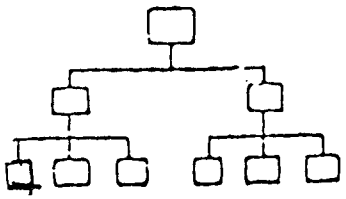
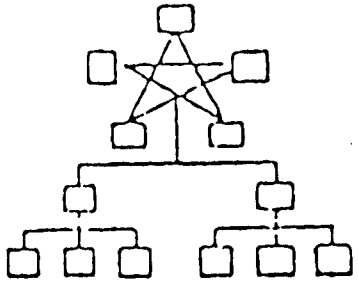
ORGANIZATIONAL CHARTS

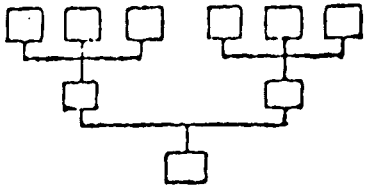
TRADITIONAL



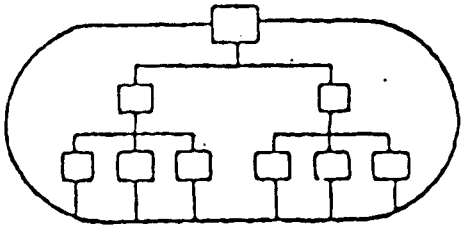
RUSSIAN



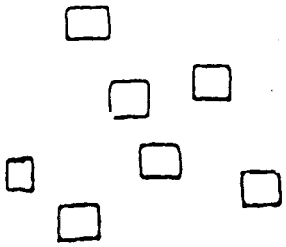
POLISH



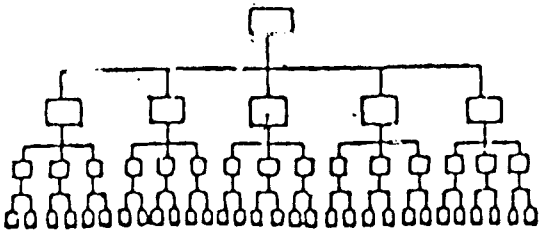
AMERICAN



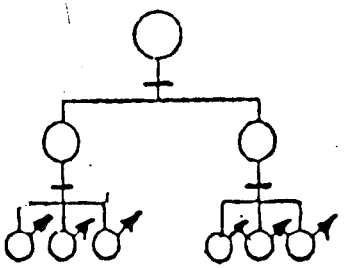
ARAB



