISIE DES DONNÉES ET AFFICHAGE CALCULS TESTS INSTRUCTION DE FIN DE TRAITEMENT INSTRUCTION DE BRANCHEMENT LES OUTILS DE PROGRAMMATION EXERCICES
TOME 4	LES VARIABLES ALPHANUMÉRIQUES ET LEUR TRAITEMENT TRANSFORMATION DES VARIABLES NUMÉRIQUES-ALPHANUMÉRIQUES BOUCLE AUTOMATIQUE TABLEAUX INSTRUCTION DE LECTURE SPÉCIALE SOUS-PROGRAMME BRANCHEMENT CALCULÉ EXERCICES
TOME 5	INSTRUCTIONS COMPLÉMENTAIRES CONCERNANT L'UNITÉ CENTRALE INSTRUCTIONS COMPLÉMENTAIRES CONCERNANT LES ENTRÉES-SORTIES - CLAVIER-ÉCRAN - IMPRIMANTE - MAGNÉTOPHONE  ANNEXES



*Note de l'auteur*



L'informatique n'est plus aujourd'hui une affaire de spécialistes seulement.

- . la miniaturisation de plus en plus poussée ( 100 000 composants électroniques sur une pastille de quelques millimètres carrés ).
- . la chute des prix des composants électroniques
- .... font que l'informatique est aujourd'hui à la portée de tous.

Les applications des micro-ordinateurs ?

Elles sont innombrables :

- . gestion ( comptabilité, salaires, stocks,... )
- . technique ( suivi de production, commande de machine-outil... )
- . statistiques, simulation, recherche,...
- . jeux

Il existe des milliers de programmes disponibles dans le commerce ( déjà enregistrés sur cassettes ou disquettes, ou publiés dans des ouvrages spécialisés ).

La micro-informatique ne peut plus être ignorée ; son évolution est prodigieuse : intégrés bientôt dans les postes de télévision, vendus dans des boutiques et même en hypermarché, les micro-ordinateurs remplaceront dans quelques années les machines à calculer classiques...

En éliminant tout vocabulaire de spécialiste, en basant l'ensemble de cette formation sur des exercices, exemples et expériences, en utilisant de nombreux schémas et illustrations, les plus novices d'entre vous en matière d'informatique parviendront très rapidement à assimiler les notions fondamentales pour pouvoir comprendre les techniques informatiques et les utiliser dans le cadre de leur profession ou de leurs loisirs.



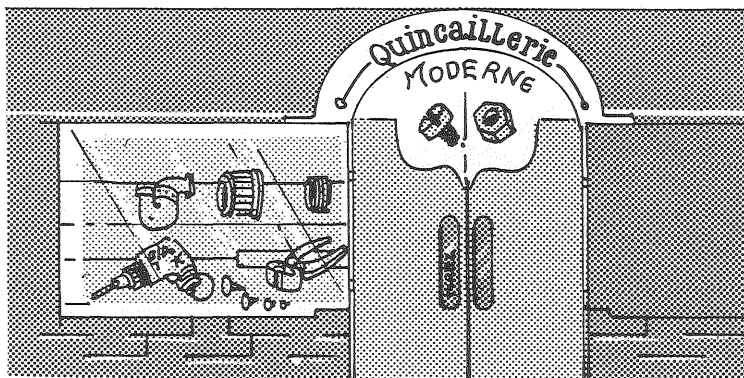
*Du'est-ce qu'un ordinateur ?*



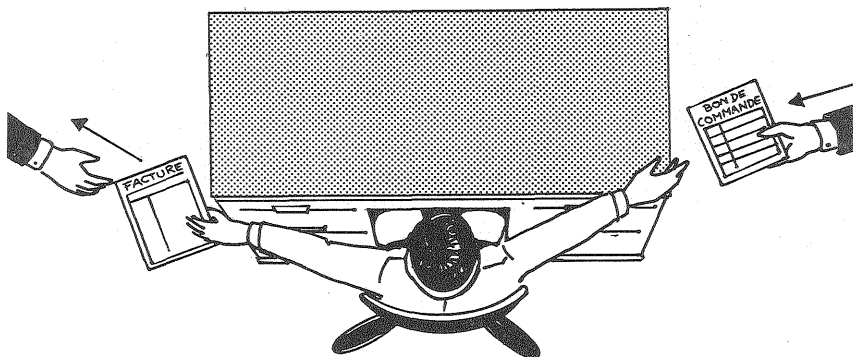


Pour bien comprendre le fonctionnement d'un ordinateur, nous allons rendre visite à Julien qui travaille dans une quincaillerie moderne.

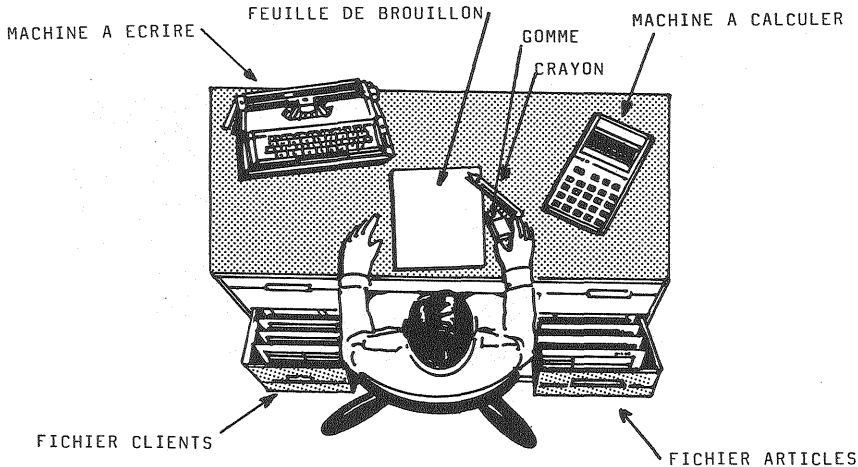
Julien vient d'être affecté au service facturation de cette entreprise.



Son travail consiste à recevoir des bons de commande et à en faire des factures.



Pour effectuer son travail, il lui faut naturellement un certain nombre d'outils : une machine à calculer, une machine à écrire, une feuille de brouillon, un crayon, une gomme, un fichier clients et articles.



En effet,

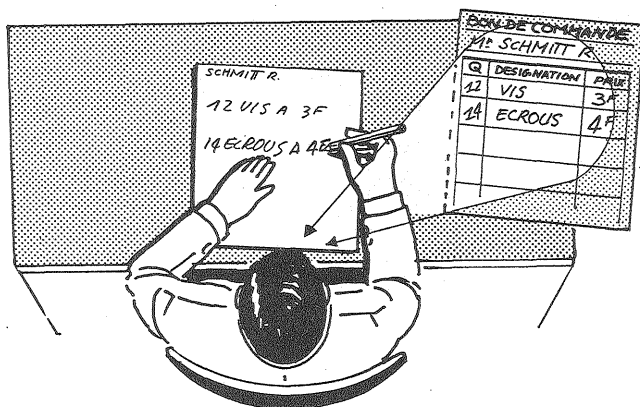
- . la machine à calculer lui servira pour effectuer les différentes opérations ( additions, multiplications,... )
- . la machine à écrire lui permettra de taper la facture.
- . la feuille de brouillon, le crayon et la gomme sont nécessaires pour noter les calculs intermédiaires.
- . les fichiers clients et articles permettront à Julien d'obtenir les informations indispensables pour pouvoir facturer ; car le bon de commande ne contient pas toujours toutes les données nécessaires.

Regardons d'un peu plus près ce qui se passe lorsque Julien doit faire une facture

- 1) La première étape de son travail est une opération de LECTURE

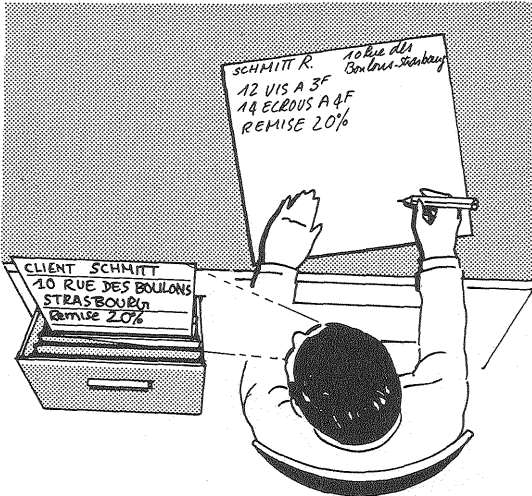
En effet, il doit rassembler sur sa feuille de brouillon toutes les informations qui lui seront nécessaires.

- a) Une partie des informations nécessaires, seront lues sur le bon de commande ( nom du client, désignation, quantité et prix de base )



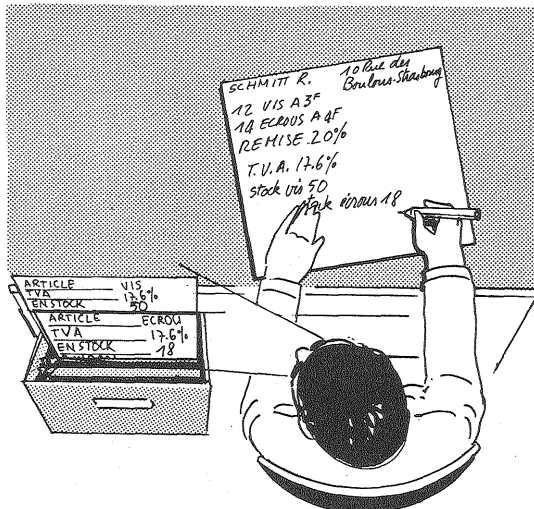
...mais avec ces seules informations, Julien ne pourra pas établir sa facture !...

- b) Il faudra donc que Julien complète les informations lues sur le bon de commande, par des informations plus précises provenant des fichiers clients et articles.  
Après avoir recherché dans le fichier clients, la fiche client, et dans le fichier articles, les fiches articles nécessaires, Julien pourra lire les informations dont il a besoin pour établir sa facture :



. la fiche client donnera par exemple l'adresse précise du client ainsi que la remise habituelle.

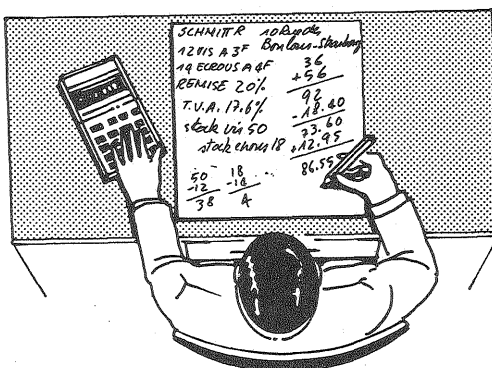
. les fiches articles indiqueront par exemple le taux de la taxe et la quantité d'articles disponibles en stock.



2) La deuxième étape de son travail est le TRAITEMENT :

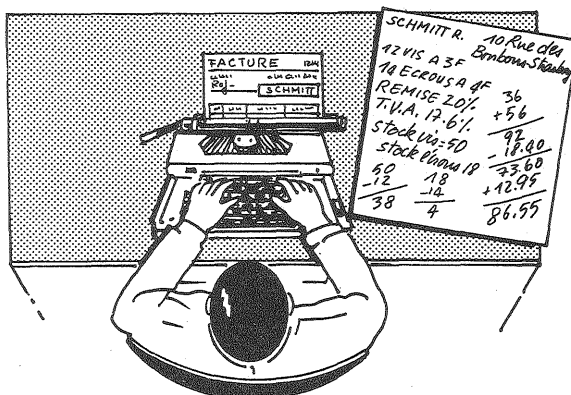
les informations notées sur la feuille de brouillon sont traitées ;  
pour l'établissement d'une facture, les opérations de traitement sont simples :

- calculs ( addition, soustraction et multiplication...)
- comparaisons ( Voir si le stock est suffisant par exemple )

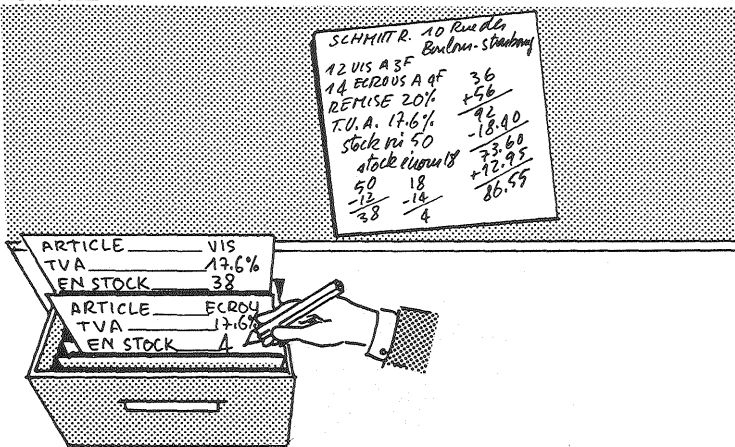
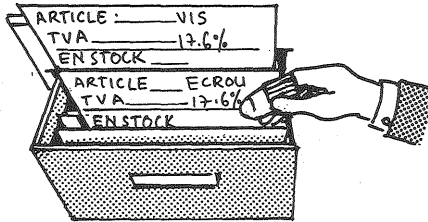


3) La troisième étape est la SORTIE des résultats :

- a) Julien reprend les informations de sa feuille de brouillon et les tape au propre sur la machine à écrire, pour établir sa facture.



- b) Julien reportera également certaines données sur les fiches des fichiers (Nouveau Stock par exemple), afin d'avoir des informations toujours à jour.



L'établissement d'une facture n'est qu'un exemple ;  
 en effet, tout travail quel qu'il soit ( établissement  
 d'une facture à partir d'un bon de commande, d'un bulletin  
 de paye à partir d'une d'une carte de pointage...)   
 peut se découper en 3 étapes :

1) Lecture ( ou ENTREE ) des informations :

Cette étape consiste à rassembler sur la feuille de  
 brouillon toutes les informations nécessaires.

- a) lecture des informations sur le document  
 en entrée
- b) lecture d'informations complémentaires  
 dans des fichiers

2) TRAITEMENT des informations :

Les informations notées sur la feuille de brouillon sont  
 traitées ( opérations d'additions, multiplications,  
 comparaisons,...)

Les résultats du calcul sont également notés sur la  
 feuille de brouillon.

3) SORTIE des résultats

- a) Toutes les informations notées sur la  
 feuille de brouillon ( ou certaines  
 d'entre-elles seulement ) sont tapées sur  
 la machine à écrire
- b) Certaines informations notées sur la feuille  
 de brouillon permettront la mise à jour des  
 fichiers

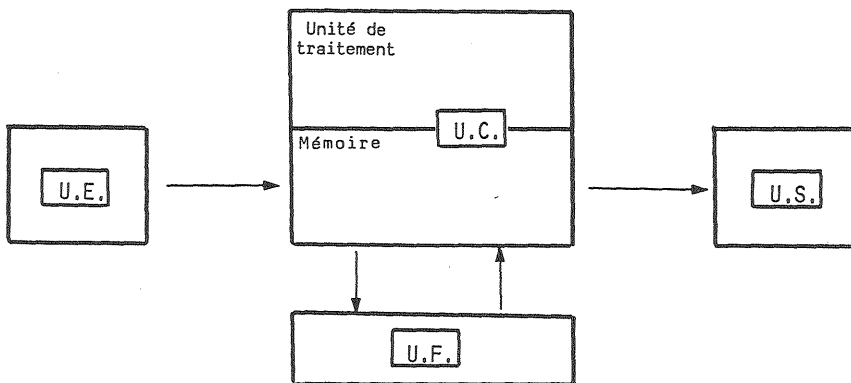
... Si nous voulons installer dans l'entreprise  
 un ordinateur capable d'établir des factures,  
 des bulletins de paye,..., il faut qu'il  
 puisse également réaliser ces fonctions :

- ENTREE ( lecture ) des informations
- TRAITEMENT des informations
- SORTIE des résultats
- LECTURE/ECRITURE sur FICHIER

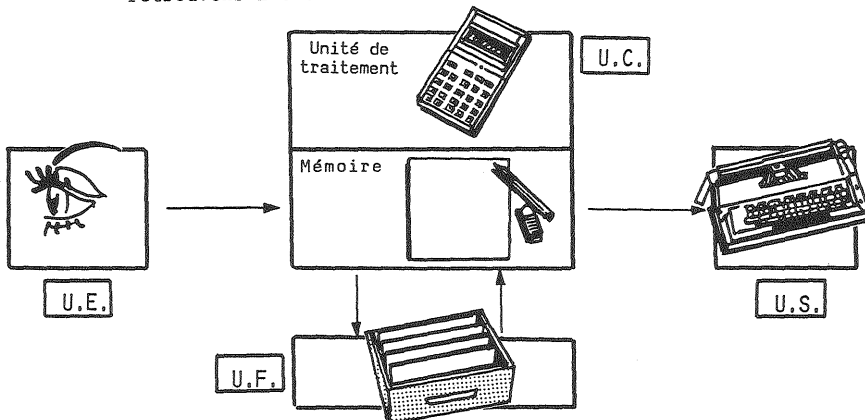
C'est ce qui explique pourquoi un ordinateur est composé de 4 appareils :

1. Une unité d'ENTREE (UE) ; son rôle est de lire les informations
2. Une unité CENTRALE (UC) ; son rôle est de traiter les informations
3. Une unité de SORTIE (US) ; son rôle est de sortir les résultats
4. Une unité de FICHIER (UF) ; son rôle est de stocker une masse d'informations

REMARQUE : L'unité centrale est composée de deux parties :  
l'unité de traitement et la mémoire.



Mais nous n'avons rien inventé, car dans l'ordinateur, nous retrouvons l'ensemble des outils utilisés par Julien :

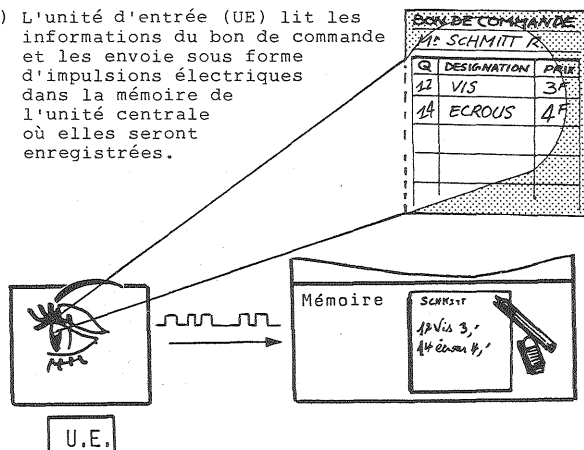




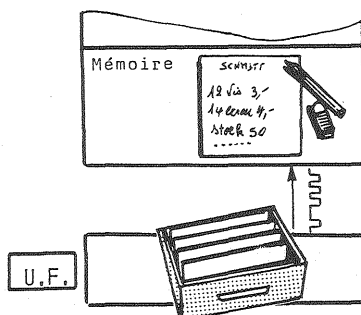
Si nous confions la facturation à notre ordinateur, sa manière de travailler sera identique à celle de Julien:

- 1) PREMIERE ETAPE : l'entrée ( ou la lecture ) des informations

- a) L'unité d'entrée (UE) lit les informations du bon de commande et les envoie sous forme d'impulsions électriques dans la mémoire de l'unité centrale où elles seront enregistrées.



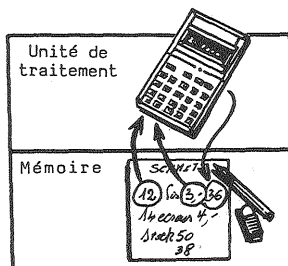
- b) Les informations sont complétées par les informations lues dans les fichiers articles et clients



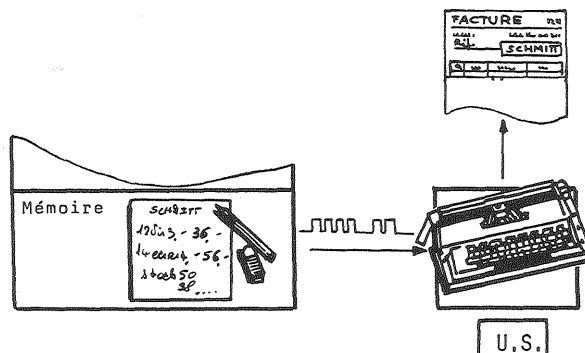
en effet , l'ordinateur recherchera les bonnes fiches articles et clients dans les fichiers, et les informations nécessaires seront envoyées sous forme d'impulsions électriques dans la mémoire de l'unité centrale où elles seront enregistrées.

2) DEUXIEME ETAPE : le traitement de l'information

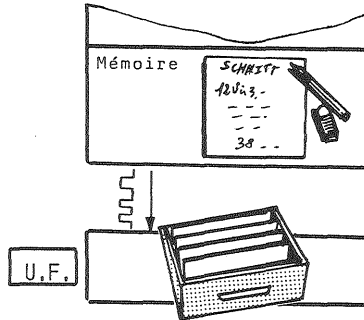
les informations enregistrées dans la mémoire sont envoyées dans l'unité de traitement pour y subir les opérations de calcul nécessaires ; les résultats de ces opérations sont également stockés dans la mémoire.

3) TROISIEME ETAPE : sortie ( ou édition ) des résultats

- a) L'unité de sortie (US) reçoit sous forme d'impulsions électriques les informations de la mémoire et édite la facture.



- b) l'unité de fichier reçoit sous forme d'impulsions électriques les informations de la mémoire, et les enregistre



Remarque :

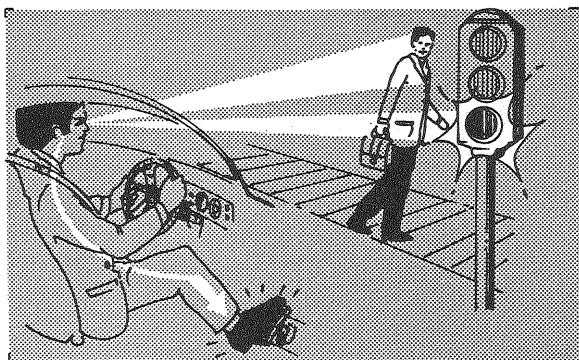
L'unité de fichier ( UF ) n'est pas toujours indispensable ; en effet, lorsque toutes les informations nécessaires au traitement se trouvent sur le document en entrée, et s'il n'est pas utile de conserver certains résultats, notre ordinateur peut se composer uniquement d'une unité d'entrée ( UE ), d'une unité centrale ( UC ) et d'une unité de sortie ( US ).



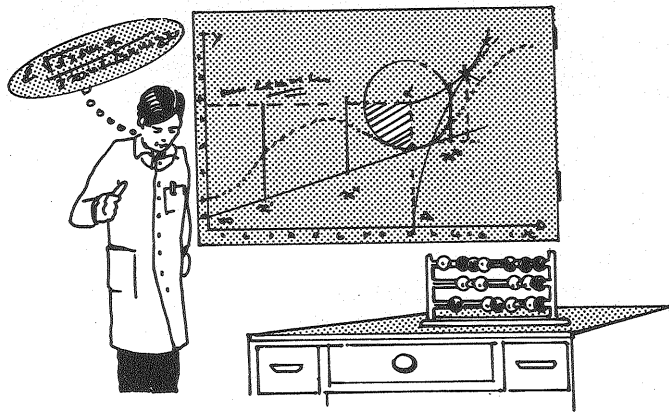
*Pourquoi un ordinateur ?*



Si dans la plupart des cas, les informations nécessaires interviennent en petit nombre, qu'un seul individu peut maîtriser, la décision est possible.



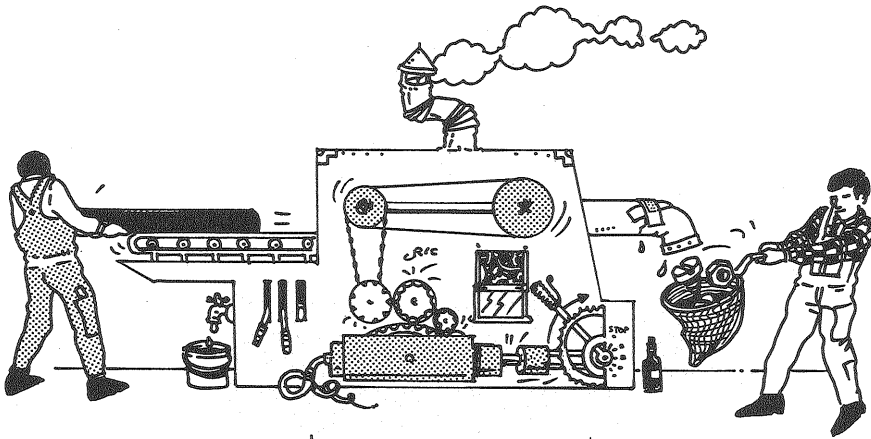
Mais lorsque la décision dépend d'un très grand nombre d'informations, et que les différentes opérations qui en permettent la synthèse sont longues et complexes, il sera nécessaire de faire appel à des outils plus performants.



Aujourd'hui le traitement de l'information devient parfois une tâche à la limite des possibilités humaines.

Seul l'ordinateur pourra faire face à une masse énorme d'informations, à des calculs mathématiques de plus en plus complexes et à des temps de réponse accordés de plus en plus courts pour prendre une décision .

Les machines que l'homme avait créées jusqu'alors fabriquaient ou traitaient quelque chose de concret...



UNITE  
D'ENTREE

UNITE  
CENTRALE

UNITE  
DE SORTIE

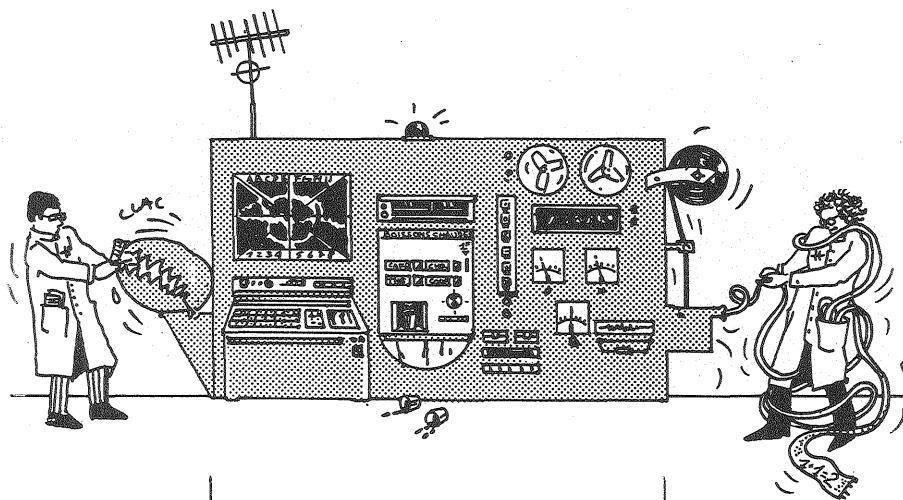
ENTREE  
de la matière première

TRAITEMENT  
de la matière première

SORTIE  
du produit fini



... pour l'ordinateur, la matière première comme le produit fini sont immatériels : ce sont des informations



UNITE  
D'ENTREE

UNITE  
CENTRALE

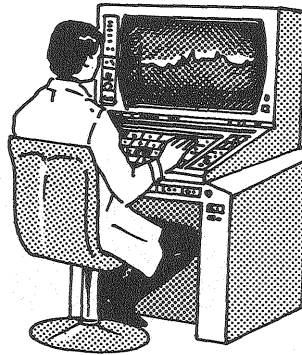
UNITE  
DE SORTIE

ENTREE  
des données

TRAITEMENT des données

SORTIE  
des résultats

L'homme s'était toujours doté d'outils manuels destinés à prolonger l'action de ses mains.



L'ordinateur est pour lui un outil intellectuel qui prolonge l'action de son cerveau.

L'informatique ( sciences et techniques du traitement de l'information ) a pris aujourd'hui un essort prodigieux dans tous les domaines :

- . gestion d'entreprise ( facturation, comptabilité, paye, statistiques,... )
- . réservation aérienne,...
- . calculs scientifiques
- . enseignement
- . médicale ( cliniques, médecins, pharmaciens, dentistes,... )
- . applications bancaires
- . applications individuelles ( jeux, enseignements,... )
- . applications industrielles ( automatismes, gestion et production,... )

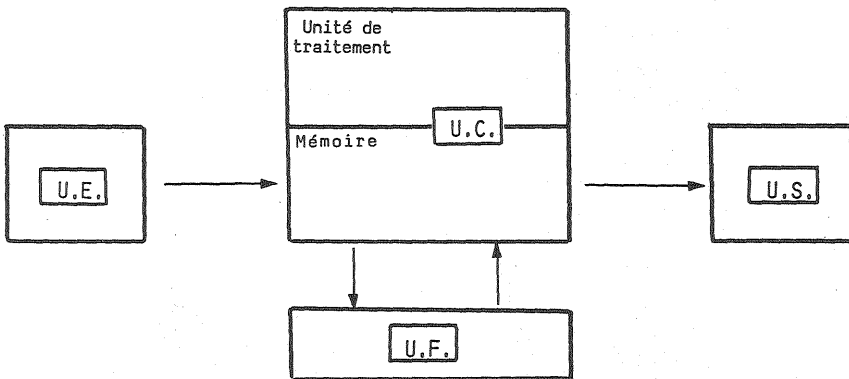
L'informatique est un domaine où chacun sera concerné, dans le cadre de son activité professionnelle ou dans le cadre des loisirs.

*Comment fonctionne un ordinateur ?*



Nous avons vu qu'un ordinateur est un ensemble de 4 appareils :

- 1'unité d'entrée (UE)
- 1'unité centrale (UC)
- 1'unité de sortie (US)
- 1'unité de fichier (UF)



Voyons dans le détail chacune de ces quatres unités.  
dans l'ordre suivant :

- 1) Unité d'entrée ( UE )
- 2) Unité de sortie ( US )
- 3) Unité de fichier ( UF )
- 4) Unité centrale ( UC ) ; unité de traitement et mémoire

# 1) UNITE D'ENTREE ( UE )

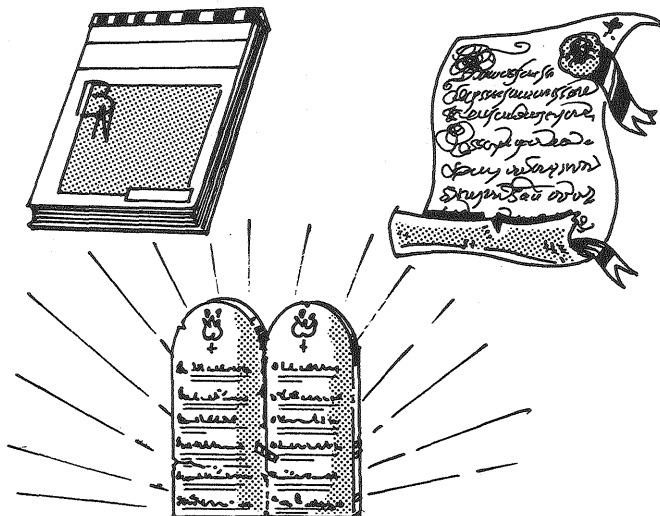


Un bon de commande est écrit à la main ; mais il faudra bien reconnaître qu'aucune machine n'est capable de lire, ni même de comprendre l'écriture humaine. pour cela, il faut donc présenter à notre ordinateur les informations d'une manière différente,

Les informations doivent donc être codées\* sur un soutpport\* pour que notre unité d'entrée soit capable de les lire.

## \* UN SUPPORT D'INFORMATION ?

Nous en avons utilisé de tout temps ( le bloc note, le parchemin... sont des supports d'information ).



L'ordinateur utilisera d'autres supports d'information ( par exemple les cartes perforées ) que nous verrons par la suite.

En informatique, il existe une grande variété de supports  
( cartes perforées, bandes perforées, cartes magnétiques,...).

Mais à chaque support d'information, correspond une unité  
d'entrée capable de lire ce support.

Exemple : si les informations sont perforées sur cartes,  
nous utiliserons un lecteur de cartes perforées  
comme unité d'entrée

si les informations sont perforées sur bandes,  
nous utiliserons un lecteur de bandes perforées  
comme unité d'entrée.

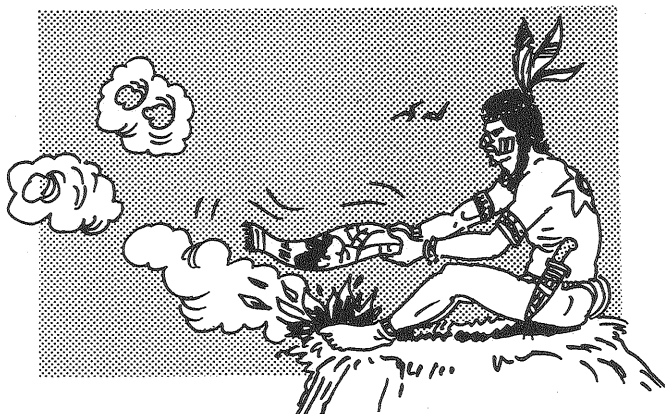
etc...

Les unités d'entrée quelles qu'elles soient, fonctionnent  
toutes selon un même principe :

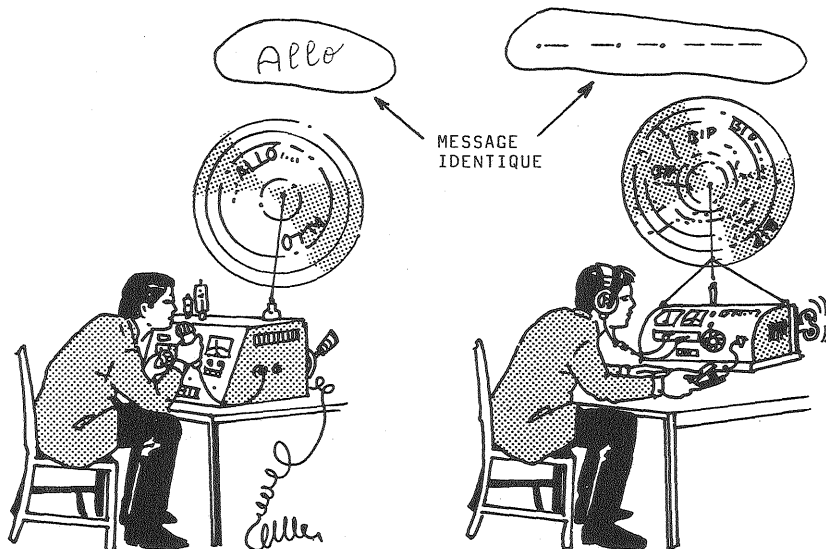
- Reconnaître l'information enregistrée sur  
le support
- Transformer cette information sous forme  
d'impulsions électriques
- Envoyer ces impulsions électriques dans la  
mémoire de l'unité centrale où elles  
seront enregistrées.

\* UNE INFORMATION CODEE ?

De tout temps, l'homme a utilisé des codes pour transmettre une information.



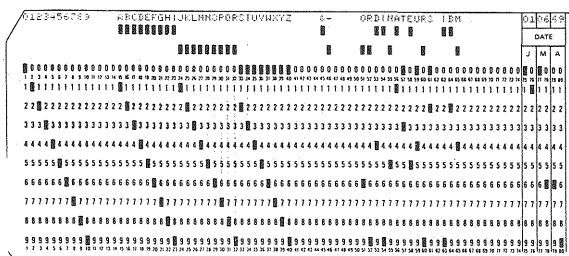
Bien entendu une information codée n'est pas une information dont le sens est transformé : par exemple un message émis par la voix ou en morse est le même ; ce n'est que la représentation de l'information qui est différente !





### QUELQUES UNITES D'ENTREE ET SUPPORTS

- Le support d'information le plus connu est la CARTE PERFOREE ( il existe plusieurs modèles de cartes perforées ) ; l'information y est représentée sous forme d'une combinaison de perforations.

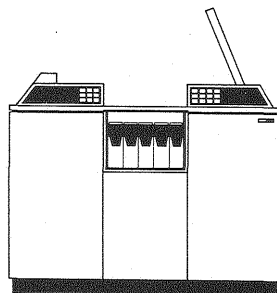


CARTE PERFOREE 80 COLONNES

La carte perforée présentée ci-dessus est une carte 80 colonnes; cela signifie que nous pouvons y perforer 80 caractères ( un caractère est un chiffre, une lettre ou un signe ).

Chaque caractère est codé  
( exemple : le T est codé par un trou en ligne 0 et en ligne 3 )

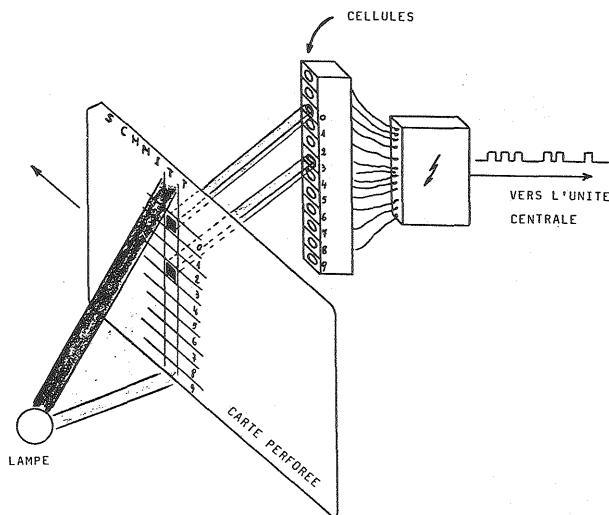
Pour lire ce support  
d'information, on utilisera  
UN LECTEUR DE CARTES  
PERFOREES 80 COLONNES



LECTEUR DE CARTES PERFOREES



- Pour lire ces cartes, notre ordinateur utilisera un lecteur de cartes perforées :



Un lecteur de cartes perforées, comporte une rangée de cellules photo-électronique ( pièce sensible à la lumière ) ;

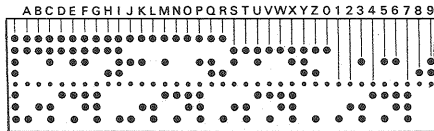
la carte perforée entraînée par un moteur, passe entre une lampe et les cellules ; ainsi, en fonction des positions des trous sur la carte, certaines cellules reçoivent de la lumière, d'autres n'en reçoivent pas.

L'électronique du lecteur de carte reconnaîtra ainsi les caractères perforés sur la carte, et les transmettra sous forme d'impulsions électriques dans la mémoire de l'unité centrale.

Dans notre exemple ci-dessus, c'est la sixième colonne de la carte qui se trouve entre la lampe et la rangée de cellules ; les trous en ligne 0 et 3 font que seules les cellules N° 0 et N° 3 reçoivent de la lumière ; le circuit électronique (⚡) reconnaîtra donc la lettre T et le transmettra sous forme d'impulsions électriques dans la mémoire de l'unité centrale.

### - LA BANDE PERFOREE

Comme pour la carte perforée, l'information est représentée sous forme d'une combinaison de perforations.

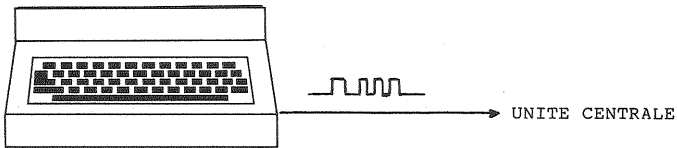


BANDE PERFOREE

Ces petites perforations  
permettent l'avancement de la  
bande dans LE LECTEUR DE BANDES  
PERFOREES

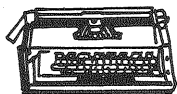
Le système de lecture d'une unité de bandes perforées est identique à celui d'un lecteur de cartes perforées ; seule la partie mécanique ( entraînement de la bande ) est bien sûr différente !

### - Le CLAVIER



Les informations tapées au clavier sont directement envoyées sous forme d'impulsions électriques dans la mémoire de l'unité centrale.

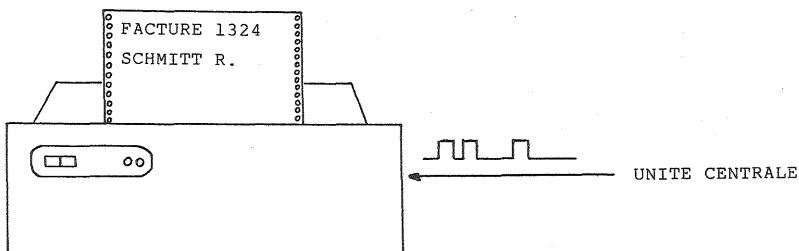
2) UNITE DE SORTIE ( US )



Il existe une grande variété d'unités de sortie :

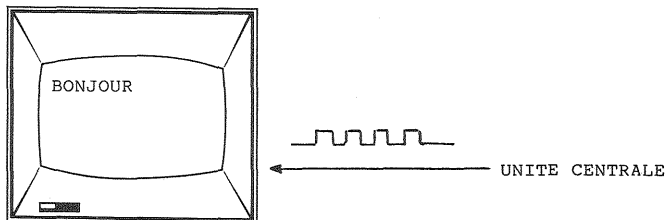
- l'imprimante

"Machine à écrire" automatique commandée par les impulsions provenant de la mémoire de l'unité centrale

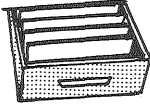


- L'écran

Dans ce cas, les informations provenant de la mémoire de l'unité centrale, sont affichées sur un écran



### 3) UNITE DE FICHIERS ( UF )



- Le rôle de l'unité de fichier est de fournir à la mémoire de l'unité centrale les informations nécessaires ; mais également de recevoir de la mémoire centrale certains résultats de traitement
- Sur une unité de fichier, les informations sont stockées sur des supports ; les supports les plus couramment utilisés sont : la bande magnétique, la cassette magnétique, les disques et les disquettes magnétiques
- L'unité de fichier est parfois appelée "unité d'entrée/sortie" ( U E/S ) ; en effet, l'unité de fichier sert d'unité d'entrée, car l'unité centrale peut y lire des informations ; mais elle sert également d'unité de sortie, car après traitement, les résultats peuvent y être envoyées pour y être stockées.
- L'unité de fichier est également parfois appelée "mémoire auxiliaire" ; en effet, l'unité de fichier se justifie lorsqu'on veut stocker une masse importante d'informations ; la mémoire centrale ayant une capacité trop faible pour les stocker !

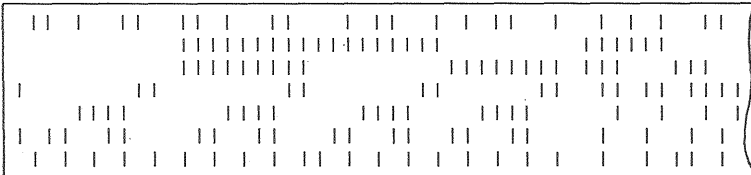
#### QUELQUES UNITES DE FICHIERS ET SUPPORTS

##### LA BANDE MAGNETIQUE

L'information y est enregistrée sous forme de petits "spots" magnétiques selon un principe voisin du magnétophone.

Comme sur la carte ou bande perforée, chaque caractère est codé :

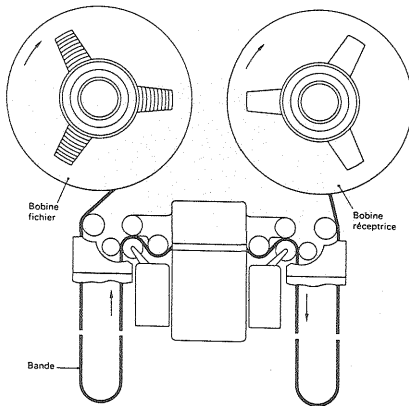
0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z . & , - F \* / , % # @



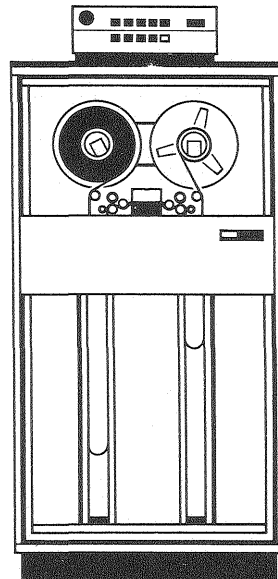
BANDE MAGNETIQUE

La bande magnétique est lue par une UNITE DE BANDES MAGNETIQUES

Contrairement à la carte ou bande perforée, où ce sont des cellules photoélectroniques ( pièces sensibles à la lumière ) qui détectent la position des perforations, nous aurons ici une série de têtes de lecture, identiques à celles d'un magnétophone, qui liront les informations enregistrées sur la bande magnétique ; et une série de têtes d'écriture qui enregistreront sur la bande, les informations



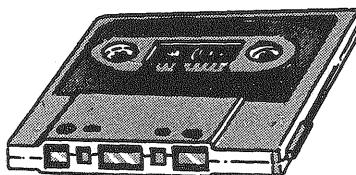
Sur certaines unités de bandes magnétiques de grande vitesse, deux tubes à décompression permettent d'assurer une boucle de sécurité du côté de la bobine débitrice et du côté de la bobine réceptrice, afin d'éviter des tensions trop importantes lors du démarrage du moteur ; ( une tension trop importante sur la bande magnétique entraînerait sa déformation d'où risque de perte d'informations ).



# - LA CASSETTE MAGNETIQUE

Le principe de codification est voisin de celui de la bande magnétique.

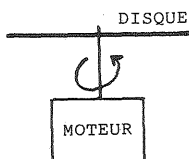
L'unité capable de lire ce support est une unité de cassettes magnétiques.



Le principe d'enregistrement et de lecture est identique à celui de la bande magnétique ; seule la partie mécanique est différente.

# - LE DISQUE ET DISQUETTE MAGNETIQUE

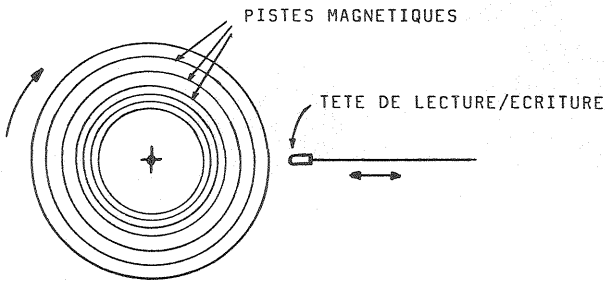
Le disque ou disquette magnétique est monté sur un axe et tourne à grande vitesse.





Le principe d'enregistrement de l'information est magnétique comme sur la bande et la cassette.

Les informations sont mémorisées sur des pistes magnétiques concentriques d'un disque tournant à grande vitesse. Les informations seront lues par une tête de lecture ou enregistrées par une tête d'écriture. Tête de lecture et tête d'écriture sont montées sur un axe se déplaçant perpendiculairement à l'axe du disque.



(Vue de dessus)

### Mais pourquoi une telle variété de supports d'informations ?

Chaque support a un certain nombre d'avantages et d'inconvénients. C'est le rôle de l'informaticien de choisir les supports ( donc les unités d'entrées, de sorties et de fichiers ) les plus adaptés au travail que notre ordinateur aura à effectuer

Chaque support possède un certain nombre de caractéristiques :

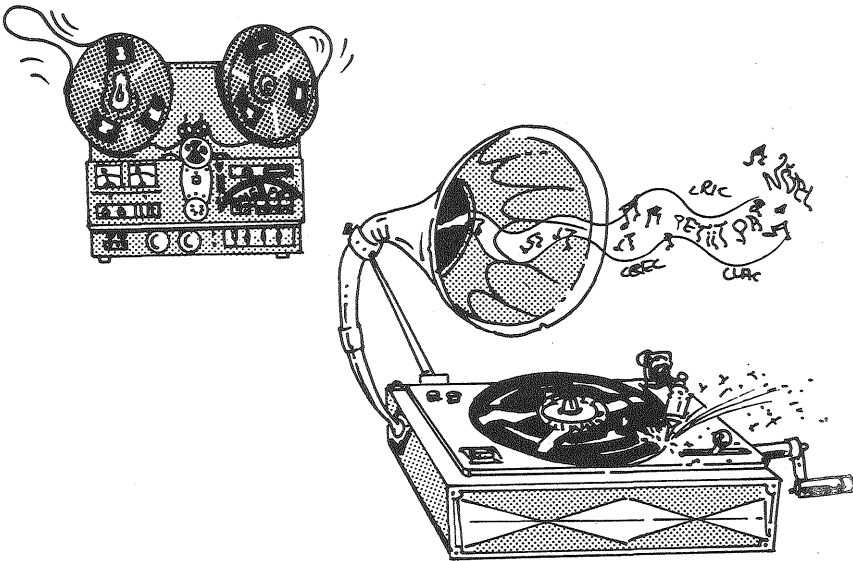
Exemple : les supports magnétiques sont sensibles à la poussière  
les supports perforés sont sensibles à l'humidité.

- Un support magnétique peut être réutilisé ; il suffit de l'effacer.
- Un support perforé ne peut pas être réutilisé ( à moins de reboucher les trous !!!... )

Pour bien comprendre la différence fondamentale qui existe entre les différents supports, prenons un exemple :

Comparaison de la bande magnétique et du disque magnétique

Pour bien comprendre les avantages et les inconvénients de ces 2 supports, comparons le magnétophone et le tourne-disques.



- Vous avez sûrement déjà remarqué qu'il vous est possible d'enregistrer beaucoup plus de chansons sur votre bande magnétique que ce qui se trouve sur un 33 tours ; de même le coût d'une bande magnétique est inférieur à celui d'un 33 tours.

Sur notre ordinateur, nous pouvons constater les mêmes avantages de la bande magnétique par rapport au disque magnétique : la bande magnétique a une capacité de stockage supérieure et un coût inférieur qu'un disque magnétique.

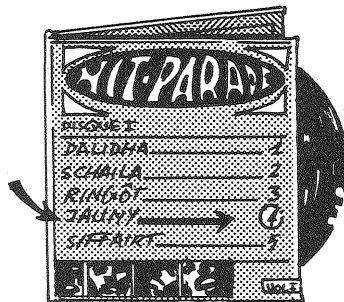
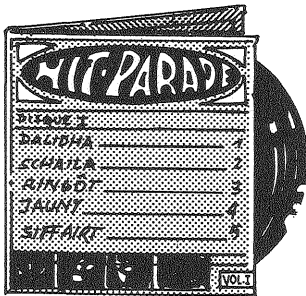
Si le problème de l'entreprise est d'archiver une quantité importante d'information, elle choisira donc la bande magnétique.

.... et l'avantage du disque ?

- Si vous décidez d'écouter une chanson sur votre magnétophone qui se trouve, dans le pire des cas, à la fin de la bande magnétique, vous serez obligés de faire défiler la bande jusqu'à l'endroit où se trouve le début de la chanson choisie.

Avec notre unité de bande magnétique le problème est identique : le défilement de la bande pour arriver jusqu'à l'information recherchée entraînera une perte de temps considérable.

Par contre, si vous écoutez votre chanson préférée sur votre tourne-disques, le temps de recherche est considérablement réduit, grâce au tableau imprimé sur la pochette de votre disque...



Imaginons que votre favori soit JAUNY ; il suffit de rechercher dans la table le numéro de la page puis de positionner directement le bras de votre tourne-disques au bon endroit.

Avec notre unité de disque ou disquette magnétique nous pouvons bénéficier du même avantage : en fournissant à notre ordinateur une table qui d'un côté donne la liste des clients par exemple, et de l'autre côté le numéro de la piste où les informations de ce client sont enregistrées, la tête de lecture pourra directement se positionner au-dessus de la piste où se trouvent les informations concernant le client en question.

client	N° piste
.	.
.	.
.	.
LAUTH T.	145
SCHMITT R.	128
.	.
.	.
.	.
WOLFF A.	62
.	.
.	.
.	.

Cette méthode de recherche de l'information permettra d'obtenir des temps d'accès ( de l'ordre de quelques millisecondes ) beaucoup plus courts que sur la bande ou la cassette magnétique

Si le problème de l'entreprise, est la recherche rapide d'informations, elle choisira donc le disque ou disquette magnétique.

- Cette comparaison entre unité de bandes et de disques ou disquettes magnétiques nous montre qu'il existe plusieurs méthodes d'accès à une information enregistrée sur un support :

- l'accès séquentiel

( sur bandes et cassettes magnétiques par exemple )

Le support d'information est lu séquentiellement à partir du début jusqu'à l'endroit où l'information est enregistrée.

- l'accès sélectif

( sur disques et disquettes magnétiques par exemple )

Grâce à une table, on délimite un sous-ensemble de recherche ( la piste ), puis la recherche se fera séquentiellement dans le sous-ensemble exclusivement.

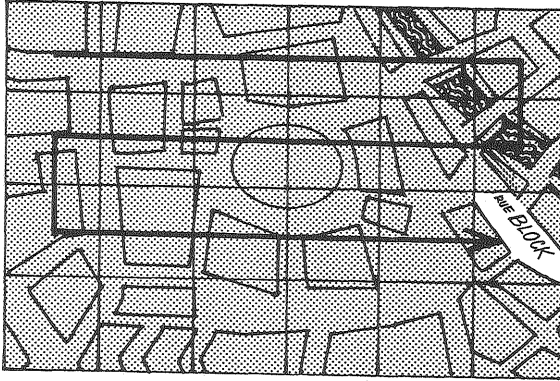
Remarque : un disque ou disquette peut, également, se lire séquentiellement aussi !

Pour bien comprendre ces 2 méthodes d'accès, prenons un exemple :

ROGER cherche  
sur le plan  
la Rue BLOCK :  
deux méthodes  
pour la trouver

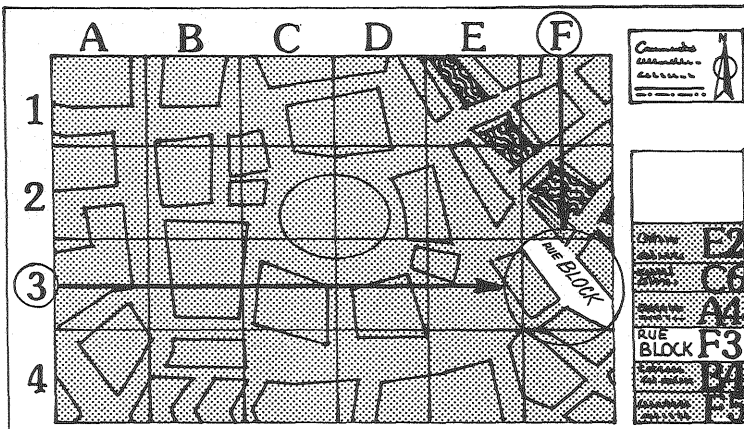


# 1. LA RECHERCHE SEQUENTIELLE



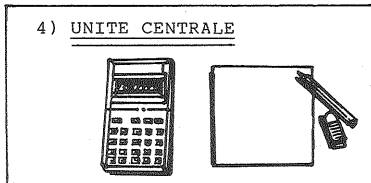
En partant de la case en haut à gauche, Roger parcourt successivement toutes les cases de son plan jusqu'au moment où il aura trouvé la rue Block.  
 Cette méthode risque d'être longue, et le temps de recherche est variable car il dépend de l'endroit où se trouve la rue Block

# 2. LA RECHERCHE SELECTIVE



La table lui indique que la rue Block se trouve en F3, il limitera donc sa recherche à ce sous-ensemble de coordonnées F3

#### 4) UNITE CENTRALE



L'unité centrale se compose d'une mémoire et d'une unité de traitement :

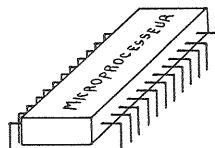
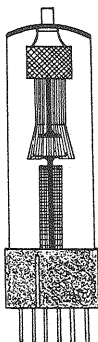
- la mémoire représente un ensemble de circuits électroniques capables de recevoir des informations, de les stocker ( conserver ) et de les restituer. En fin de travail, elle peut s'effacer comme la feuille de brouillon de Julien.

Dans la mémoire se trouvent stockées les informations lues par l'unité d'entrée et par l'unité de fichier mais également les résultats des opérations de notre unité de traitement.

- les circuits électroniques de notre unité de traitement permettent d'effectuer plusieurs centaines de milliers d'opérations arithmétiques ou logiques ( comparaisons, tests,... ) par seconde.

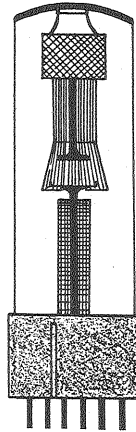
Alors qu'il y a encore une dizaine d'années, l'unité centrale représentait la partie la plus importante ( en volume ), ses principaux circuits se retrouvent aujourd'hui miniaturisés sur une pastille de quelques millimètres carrés.

L'évolution de la technologie et de la miniaturisation est prodigieuse

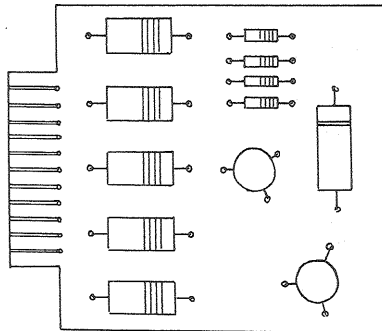
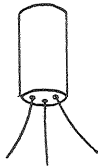


- . Pour la première fois, en 1885, on utilisera un ordinateur à cartes perforées mis au point par le Docteur HOLLERITH pour le dépouillement du recensement de la population des Etats-Unis.
- . Mais le premier véritable ordinateur date de 1946 : l'ENIAC ; il comportait 18000 tubes à vide. Depuis 1946, les progrès ont été considérables ; les générations d'ordinateurs se sont succédées :

. la première ( 1946 ) fut celle des tubes à VIDE



. la deuxième ( 1959 ) fut celle des transistors, et des circuits imprimés



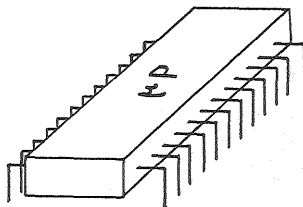


. la troisième ( 1964 ) fut celle des circuits intégrés

. la quatrième ( 1970 ) est caractérisée par la super miniaturisation des circuits

Aujourd'hui, la miniaturisation a permis d'intégrer sur une pastille de quelques millimètres carré (CHIP ou PUCE) la plupart des circuits électroniques de notre unité centrale :

le microprocesseur (  $\mu P$  )

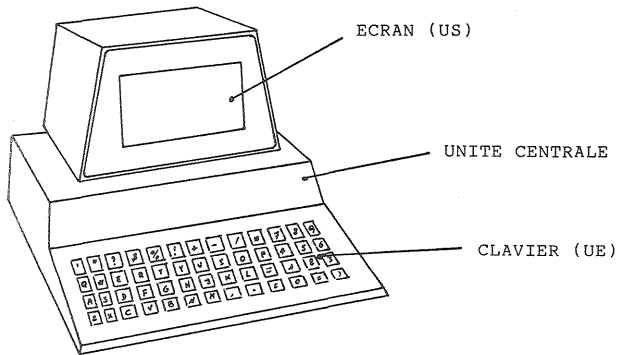


Plusieurs centaines de milliers de transistors sur une puce de quelques millimètres carrés !

Remarque :

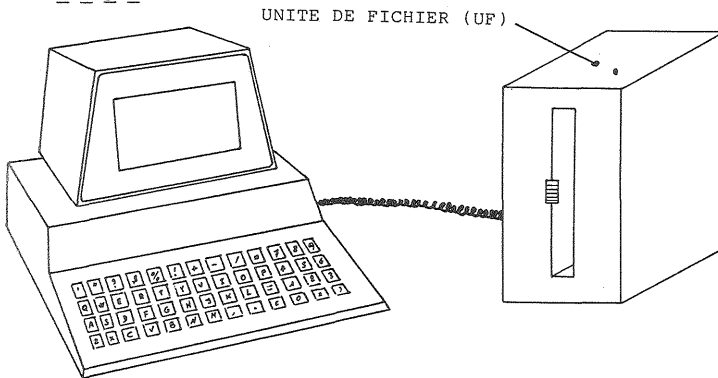
- le plus petit système informatique est composé au minimum d'une unité d'entrée ( UE ), d'une unité de sortie ( US ) et d'une unité centrale ( UC )

Exemple :



- Mais pour beaucoup d'applications, il sera nécessaire de disposer d'une unité de fichier ( UF ) pour y stocker les informations

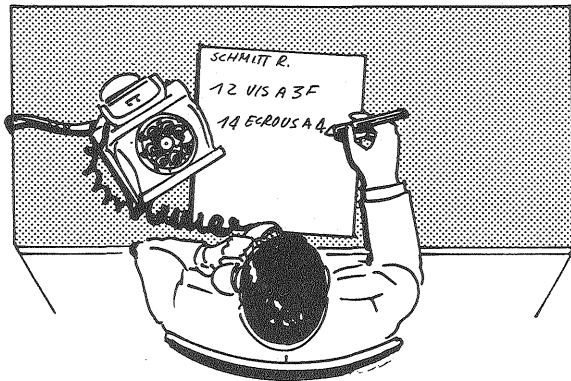
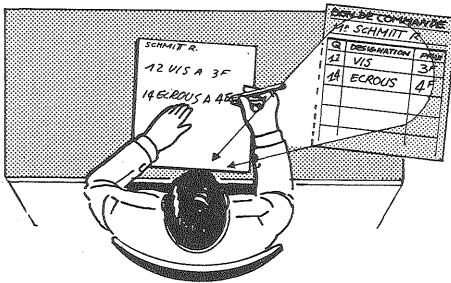
Exemple :



- En fonction des applications à informatiser, il sera nécessaire de bien choisir les unités d'entrées, de sorties et de fichiers.

Il n'est pas exclu de concevoir un système informatique avec plusieurs unités d'entrée, plusieurs unités de sortie ou plusieurs unités de fichier ; on choisira en fonction des applications à informatiser les unités les mieux adaptées

... mais au fond, nous n'avons toujours rien inventé !  
Julien, ne faisait-il pas la même chose !



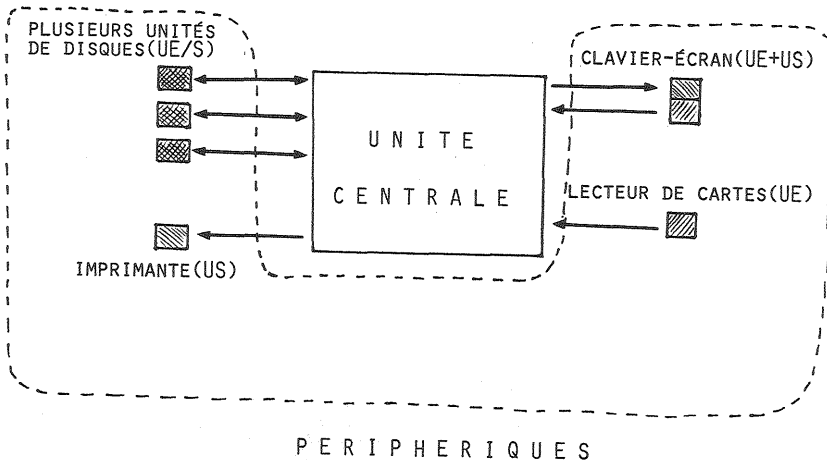
En effet, au lieu de communiquer à Julien visuellement le bon de commande, on peut dans certains cas ( commande téléphonique, par exemple ), lui communiquer les mêmes informations oralement ! la suite du traitement restera identique. En fait, Julien dispose lui aussi plusieurs unités d'entrées ( ses sens ! ).

**En résumé**

un ordinateur ou un système informatique se compose :

- d'une unité centrale
- d'une ou plusieurs unité d'entrée
- d'une ou plusieurs unité de sortie
- éventuellement d'une ou plusieurs unité de fichier

Toutes ces unités qui gravitent autour de l'unité centrale sont appelées des périphériques.

Exemple :

*La «quincaillerie moderne »  
s'est équipée d'un ordinateur*



Après avoir bien analysé les besoins, la société a décidé de s'équiper d'un ordinateur pour faire la facturation.

L'ordinateur choisi se compose :

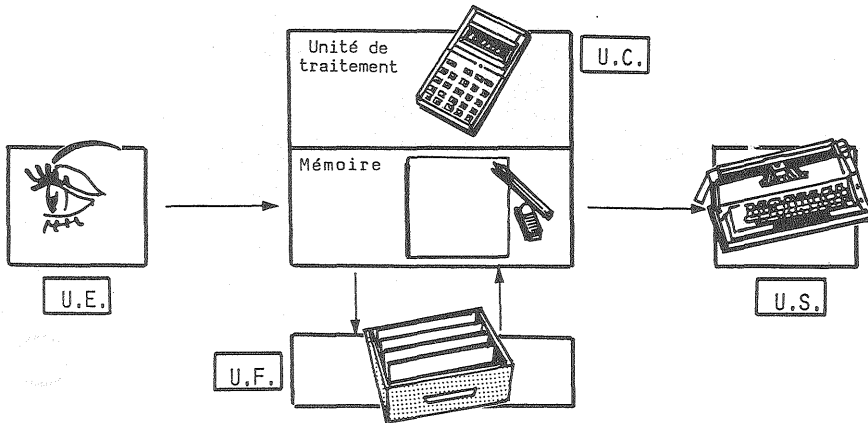
- d'une unité centrale
- d'un clavier
- d'une imprimante
- d'une unité de disquette

Après installation du matériel, l'on décida d'éditer les premières factures.

Après avoir tapé sur le clavier toutes les informations du bon de commande, aucun résultat n'est apparu sur l'imprimante. Seuls des messages d'erreur incompréhensibles ont été imprimés.....?????.....

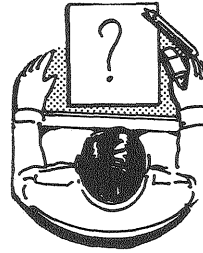
Cela est impossible ! rien n'a été oublié ; nous avons bien tous les outils nécessaires !

... tous les outils que Julien disposait sont bien là !

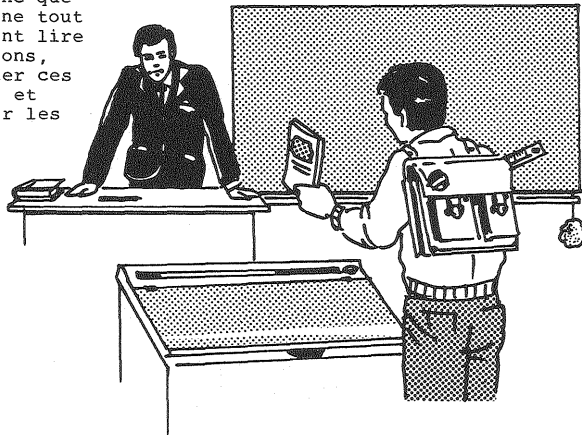


? Nous avons du oublier quelque chose : pour en avoir le coeur net, rendons une nouvelle visite à Julien...

Mais c'est bien sûr... Après quelques instants de discussion avec notre ami Julien, le mystère s'est éclairci. Julien nous a avoué que lui également était incapable d'établir une facture au début ! car il venait d'être engagé récemment dans le service facturation et ignorait tout de la facturation. Il ne savait pas ce qui se trouve sur un bon de commande, il ne savait pas comment calculer le hors taxe, la taxe, etc... et il ne savait pas non plus comment présenter une facture.



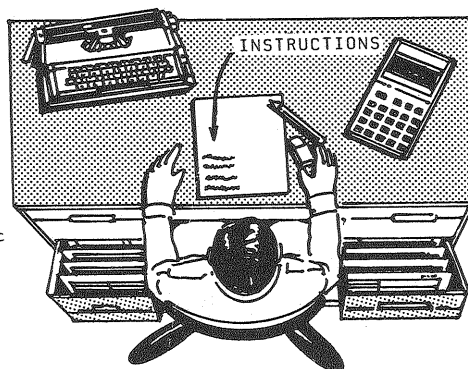
Il fallait donc que Julien apprenne tout d'abord comment lire les informations, comment traiter ces informations, et comment éditer les résultats.





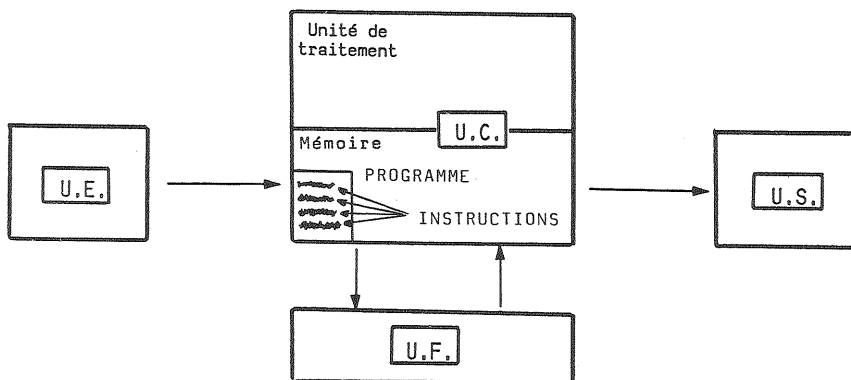
Julien se nota soigneusement sur un coin de sa feuille de brouillon, l'ensemble des instructions que le responsable du service de facturation lui avait donné.

Il lui suffisait ainsi de suivre minutieusement ces instructions pour réaliser correctement son travail.



Pour l'ordinateur il en est de même; avant de lui demander de réaliser un travail, il faut lui expliquer ce qu'il y a à faire, c'est-à-dire, enregistrer dans sa mémoire les instructions précises qui lui permettront de résoudre le problème. ( Ces instructions seront fournies à l'ordinateur par l'intermédiaire d'une unité d'entrée ) .

L'ensemble des instructions qui expliquent à notre ordinateur le travail à réaliser, s'appelle un programme.



L'ordinateur est la machine la plus stupide que l'homme ait inventé. Pour remplir sa mission, il lui suffit d'exécuter bêtement les instructions qu'on lui aura données.

Le mystère du non fonctionnement de l'ordinateur de la quincaillerie industrielle est maintenant totalement éclairci.

Il suffit de lui expliquer le travail qu'il doit faire :

pour cela il faudra écrire un programme !  
... et lorsque ce même ordinateur devra effectuer d'autres applications ( paye, comptabilité... ) il faudra également lui fournir d'autres programmes.

. Un système informatique est en fait composé de 3 éléments :

- 1) le HARDWARE ( le MATERIEL en français )  
en traduction littérale, Hardware veut dire "tas de ferraille"
- 2) le SOFTWARE ( le LOGICIEL en français )  
en traduction littérale, software veut dire "matière grise" ;  
il s'agit en fait des programmes, qui mis en mémoire,  
permettent au matériel de réaliser le travail demandé  
...et n'oublions pas :
- 3) les informations  
en effet, la qualité des résultats fournis par un système informatique, dépend bien sur du bon choix du matériel d'un logiciel bien écrit, bien adapté aux besoins, tenant compte de tous les cas particuliers de l'application, mais aussi de la qualité des informations fournies au système.

Il serait utopique d'attendre des merveilles d'un système informatique si les données qui lui ont été transmises sont erronées ou incomplètes.

*Ecrivons un programme*

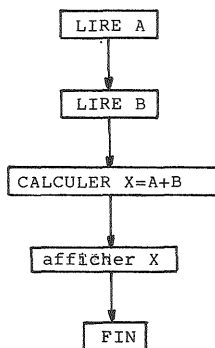


Imaginons que nous disposions du matériel suivant :

- un clavier comme unité d'entrée
- une unité centrale
- un écran comme unité de sortie

et que nous voulions par exemple demander à notre ordinateur de lire deux nombres, d'en faire l'addition et d'afficher le résultat sur l'écran

Pour écrire notre programme, nous devons d'abord analyser le problème et décrire d'une façon élémentaire et précise chacune des instructions.



Cette représentation schématique décrivant la suite des opérations à effectuer pour arriver au résultat s'appelle un organigramme

- . Dans les deux premières instructions nous expliquons à l'ordinateur qu'il doit lire successivement deux nombres appelés par exemple A et B et les stocker dans la mémoire de l'unité centrale.
- . L'instruction suivante indique à l'ordinateur qu'il doit prendre les nombres A et B stockés dans la mémoire, les envoyer dans l'unité de traitement pour qu'il puisse les additionner ; le résultat de cette addition appelé par exemple X sera également stocké dans la mémoire.
- . L'avant dernière instruction indique à l'ordinateur qu'il doit récupérer dans la mémoire le nombre X puis l'afficher sur l'écran.
- . Enfin, la dernière instruction signale à l'ordinateur que le traitement est terminé.

Il va falloir communiquer maintenant ce programme à notre ordinateur ;  
pour qu'il puisse comprendre il va falloir lui donner le programme dans un langage qu'il comprend ; imaginons que notre ordinateur comprenne le BASIC ( ce qui est le cas pour la plupart des micro-ordinateurs ) :

En BASIC ce programme s'écrit :

```
1 INPUT A          INPUT veut dire lire
2 INPUT B
3 X=A+B
4 PRINT X          PRINT veut dire afficher
5 END
```

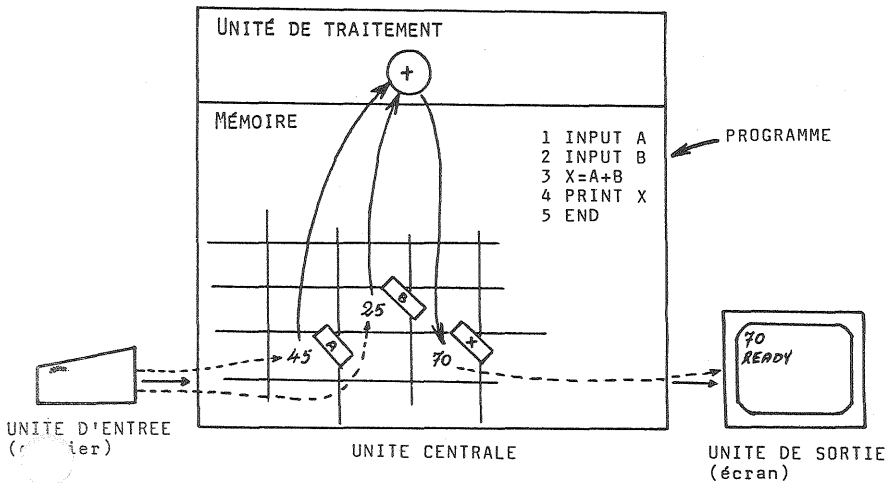
Chaque instruction est numérotée dans l'ordre dans lequel elles doivent être exécutées.

Après avoir chargé ce programme dans la mémoire de l'unité centrale ( cette opération se fait en tapant les 5 instructions sur le clavier ), il pourra être exécuté :

- l'opérateur donne ordre à l'ordinateur d'exécuter le programme qui se trouve dans sa mémoire.
- l'ordinateur exécute alors la première instruction : il demande à l'opérateur d'introduire le 1er nombre ; l'opérateur tape sur le clavier le nombre 45 par exemple, ce nombre sera stocké dans une "case" libre de la mémoire et l'ordinateur "colle" l'étiquette A pour pouvoir retrouver ce nombre.
- l'ordinateur exécute la deuxième instruction, il demande à l'opérateur d'introduire le deuxième nombre, l'opérateur tape sur le clavier le nombre 25 par exemple, ce nombre sera également stocké dans la mémoire.
- l'ordinateur exécute ensuite la troisième instruction. Il récupère en mémoire le contenu des "cases" A et B ( c'est à dire les nombres 25 et 45 ), les envoie dans l'unité de traitement qui les additionne, le résultat de l'addition (70) est également stocké en mémoire dans une "case" libre.

- Puis l'ordinateur exécute la quatrième instruction, il récupère en mémoire le contenu de la "case" X ( c'est à dire le nombre 70 ) et l'affiche sur l'écran.
- Enfin la cinquième instruction indique à l'ordinateur que le traitement est terminé, il affichera à l'écran le message : READY, ce qui veut dire :

" Prêt pour un nouveau JOB. "







## *Les langages de programmation*



Il existe de nombreux langages de programmation :

BASIC, FORTRAN, COBOL, PASCAL, PL/1, ALGOL, GAPII,....etc

Chaque langage possède un certain nombre d'avantages et d'inconvénients ; ils sont généralement classés en 2 catégories :

les langages de gestion, comme le COBOL,  
utilisés pour des applications de gestion.

les langages scientifiques, comme le FORTRAN,  
utilisés par des applications de calcul.

le BASIC, lui est un langage à part ; il a été spécialement mis au point pour les non-spécialistes en informatique

Parmi tous ces langages, les plus couramment rencontrés sont :

#### . LE BASIC

BASIC vient de "Beginner's All Purpose Symbolic Instruction Code", qui veut dire Code Symbolique d'Instructions pour tous usages à l'intention des débutants.

#### . le FORTRAN

Son nom vient de FORMULA TRANSLATOR

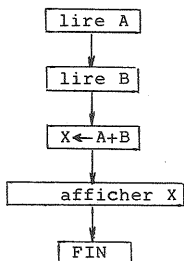
Ce langage date de 1956 ; c'est un langage scientifique qui permet de traiter la quasi totalité des calculs mathématiques.

#### . le COBOL

Son nom vient de de COMMON BUSINESS ORIENTED LANGUAGE.

Il a été créé en 1959 par les utilisateurs et constructeurs d'ordinateurs ; leur objectif était de créer un langage de programmation totalement indépendant de la machine, et d'un apprentissage facile, car les phrases utilisées sont très proches de l'anglais

Ecrivons notre programme de tout à l'heure en FORTRAN et en COBOL :



```

1 INPUT A
2 INPUT B
3 X=A+B
4 PRINT X
5 END
  
```

BASIC

en\_COBOL\_:

```

1 ACCEPT A
2 ACCEPT B
3 ADD A TO B GIVING X
4 DISPLAY X
5 END
  
```

en\_FORTRAN\_:

```

1 INTEGER A, B, X
2 READ ( 2, 6 ) A, B
3 X=A+B
4 WRITE ( 3, 7 ) X
5 END

6 FORMAT ( 2I4 )
7 FORMAT ( 1I5 )
  
```

indique que les variables A, B, et X sont entiers

lecture de A et B sur le périphérique N°2 (clavier) selon le format N°6

sortie de X sur le périphérique N°3 (écran) selon le format N°7

format N°6 : 2I4 veut dire 2 nombres entiers de 4 chiffres

format N°7 : 1I5 veut dire 1 nombre entier de 5 chiffres

*Réalisons quelques petits programmes  
en Basic*



### Rappels et remarques

- 1) Le programme est une suite d'instructions ( explication théorique ) que l'ordinateur va exécuter ; chaque instruction est numérotée dans l'ordre dans laquelle elle doit être exécutée.

exemple :

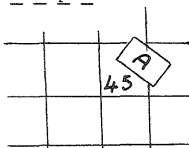
Numéros qui indiquent l'ordre le quel les instructions doivent être exécutées

```
1 INPUT A
2 INPUT B
3 X=A+B
4 PRINT X
5 END
```

Suite d'instructions qui explique à l'ordinateur le travail qu'il aura à accomplir

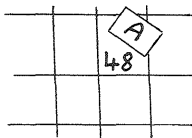
- 2) la mémoire de l'ordinateur est subdivisée en "cases" ; chaque "case mémoire" est réservée à une variable ; dès que l'on affecte à une variable une valeur, la valeur précédente disparaît ; c'est-à-dire que dans une "case mémoire" ou variable il ne peut y avoir qu'une valeur ; lorsqu'on place dans cette "case mémoire" ou variable une autre valeur, la précédente sera détruite.

exemple :



soit en mémoire, une " case " ( ou variable ) appelée A, dans laquelle se trouve le nombre 45

Après avoir exécuté l'instruction A=48 par exemple, il y aura la valeur 48 dans la "case mémoire" A ; le nombre 45 aura disparu !



3) Dans toute résolution de problème, notre ordinateur devra pouvoir :

- lire des données
- traiter ces données
  - calculer
  - comparer, tester
- sortir les résultats

Les 4 instructions BASIC permettant d'effectuer ces opérations sont :

INPUT pour lire      exemple: INPUT A

PRINT pour sortir les résultats    exemple :PRINT X

.....= calcul pour calculer    exemple X=A+B

IF condition THEN instruction pour tester  
exemple : IF X<3 THEN PRINT X

. Voyons ces 4 instructions de base ( Bien sûr, le langage Basic comprend bien plus que 4 instructions ! )

#### 1) Instruction de lecture : INPUT

- . exemple : l'ordinateur doit lire une variable appelée A
- . cette instruction s'écrit dans l'organigramme : lireA
- . cette instruction s'écrit en BASIC : INPUT A
- . effet : lorsque l'ordinateur exécute cette instruction, il lira sur son unité d'entrée ( clavier par exemple ) la valeur qu'on lui fournit et la placera dans une case mémoire qu'il appellera A.



2) instruction de sortie : PRINT

- . exemple : l'ordinateur doit afficher une variable appelée X
- . cette instruction s'écrit dans l'organigramme :  

afficher X
- . cette instruction s'écrit en BASIC : PRINT X
- . effet : lorsque l'ordinateur exécute cette instruction, il recherchera le contenu de la "case mémoire" X et le sortira sur l'unité de sortie ( affichage sur l'écran par exemple )
- . remarque : si l'on veut afficher un texte et non une variable, par exemple le mot BONJOUR, on écrira ce texte entre ""  

exemple : PRINT "BONJOUR" l'ordinateur affichera alors tout simplement le texte qui se trouve entre ""

3) instruction de calcul : .....=calcul

- . exemple : si R représente le rayon d'un cercle, calculer son diamètre D
- . cette instruction s'écrit dans l'organigramme :  

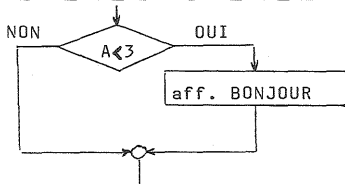
$D \leftarrow R * 2$
- . Remarque : le signe \* représente la multiplication et le signe / représente la division.
- . cette instruction s'écrit en BASIC : D=R\*2
- . effet : lorsque l'ordinateur exécute cette instruction, il cherchera en mémoire le contenu de la case R ; le multipliera par 2 et placera le résultat de ce calcul dans une case mémoire qu'il appellera D.
- . Remarques : . dans cette opération, le contenu de R reste inchangé, tandis que si une case mémoire appelée D avait déjà existée avant l'exécution de l'instruction, son contenu serait détruit et remplacé par la nouvelle valeur

. dans une opération de calcul, ce qui se trouve à droite du signe "égal" indique à l'ordinateur ce qu'il doit calculer, et ce qui se trouve à gauche du signe "égal" lui indique où il doit ranger le résultat du calcul.

4) instruction de test : IF.....THEN.....

. exemple : afficher le mot BONJOUR dans le cas seulement où la variable A est plus petite que 3.

. cette instruction s'écrit dans l'organigramme :



. cette instruction s'écrit en BASIC :  
IF A < 3 THEN PRINT "BONJOUR"

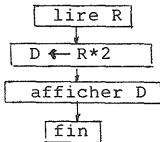
. effet : lorsque l'ordinateur exécute cette instruction, il compare le contenu de la "case mémoire" A à la valeur 3 ; si A est plus petit que 3 et seulement dans ce cas, il exécutera l'instruction PRINT "BONJOUR" ( c'est à dire il affiche le mot BONJOUR )

. remarque : . la structure générale de l'instruction de test est :  
IF condition THEN instruction basic  
si la condition est vérifiée, l'ordinateur exécute l'instruction basic  
... et si l'on sait que IF veut dire SI et que THEN veut dire ALORS, on comprend tout à fait le sens de cette instruction.

. écrivons quelques programmes

- 1) l'ordinateur doit lire le Rayon d'un cercle, calculer le diamètre et afficher le résultat

organigramme :



programme :

```

1 INPUT R
2 D=R*2
3 PRINT D
4 END
  
```

pour que le résultat affiché soit plus compréhensible, on pourrait écrire le programme de la manière suivante :

```
1 INPUT R
```

```
2 D=R*2
```

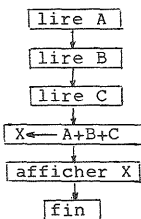
```
3 PRINT "LE DIAMETRE EST"
```

```
4 PRINT D
```

```
5 END
```

ces 2 instructions  
peuvent s'écrire en une  
seule :  
PRINT "LE DIAMETRE EST" ; D

- 2) l'ordinateur doit lire 3 nombres et afficher le résultat de l'addition des 3 nombres



```
1 INPUT A
```

```
2 INPUT B
```

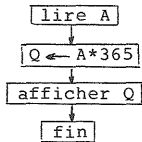
```
3 INPUT C
```

```
4 X=A+B+C
```

```
5 PRINT X
```

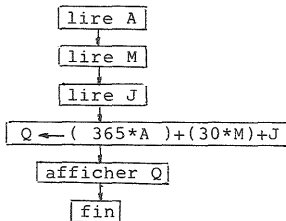
```
6 END
```

- 3) l'ordinateur doit lire votre âge et calculer le nombre de jours environ que vous avez vécu.



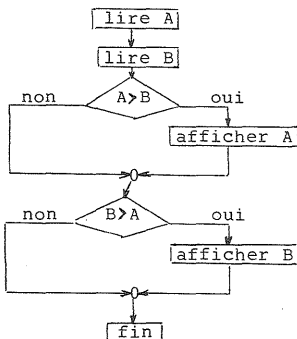
1 INPUT A  
2 Q=A\*365  
3 PRINT Q  
4 END

- 4) l'ordinateur doit lire votre âge en année, mois et en jour ; il calcule le nombre de jour environ que vous avez vécu. ( calcul plus précis que dans l'exercice 3 )



1 INPUT A  
2 INPUT M  
3 INPUT J  
4 Q=(365\*A)+(30\*M)+J  
5 PRINT Q  
6 END

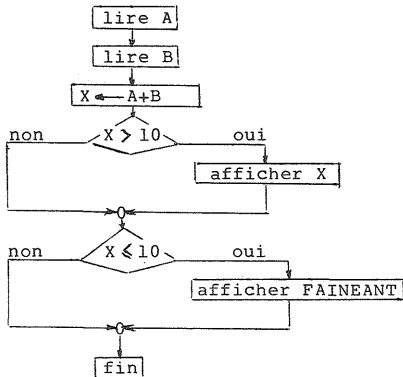
- 5) l'ordinateur doit lire 2 nombres et afficher le plus grand des deux.



1 INPUT A  
2 INPUT B  
3 IF A>B THEN PRINT A  
4 IF B>A THEN PRINT B  
5 END

Remarque : dans le cas où A=B, l'ordinateur n'affichera rien car dans le programme on n'a pas prévu ce cas !

- 6) l'ordinateur doit calculer l'addition de 2 nombres qu'il aura lu ; il doit afficher le résultat que s'il est plus grand que 10 ; dans le cas contraire il doit nous répondre : FAINEANT



```

1 INPUT A
2 INPUT B
3 X=A+B
4 IF X >10 THEN PRINT X
5 IF X<=10 THEN PRINT
  "FAINEANT"
6 END

```

- 7) l'ordinateur lit le sexe (1 pour masculin, 2 pour féminin), la taille ( en cm ) et le poids ( en kg ) ; il doit faire le diagnostic de votre poids

```

1 INPUT S
2 INPUT T
3 INPUT P
4 IF S=1 THEN PI=T-100
5 IF S=2 THEN PI=T-110
6 IF PI > P THEN E=PI-P
7 IF PI < P THEN E=P-PI
8 IF P > PI THEN PRINT E ; "KG A PERDRE "
9 IF P < PI THEN PRINT E ; "KG A PRENDRE"
10 IF P=PI THEN PRINT "BRAVO"
11 END

```

- . les instructions 1, 2 et 3 lisent les 3 variables :

S ( le sexe )    T ( la taille )    P ( le poids )

- . instructions 4 et 5 : calcul du poids idéal ( PI ) en fonction du sexe ( S )

si le sexe est masculin ( S=1 ), alors le poids idéal (PI)  
est égal à la taille moins 100  
si le sexe est féminin ( S=2 ), alors le poids idéal (PI)  
est égal à la taille moins 110

- . instructions 6 et 7 : calcul de l'écart (E) entre le poids (P)  
et le poids idéal (PI)  
si le poids idéal (PI) est plus grand que le poids (P)  
- l'écart (E) se calcul en faisant PI-P  
si le poids idéal (PI) est plus petit que le poids (P)  
l'écart (E) se calcul en faisant P-PI

- . instructions 8, 9 et 10 : affichage du diagnostic  
si le poids (P) est supérieur au poids idéal (PI), il  
faudra perdre E kilos

si le poids (P) est inférieur au poids idéal (PI), il  
faudra prendre E kilos

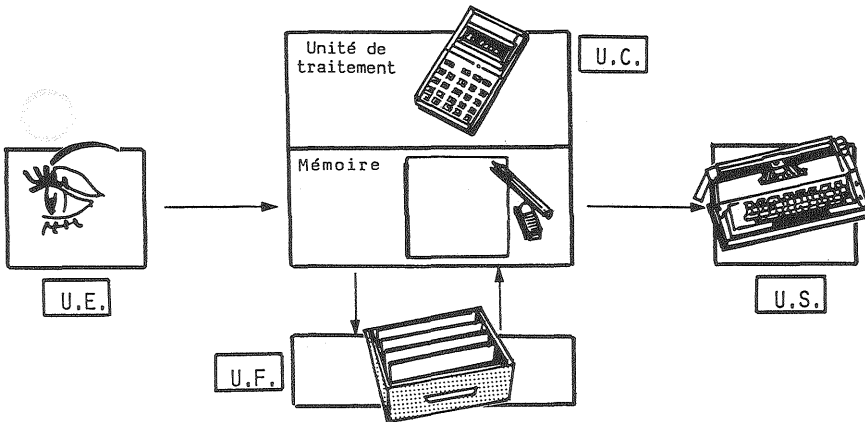
si le poids (P) est égal au poids idéal, Bravo !

*Résumé*





Jusqu'à présent, nous avons considéré l'ordinateur comme un ensemble de plusieurs unités fonctionnelles :



- Une ou plusieurs unités d'entrées (UE).

Son rôle est de lire les données à traiter ( mais également le programme qui permettra de traiter ces données ). Les données et les instructions du programme, si elles ne sont pas directement tapées sur clavier, sont préalablement saisies sur un support (carte perforée, disquette,...) ; à chaque support correspond une unité d'entrée capable de lire ce support.

Exemple : Si nous avons enregistré les informations sur cartes perforées, nous devons utiliser un lecteur de cartes perforées comme unité d'entrée pour pouvoir les lire.

Chaque unité d'entrée réalise les fonctions suivantes :

- reconnaître les informations enregistrées sur le support
- transformer ces informations sous forme d'impulsions électriques.
- envoyer ces impulsions dans la mémoire de l'unité centrale où elles seront stockées.

- Une ou plusieurs unités de sorties (US)

Cette unité permet de "sortir" les informations stockées en mémoire, sur un écran cathodique pour les visualiser ou sur une imprimante pour les imprimer par exemple.

- Eventuellement une ou plusieurs unités de fichiers (UF)

Il existe de nombreuses unités de fichier : unité de cassette ou de bande magnétique, unité de disque ou disquette magnétique,...

Ces unités jouent un double rôle :

- mémoriser ( stocker ) sur supports ( cassettes, bandes, disques, disquettes... ) les informations reçues de l'unité centrale
- pouvoir restituer ultérieurement à l'unité centrale les informations enregistrées

l'unité de fichier est en fait une unité d'entrée/sortie ( U E/S ) car l'unité centrale peut y lire des informations, mais également lui en envoyer afin que l'unité de fichier les enregistre et les conserve.

- Une unité centrale (UC)

L'unité centrale se compose de deux ensembles :

- . la mémoire : elle permet de stocker les instructions du programme, les données lues par l'unité d'entrée et les résultats après traitement ( calcul ) dans l'unité de traitement
- . l'unité de traitement : Elle permet d'effectuer différentes opérations de calcul et de test sur les données stockées en mémoire ; après traitement, les résultats seront également stockés en mémoire

- Pour que l'ensemble de ces unités puissent fonctionner, il est nécessaire de stocker en mémoire un programme.

- . un programme est une suite d'instructions qui expliquent à l'ordinateur d'une façon précise et élémentaire le travail qu'il aura à accomplir
- . les instructions sont écrites dans un langage compréhensible par l'ordinateur : le BASIC par exemple

---

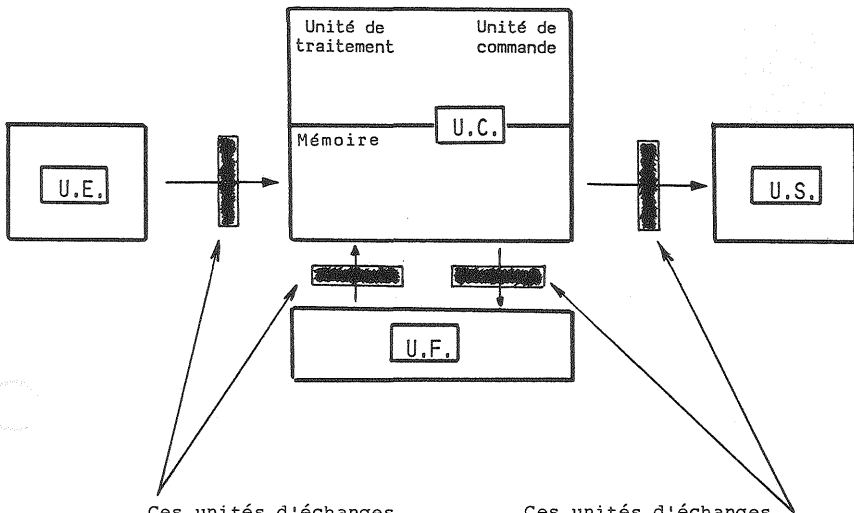
Pour des raisons pédagogiques, cette représentation a été volontairement simplifiée ; mais pour bien comprendre le fonctionnement d'un ordinateur, complétons maintenant ce schéma un peu simpliste

- 1) Il est impossible de brancher directement notre unité d'entrée, notre unité de sortie ou notre unité de fichier sur l'unité centrale.

En effet, chaque unité d'entrée délivre des impulsions électriques différentes, et chaque unité de sortie pour qu'elle puisse fonctionner devra recevoir des signaux électriques également différents

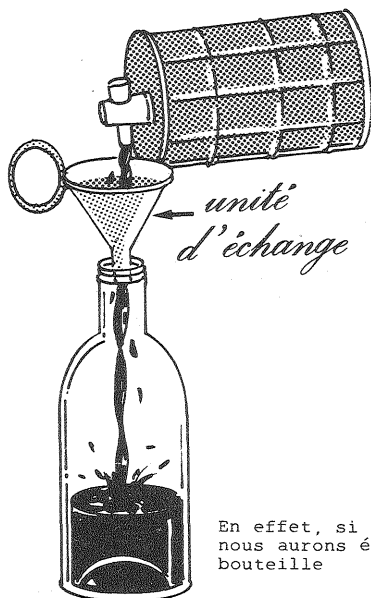
Il faudra donc intercaler entre l'unité d'entrée et l'unité centrale, entre l'unité centrale et l'unité de sortie et entre l'unité de fichier et l'unité centrale, un appareil ( appelé parfois unité d'échange, interface ou coupleur ) qui aura pour rôle de transformer les signaux électriques.

C'est comme si nous voulions brancher un rasoir 110 volts sur une prise de courant 220 volts, nous devrions utiliser un transformateur!



Ces unités d'échanges transforment le signal venant de l'unité d'entrée ou de l'unité de fichier afin que ces signaux soient compréhensibles par l'unité centrale.

Ces unités d'échanges transforment le signal électrique venant de l'unité centrale afin d'être utilisable, par l'unité de sortie ou l'unité de fichier

REMARQUE :

Ces unités d'échanges ne changent en rien les informations venant de l'unité d'entrée ou de l'unité de fichier vers l'unité centrale ou qui vont de l'unité centrale vers l'unité de sortie ou vers l'unité de fichier ; ce n'est que la représentation du signal électrique qui est modifiée.

En effet, si le bidon contient du Beaujolais, nous aurons également du Beaujolais dans la bouteille

- 2) Enfin, dans notre schéma initial, nous avons supposé que lorsque le programme se trouve en mémoire, celui-ci va se "dérouler" (s'exécuter) comme par miracle !

En réalité, chaque instruction doit être analysée par l'ordinateur ; et en fonction de cette analyse, l'ordinateur devra déclencher une suite d'opérations pour que l'instruction soit exécutée.

Julien devrait également, pour chaque instruction, effectuer une suite d'opérations.

L'instruction calculer STOCK RESTANT=STOCK-QUANTITE COMMANDEE nécessite de la part de Julien toute une série d'opérations :

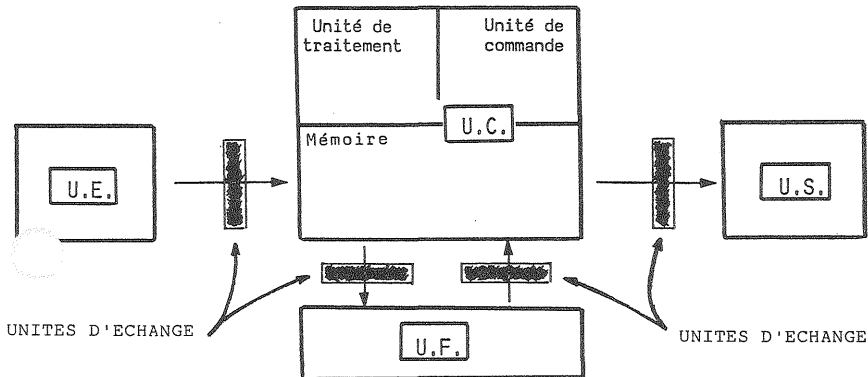
- prendre le "stock" noté sur la feuille de brouillon et l'introduire dans la machine à calculer.
- appuyer sur la touche ⊖
- prendre la "quantité commandée" notée sur la feuille de brouillon et l'introduire dans la machine à calculer
- appuyer sur la touche =
- prendre le résultat et le noter sur la feuille de brouillon

Tous ces détails n'étaient pas précisés dans le "programme" donné par le responsable de facturation ; c'est en fait une partie du cerveau de Julien qui "reconnait" l'instruction et qui déclenche cette suite d'opérations

Pour l'ordinateur, les choses se passent de la même manière : le rôle de l'unité de commande est de sélectionner une instruction après l'autre du programme, de l'analyser et d'envoyer aux différents circuits électroniques les impulsions nécessaires pour arriver au résultat. L'instruction  $SR=S-Q$  par exemple, déclenchera dans l'unité de commande les opérations suivantes :

- chercher la "case mémoire" appelée S
- prendre le contenu de cette case et l'envoyer dans l'unité de traitement
- chercher la "case mémoire" appelée Q
- prendre le contenu de cette case et l'envoyer dans l'unité de traitement
- effectuer la soustraction
- etc.....

LE SCHEMA COMPLET DE NOTRE ORDINATEUR DEVIENT DONC :



NOUS ALLONS DANS LE CHAPITRE SUIVANT ANALYSER EN DETAIL CHACUNE DE CES FONCTIONS



*Compléments techniques*





Remarque : il n'est pas nécessaire de comprendre entièrement ce chapitre, qui est très technique, pour pouvoir programmer !

### 1) la mémoire

#### . une très grande variété de codes à mémoriser

Nous avons l'habitude d'utiliser un grand nombre de signes pour transmettre des informations, des concepts, des ordres, ... :

- les lettres de l'alphabet : A, B, C, ..... , Z
- les chiffres : 0, 1, ..... 9
- les signes : ; ! ? + - =, .....

Pouvoir mémoriser une telle variété de symboles pose des problèmes technologiques !  
Il a fallu trouver une méthode pour représenter les informations, les concepts et les ordres avec un plus petit nombre de signes : le Binaire par exemple

Le système binaire ( base 2 ) est utilisé par la plupart des ordinateurs ;  
Un chiffre écrit en binaire, s'écrit seulement avec 2 symboles : le 0 et le 1  
Si nous comptons en base 2, nous obtenons : 0 (zéro unité), 1 (une unité), 10 ( une "deuzaine", zéro unité ), 11 (une "deuzaine", une unité ), 100, 101, 110, 111, 1001, ... etc ... Le tableau ci-dessous donne la valeur en binaire des dix chiffres :

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

De même les lettres de l'alphabet, les signes et les instructions peuvent être codés en binaire

exemple : codification de la lettre A : 0100 0001

codification de l'instruction INPUT : 1100 010

L'avantage du système binaire devient alors évident car toute information, toute instruction sera représentée par une suite de "0" et de "1" ; un code simple à mémoriser, à transmettre et à traiter en électronique car le "0" peut être caractérisé par l'absence d'un courant électrique, et le "1" par la présence d'un courant électrique par exemple.

Mais comment mémoriser une information

Mémoriser une information consiste à stocker provisoirement ou définitivement cette information et de pouvoir la relire.

La mémoire de l'ordinateur est réalisée par un assemblage de petites cellules élémentaires appelées BITS ( ce terme vient de Binary Digit ) chacun de ces bits ( des tores de ferrite dans le temps, des cellules électroniques sur les ordinateurs d'aujourd'hui ) peut prendre deux états, appelons ces deux états : l'état "0" et l'état "1".

Pour coder en mémoire les différentes informations (lettres, chiffres, signes, instructions,... ) il sera nécessaire de créer un assemblage de plusieurs BITS.

Combien de BITS devons nous assembler pour pouvoir coder toutes ces informations ?

Si nous n'avons qu'1 BIT : 2 codes sont possibles : 0 ou 1

Si nous assemblons 2 BITS : 4 codes sont possibles : 00 ou 01 ou 10 ou 11

Si nous assemblons 3 BITS : 8 codes sont possibles : 000 ou 001 ou 010 ou 011 ou 100 ou 101 ou 110 ou 111.

la formule suivante peut-être utilisée :

Soit  $n$  est le nombre de BITS assemblés, le nombre de codes différents =  $2^n$

Exemple : si  $n$  vaut 5 ( assemblage de 5 BITS ) nous pouvons coder  $2^5$  soit 32 informations ( codes ) différentes:

00000 ou 00001 ou 00010 ou 00011 ou .....

Pour coder une information, il est donc nécessaire de disposer ( juxtaposer ) de  $n$  cellules élémentaires (BITS) ; cet assemblage de  $n$  cellules permettant de représenter (coder) une information (lettre; chiffre,...) s'appelle un MOT.

Si le MOT est formé de 8 BITS, ce qui est le cas de la plupart des ordinateurs on ne parle plus de MOT mais d'OCTET (ou BYTE en anglais) ; en fait, un mot de 8 bits s'appelle OCTET

Exemple : représentation ( codage ) du chiffre 5 sur un mot de 4 BITS : 0101

représentation ( codage ) de la lettre A sur un mot de 8 BITS ( OCTET ) : 0100 0001

représentation ( codage ) de l'instruction INPUT ( instruction de lecture ) sur un mot de 8 BITS ( OCTET ) : 1100 0101

La mémoire est donc formée d'un ensemble de MOTS, chaque mot est formé de n BITS ( si n=8, le mot est appelé OCTET ou BYTE ).

Plus n est grand, plus le codage d'un grand nombre d'informations différentes sera possible ; avec n=8, nous pourrons coder  $2^8$ , soit 256 informations différentes ( lettres minuscules, majuscules, signes, chiffres, instructions, etc..... )

La mémoire est une suite de MOTS, chaque mot pouvant contenir le code binaire d'un caractère ( lettre, chiffre, signe,... )

: — Mot 0 : Mot 1 : Mot 2 : Mot 3 : Mot 4 : — .....

La capacité de stockage d'une mémoire dépend du nombre de mots. Chaque mot ( octet ) porte un Numéro : 0, 1, 2, ..., c'est l'"adresse Mémoire"

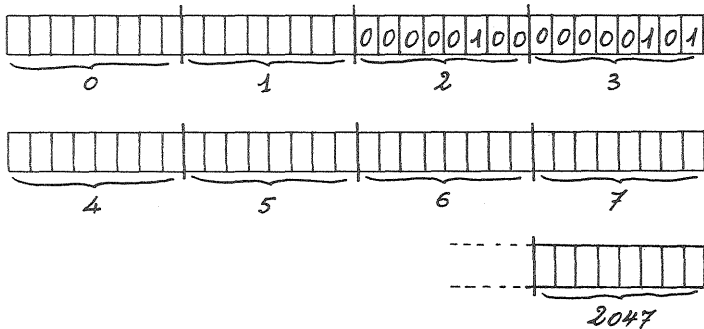
L'unité de mesure de capacité mémoire est le K ( Kilo ) mot ( ou K octet si le mot fait 8 BITS ).

1 K octet représente une capacité de 1024 octets ( 1024 caractères )

Remarque : ce que nous avons appelé ( jusqu'à présent "case mémoire", est un assemblage de plusieurs octets pour y stocker un Nombre.

. Exemple

Soit un "ordinateur 8 bits" de 2 K de mémoire ; c'est à dire que le Mot est un assemblage de 8 bits et que sa capacité est de 2048 octets, numérotés de 0 à 2047.



Pour mémoriser la variable A qui vaut 45 il nous faudra 2 octets :

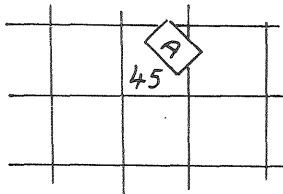
le 4 est mémorisé dans l'octet d'adresse 2 par exemple :  
00000100

le 5 est mémorisé dans l'octet d'adresse 3 par exemple :  
00000101

Plus l'information à mémoriser est "grande", plus il faudra d'octet pour la stocker :

pour stocker le Nombre 1983 par exemple il faudra 4 octets  
pour stocker le mot BONJOUR par exemple il faudra 7 octets

jusqu'à présent nous avons parlé de "cases mémoire"



Nous avons imaginé qu'un nombre se place dans une "case mémoire" et que l'ordinateur y "colle" une étiquette. ( le nom de la Variable ) pour pouvoir le retrouver ; en réalité l'ordinateur se construit un tableau qui lui indique

Variable	Adresse Début	Adresse Fin
A	2	3

d'une manière précise l'endroit où se trouve toutes les variables. Ce tableau étant également stocké en mémoire et codé en binaire

Mais l'avantage aujourd'hui des langages de programmation évolués comme le BASIC est qu'il n'est plus nécessaire de se soucier de tous ces détails : adresse de début et de fin d'une information, place occupée en mémoire,..... ; l'ordinateur gère tout ceci lui même.

#### . Attention

Si votre ordinateur possède une mémoire de 8 K octets par exemple ( soit 8192 octets ) cela ne veut pas dire que vous disposez de 8192 octets pour stocker votre programme et les informations ;

En effet, dans beaucoup de cas, nous le verrons par la suite, l'ordinateur utilise lui-même une partie de sa mémoire pour ses propres "travaux" ; on parle alors de capacité BRUT et de capacité NET.

La capacité BRUT est la capacité réelle de l'ordinateur, par exemple 8 K ( 8192 octets )

La capacité NET est la capacité qui reste disponible pour mettre son programme et les données ( la différence entre le BRUT et le NET peut parfois être très importante ! )

#### . Il existe différents types de mémoires.

- les mémoires RAM ( Random Access Memory ) ( mémoire Vive )  
Dans cette mémoire il est possible d'écrire et de relire les informations. Toute écriture dans cette mémoire, détruit le contenu précédent. Le contenu de cette mémoire est perdu lors d'une coupure de l'alimentation.

- les mémoires ROM ( Read Only Memory ) ( mémoire Morte )  
Cette mémoire ne peut qu'être lue. Son contenu ne disparaît pas lors d'une coupure de l'alimentation.  
La mémoire ROM est enregistrée par le constructeur, c'est à dire, que le contenu de cette mémoire y est "gravé" au moment même de sa fabrication et ne peut être modifié par la suite.

## 2) l'unité de traitement

L'unité de traitement comprend :

- l'unité arithmétique et logique (UAL)

l'unité arithmétique et logique a pour fonction d'exécuter les instructions arithmétiques (addition...) et les instructions logiques (comparaison...).

- Quelques registres de travail qui sont des mémoires à accès rapide, ce qui permet d'accélérer les opérations de calcul.

Remarque : l'un des registres est souvent appelé Accumulateur.

- Le registre condition ou d'état ( en anglais FLAG )

qui reflète l'état d'une opération arithmétique ( ce registre indique par exemple si le résultat est nul, ou positif, ou s'il y a eu dépassement de capacité...).

## 3) l'unité de commande

L'unité de commande comprend :

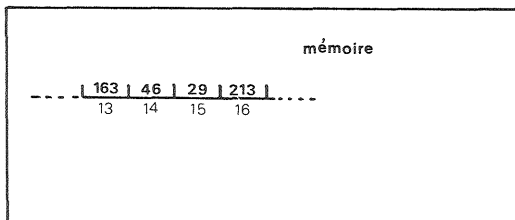
- le compteur ordinal ( en anglais Program Counter (PC) )

qui contient l'adresse de la prochaine instruction à exécuter.

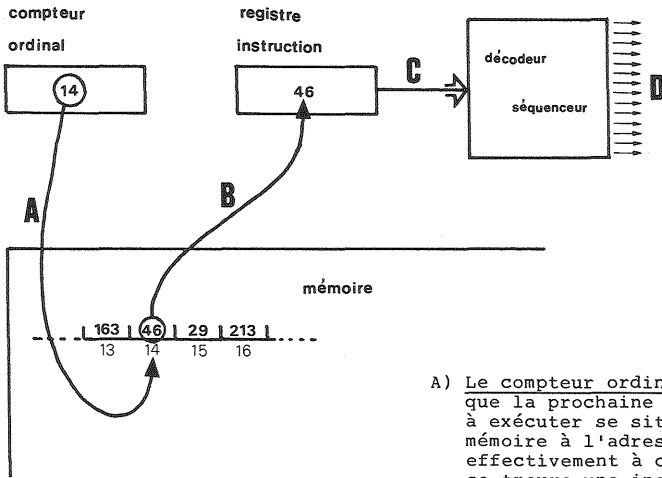
- le registre instruction où sera stockée l'instruction à exécuter

- le décodeur d'instruction et séquenceur qui analyse l'instruction et déclenche un train d'ordres vers les différents composants électroniques qui devront intervenir pour réaliser l'instruction.

Notre programme se trouve stocké sous forme de codes dans la mémoire de l'unité centrale

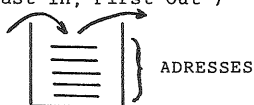


Nous voyons dans les octets 13, 14, 15, 16,... les différentes instructions 163, 46, 29, 213 par exemple.



- A) Le compteur ordinal indique que la prochaine instruction à exécuter se situe dans la mémoire à l'adresse 14, effectivement à cette adresse se trouve une instruction codée 46, (instruction de lecture par exemple ).
- B) Cette instruction est chargée dans le registre instruction
- C) Envoyée dans le décodateur séquenceur celui ci analyse le code de cette instruction
- D) Puis ce dernier envoie les différentes impulsions électriques pour exécuter cette instruction
  - Puis le compteur ordinal s'incrémente de 1 (c'est à dire passe à 15) et le cycle recommence.

. La pile : Elle fait également partie de l'unité de commande ; dans la pile, l'ordinateur range des adresses mémoires ; le fonctionnement d'une pile, veut que la dernière adresse rentrée, soit la première sortie ; c'est le système LIFO ( Last In, First Out )

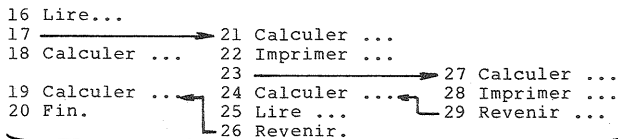


mais pour comprendre sa signification, il est nécessaire de définir la notion de sous-programme.

Exemple

Adresse mémoire des instructions	Instructions
16	Lire ...
17	Aller à l'instruction à l'adresse 21 (sous-programme)
18	Calculer ...
19	Calculer ...
20	Fin de programme
21	Calculer ...
22	Imprimer ...
23	Aller à l'instruction à l'adresse 27 (sous-programme)
24	Calculer ...
25	Lire ...
26	Revenir
27	Calculer ...
28	Imprimer ...
29	Revenir

Dans ce programme nous demandons à l'instruction d'adresse 17, de passer à l'instruction d'adresse 21 ; puis à l'instruction d'adresse 23, de passer à l'instruction 27.



Ceci est appelé  
PROGRAMME PRINCIPAL.

Ceci sont 2 SOUS PROGRAMMES

.Un sous-programme est une partie de programme.  
 .Une instruction (aller à...) permet d'aller dans un sous-programme, une autre instruction (revenir) permet de revenir dans notre sous-programme ou programme de départ.  
 .A quoi sert un sous-programme ? Vous verrez par la suite en programmant que cette technique permet de simplifier très souvent le programme.



Exemple : Un calcul complexe qui "revient" plusieurs fois dans un programme ; il suffira de l'écrire une fois, sous forme de (sous-programme) et on ira dans ce sous-programme chaque fois qu'on veut effectuer ce calcul.

### Problème.

Avec de tels sous-programmes, notre compteur ordinal ne pourra plus s'en sortir :  
le compteur ordinal s'incrémentent normalement de 1, il passe donc de 16 à 17, puis il passe à 21 car l'instruction précise qu'il faut aller à l'adresse 21, puis s'incrémentant normalement de 1, il passe à 22, 23, puis, ensuite 27 puis 28, 29? puis ???!!! Problème !

Il aurait fallu mémoriser les adresses de retour. C'est le rôle de la pile chaque fois que nous irons dans un sous-programme, la pile gardera (mémorisera) l'adresse de retour:

Le compteur ordinal passe de 16 à 17, puis 21, ( à ce moment, la pile stockera l'adresse de retour : 18 ), puis le compteur ordinal passe à 22, 23, puis 27, ( là encore, la pile stockera l'adresse de retour 24 ), puis le compteur ordinal passe à 28, puis 29, puis, grâce à la pile, la suite du traitement est possible, car elle a stocké (empilé), les adresses de retour :

<div style="border: 1px solid black; padding: 2px; display: inline-block;"> Pile 24 18 </div>	Chaque fois que l'on rencontre l'instruction "revenir", le compteur ordinal cherchera l'adresse dans la pile.
-------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------

Le compteur ordinal fonctionnera donc de la manière suivante :

- . on se trouve sur une instruction "Normale" à l'adresse X ;  
le compteur ordinal s'incrémente de 1 ; il passe à X+1
- . on se trouve sur une instruction à l'adresse X qui dit  
"aller à" Y ; le compteur ordinal prend l'adresse Y, et  
charge en Pile l'adresse X+1
- . on se trouve sur une instruction "Revenir" ; le compteur  
ordinal prend la dernière adresse placée dans la pile.

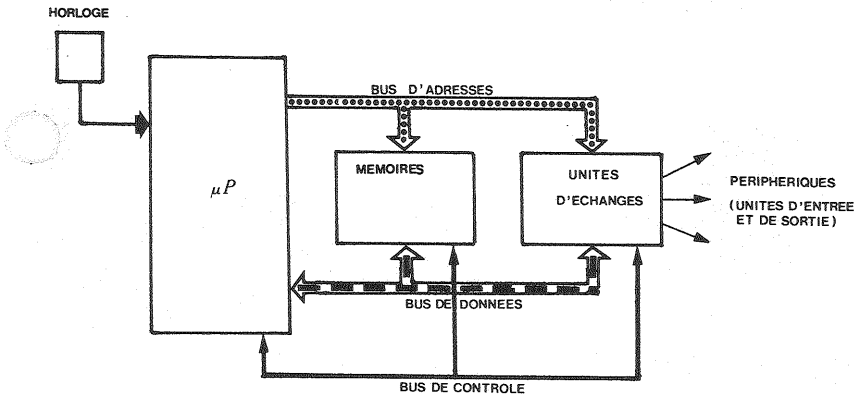
#### 4) Le microprocesseur et son environnement

- . Un micro-ordinateur est un ordinateur construit autour d'un microprocesseur
- . Le microprocesseur, puce de quelques millimètres carré comprend des centaines de milliers de composants électroniques; il comprend la plupart des circuits de notre unité centrale :
  - l'unité de traitement
    - . l'unité arithmétique et logique
    - . les registres de travail
      - . accumulateur
      - . ...
    - . le registre condition
  - l'unité de commande
    - . le compteur ordinal
    - . le registre instruction
    - . le décodeur-séquenceur

Certains microprocesseurs comprennent même de la mémoire RAM et de la mémoire ROM ( de petite capacité bien sûr ( 4K de ROM et 300 octets de RAM par exemple )

Nous allons voir comment le microprocesseur va "dialoguer" avec les autres circuits qui l'entourent c'est-à-dire avec la mémoire et les périphériques ( par l'intermédiaire des unités d'échanges ).

• SCHEMA D'UN MICRO-ORDINATEUR



le microprocesseur (  $\mu P$  ) comprend l'unité de traitement et l'unité de commande

l'horloge délivre des tops de synchronisation à l'ensemble des circuits électroniques afin que ces circuits fonctionnent en harmonie. ( c'est un peu comme le chef d'orchestre devant ses musiciens ).

Un bus C'est un ensemble de fils électriques qui remplissent une fonction précise.

Le Bus Données Sur ce bus, circulent les données entre le  $\mu P$  et la mémoire ou entre le  $\mu P$  et les unités d'échanges ( c'est-à-dire les périphériques ).

Le Bus Adresse Sur ce bus circule, soit l'adresse mémoire, soit l'adresse du périphérique où les données doivent être lues ou écrites

Le Bus Contrôle Dans la forme simplifiée, ce bus indique si le microprocesseur veut lire ou écrire.

Exemple :

Après calcul, le  $\mu P$  veut ranger le résultat de l'opération ( exemple 59 ) en mémoire à l'adresse 1731 :

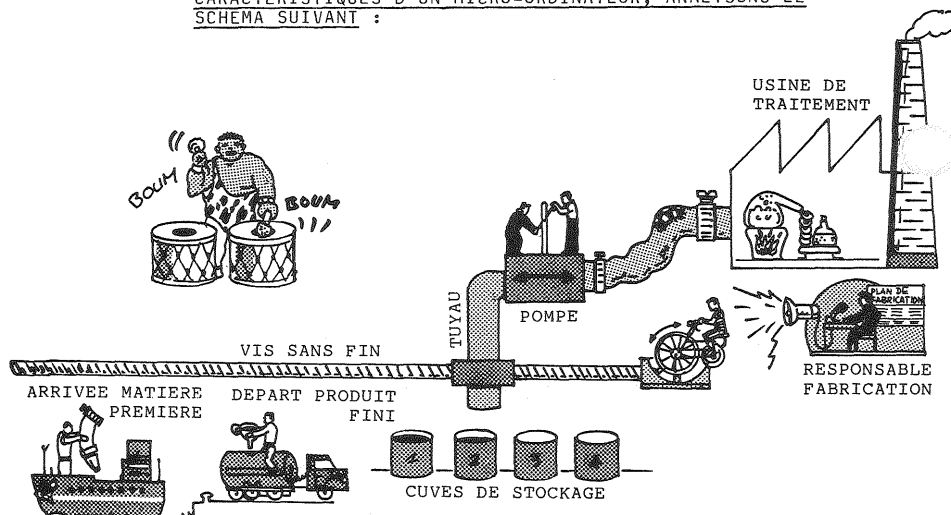
Sur le bus adresse il enverra 1731

Sur le bus donnée il enverra 59

Sur le bus contrôle il enverra 1 ( qui signifie écrire par exemple ).

Ceci entraînera la mémorisation de la valeur 59, dans la mémoire à l'adresse 1731.

MAIS POUR BIEN COMPRENDRE LE FONCTIONNEMENT ET LES CARACTERISTIQUES D'UN MICRO-ORDINATEUR, ANALYSONS LE SCHEMA SUIVANT :



#### VISITE D'UNE USINE DE FABRICATION DE PARFUM

- . Un tuyau relie l'usine de traitement, à une série de cuves de stockage et aux véhicules transportant des matières premières et finies
  - . C'est grâce à une vis sans fin, que notre tuyau pourra se positionner.
  - . C'est grâce au raccord et à l'entonnoir tenus par les deux personnages, qu'on pourra charger ou décharger les produits
  - . Une pompe bidirectionnelle permet de véhiculer les liquides dans les deux sens
  - . Le responsable de fabrication donne les ordres nécessaires pour transformer les matières en produits finis
  - . L'homme au tambour assure un rythme constant au travail
- Vous avez sûrement deviné que notre usine représente le schéma complet d'un micro-ordinateur !?!!

- |                                                                |                     |
|----------------------------------------------------------------|---------------------|
| - l'usine de traitement représente l'unité de traitement       | } <sup>UP</sup>     |
| - le responsable de fabrication représente l'unité de commande |                     |
| - l'homme au tambour représente l'horloge                      | } les bus           |
| - la vis sans fin représente le bus adresse                    |                     |
| - le tuyau représente le bus des données                       |                     |
| - la pompe représente le bus de contrôle                       |                     |
| - les cuves de stockage représentent la mémoire                | } les périphériques |
| - le bateau représente une unité d'entrée                      |                     |
| - le camion représente une unité de sortie                     |                     |
| - le raccord et l'entonnoir représentent les unités d'échanges |                     |

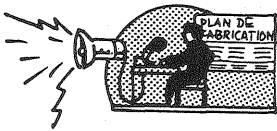
### L'horloge



l'horloge délivre des tops  
afin de synchroniser les  
différents départements de  
notre usine :

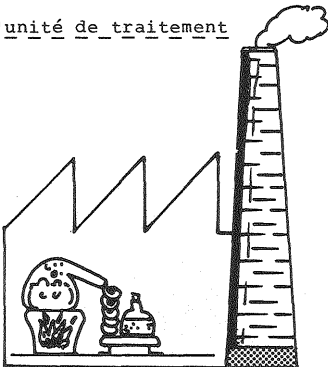
Imaginez que le liquide soit  
envoyé dans le tuyau avant que  
celui-ci soit bien positionné  
au-dessus de la cuve!!!

### L'unité de commande



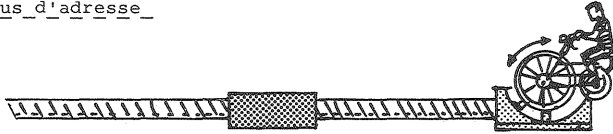
Grâce à un plan de fabrication  
( le programme ), le responsable  
de fabrication  
( le décodeur-séquenceur ) envoie  
les ordres à l'ensemble des  
départements de l'usine.

### L'unité de traitement



Son rôle est la transformation  
( le traitement ) des produits  
( données ) qui lui sont envoyés.

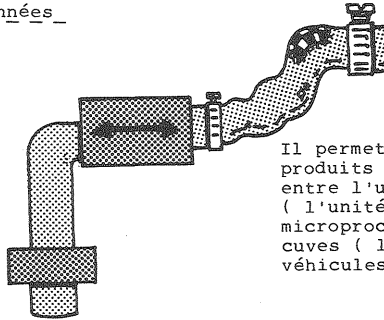
Le microprocesseur (  $\mu P$  ) C'est l'unité de commande + l'unité  
de traitement.

Le bus d'adresse

Il permet de sélectionner la cuve ( la case mémoire ) ou le véhicule ( le périphérique ).

- . Plus la vis sans fin est longue, plus nous pourrons disposer de cuves et de véhicules ; de même, plus le bus adresse comporte de fils, plus nous pourrons adresser une mémoire importante et un nombre de périphériques importants.

Exemple : si le bus adresse comporte 12 fils, le uP peut envoyer  $2^{12}$  codes différents c'est à dire 4096 adresse différentes ; nous pourrions donc utiliser une mémoire de max. 4K

Le bus données

Il permet de véhiculer les produits ( les données ) entre l'usine de traitement ( l'unité de traitement du microprocesseur ) et les cuves ( la mémoire ) ou les véhicules ( les périphériques )

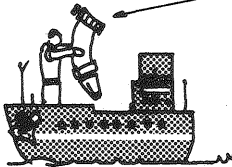
- . Plus le tuyau a un diamètre important plus le débit sera grand ; de même, plus le bus données comporte de fils, plus les opérations de transfert entre le uP et la mémoire ou entre le uP et les périphériques seront rapides.

La mémoire

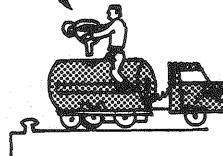
Les cuves (les cases mémoire) permettent de stocker les produits en cours de fabrication (en cours de traitement).

Les périphériques

Unités d'échanges



unité d'entrée

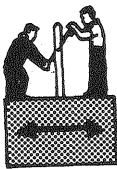


unité de sortie

Les véhicules (les périphériques) permettent d'amener la matière première (entrée des informations) et d'expédier le produit fini (sortie des résultats).

Chaque périphérique dispose de son unité d'échange avec le bus de données

Le bus contrôle



Il permet de véhiculer les produits (les données) dans un sens précis.

### 5) ... Et la matière grise ?

Après avoir vu en détail le fonctionnement du "HARDWARE" ( le matériel ) voyons encore quelques compléments concernant le "SOFTWARE" ( le logiciel )

#### A) langage de programmation

Aussi surprenant que cela puisse paraître, aucun ordinateur ne comprend le BASIC, ni aucun autre langage comme le COBOL, FORTRAN, ... etc.

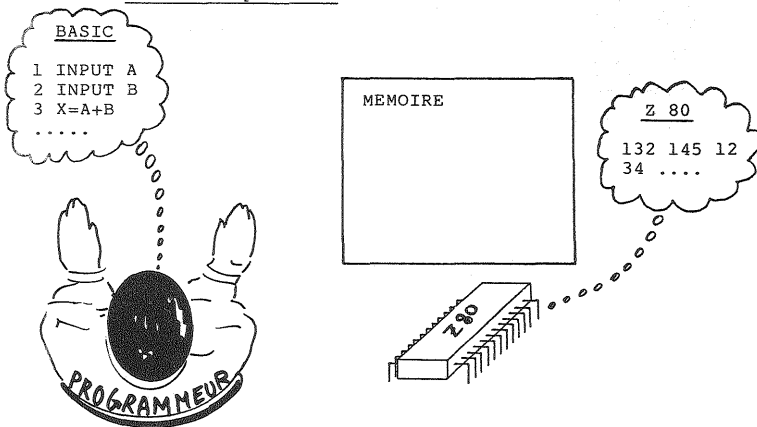
Un ordinateur ne comprend que le langage machine très lié à l'architecture électronique interne ; un ordinateur construit autour du microprocesseur Z80 par exemple, ne comprendra que le langage machine Z80.

Cela veut dire que chaque fois que nous changeons d'ordinateur, il faudra apprendre un nouveau langage ( à moins qu'il ait le même microprocesseur ) ! En plus, le langage machine est très complexe et nécessite une très bonne connaissance de l'architecture de l'ordinateur et du microprocesseur.

... Et que devient le langage BASIC ( et les autres langages ) utilisable sur presque tous les ordinateurs ?

Pour bien comprendre ce qui se passe, reprenons l'exemple de Julien

#### .Posons le problème

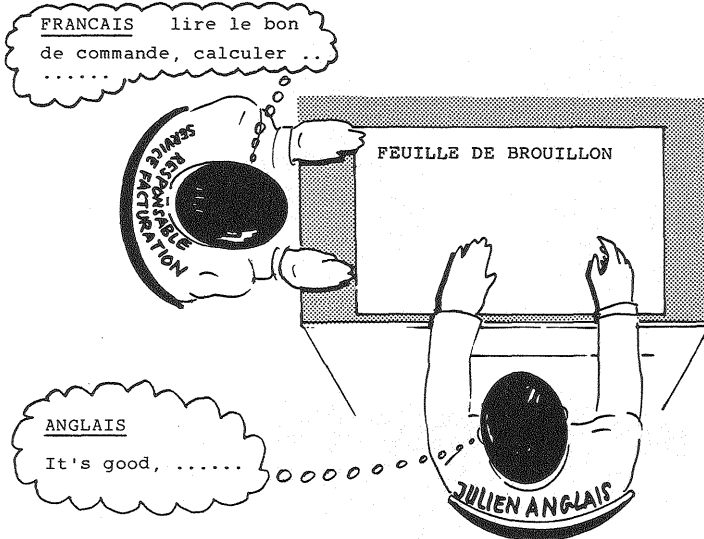


Un programmeur c'est celui qui écrit des programmes ;  
( c'est-à-dire celui qui explique à l'ordinateur ce qu'il doit faire )



- . Nous avons un programmeur qui connaît le BASIC, car c'est un langage simple à utiliser ; le programmeur veut communiquer à l'ordinateur un programme en BASIC ( c'est-à-dire qu'il veut mettre ce programme dans la mémoire de l'ordinateur )
- . Mais le microprocesseur de son ordinateur ( un Z80 par exemple ) ne pourra jamais comprendre ce programme, car il ne comprend que le langage machine Z80.
- . Bien sûr le programmeur peut apprendre le langage du Z80 ; mais c'est long et surtout s'il change d'ordinateur il risque de devoir apprendre un nouveau langage !

- . Nous pourrions comparer cette situation avec :



- . Le chef de service facturation est français ; il veut communiquer à Julien les explications pour qu'il puisse facturer ( c'est-à-dire qu'il doit lui noter sur la feuille de brouillon les instructions nécessaires )
- . Mais Julien est anglais et ne pourra pas comprendre les explications données en français.
- . Bien sûr le chef de service peut apprendre l'anglais, mais c'est long et s'il change de personnel il risque de devoir apprendre de nouvelles langues !

. Y a-t-il une solution pour notre chef de service ?

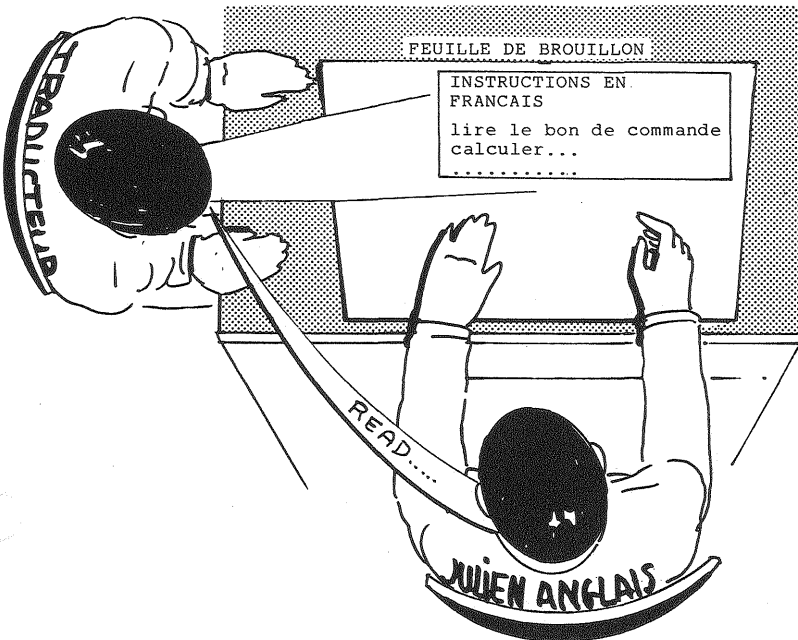
La solution bien sûr a adopter lorsqu'on ne se comprend pas,



est, d'utiliser les services d'un traducteur

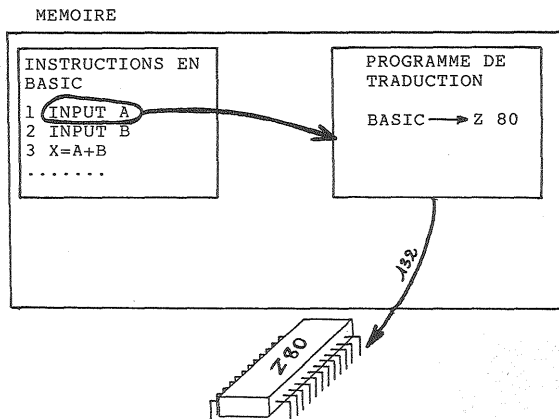


- . et la solution pour notre programmeur ?  
Elle sera la même !  
Nous allons faire appel à un traducteur pour que notre programme écrit en basic puisse être compris par le microprocesseur.  
En fait ce traducteur est lui-même un programme livré par le constructeur qui a pour but de traduire le programme écrit en BASIC, par le programmeur, en un langage compréhensible par l'électronique interne (c'est-à-dire en langage machine Z80 dans notre cas )
- . mais il y a plusieurs manières de traduire :
  - a) -la méthode "au fur et à mesure"



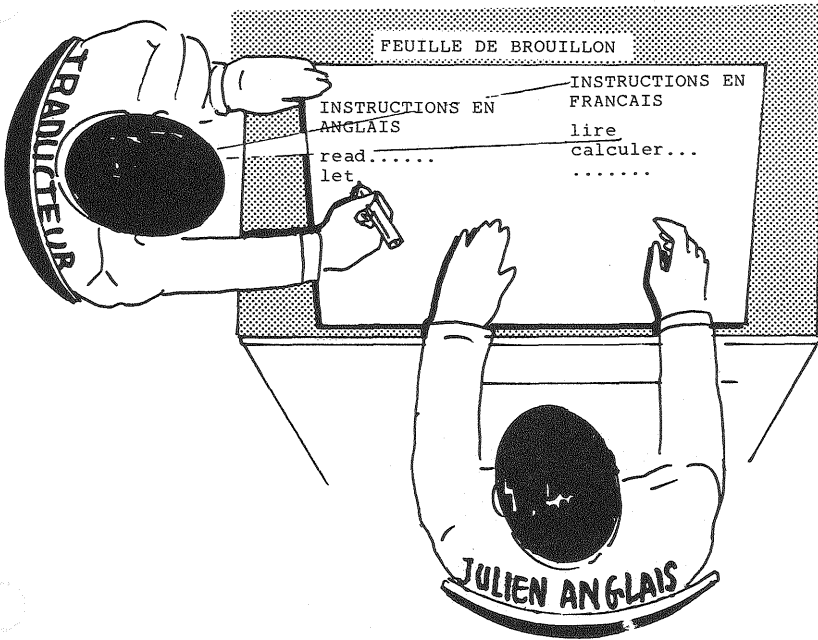
- les instructions sont notées en FRANCAIS sur la feuille de brouillon
- le rôle du traducteur est de prendre une instruction après l'autre et la traduire au fur et à mesure à Julien en ANGLAIS

... le processus est identique pour notre ordinateur :



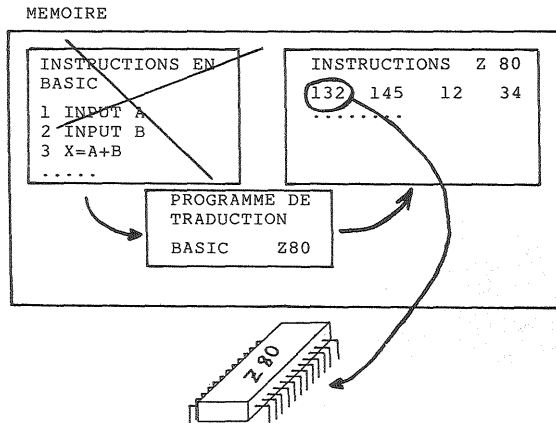
- les instructions sont notées en BASIC dans la mémoire
- le rôle du programme de traduction qu'on appelle INTERPRETEUR dans ce cas est de prendre une instruction après l'autre et la traduire au fur et à mesure au microprocesseur, en langage Z80

b) -la méthode "Globale"



- les instructions écrites en français, sont globalement traduites par le traducteur en ANGLAIS ; les instructions en français sont effacées
- Reste donc sur la feuille de brouillon, les instructions en ANGLAIS directement compréhensibles par Julien

... le processus est identique pour notre ordinateur :



- les instructions écrites en basic, sont globalement traduites par le programme de traduction, qui s'appelle COMPILATEUR dans ce cas ; les instructions écrites en basic sont effacées

- Reste en mémoire les instructions en langage Z80 directement compréhensibles par le microprocesseur

• Avantages et inconvénients du Compilateur et de l'interpreteur

Si nous prenons le cas "Julien" :

- la méthode a ("au fur et à mesure") a l'avantage que les instructions en Français restaient sur la feuille de brouillon ; ainsi, à tout moment, le responsable du service facturation pouvait apporter une modification à ces instructions ; mais son inconvénient est que julien sera freiner dans l'exécution de son travail (le traducteur devant chaque fois lui traduire les instructions)

- la méthode b ("globale") a l'avantage d'être très rapide ; car après la traduction, les instructions sont directement compréhensibles par Julien ; mais l'inconvénient de cette méthode est que le responsable du service facturation ne peut plus intervenir à tout moment car il ne peut pas modifier les instructions en Anglais ; il faudra corriger les instructions en français et refaire traduire l'ensemble par le traducteur !

La comparaison est identique pour notre ordinateur :

INTERPRETEUR : . les corrections, modifications sont très faciles à faire !

. l'exécution du programme est lente !

COMPILATEUR : . les corrections, modifications du programme sont lourdes et longues!

. l'exécution du programme est rapide !

Mais au fait où se trouve ce compilateur ou interpréteur ?

Il s'agit en fait d'un programme écrit par le constructeur et livré avec la machine. Ce programme (interpréteur ou compilateur) se trouve stocké en mémoire ROM ou est livré sur disquette par exemple, qu'il faudra lire et charger en mémoire RAM ; dans ce cas une grande partie de la mémoire RAM peut être utilisée par ce programme, ce qui se traduit en pratique par une diminution importante de la place mémoire disponible.

L'avantage d'avoir un interpréteur ou compilateur stocké en mémoire ROM est que, dès la mise sous tension de l'ordinateur, on pourra commencer à travailler ; dans ce cas il est inutile d'avoir un lecteur de disquette par exemple pour devoir charger la "disquette compilateur ou interpréteur" ; et surtout, dans ce cas nous ne perdons pas une place importante en mémoire RAM.

Mais l'avantage d'avoir son compilateur ou interpréteur sur disquette, est que l'on va pouvoir profiter de toutes les améliorations que le constructeur aura régulièrement put apporter à ces programmes (nouvelles instructions par exemple) ; et pourquoi pas un jour si l'on veut abandonner le BASIC au profit d'un langage plus performant et peut être plus adapté comme le COBOL, il suffira de changer la disquette (on achètera alors le compilateur COBOL)

B) les autres outils

le compilateur et l'interpreteur ne sont pas les seuls outils livrés par le constructeur ; d'autres outils appelés parfois outils de développement, outils d'exploitation ou utilitaires sont souvent disponibles chez le constructeur et permettent souvent un gain de temps et de performance très important dans l'écriture des logiciels.

Tous ces programmes sont stockés en ROM ou livrés sur disquettes, ou cassettes, etc.... à charger en RAM  
Ces programmes peuvent être par exemple :

- programmes de tri de fichier
- gestion automatique de la table pour un fichier en accès sélectif
- jeu d'instructions en Basic pour
  - les calculs complexes
  - les graphismes
  - etc...
- etc...

La quantité d'"intelligence" (de software) livrée par le constructeur avec votre machine peut se calculer de la manière suivante :

( Capacité Brut - Capacité Net ) + Capacité ROM

Ceci vous donne en kilo, la matière grise livrée avec le tas de ferraille

C'est selon moi, aujourd'hui le critère le plus important dans le choix d'un micro-ordinateur.

Bien sûr, le " poids " de cette matière grise ne nous renseigne pas sur sa qualité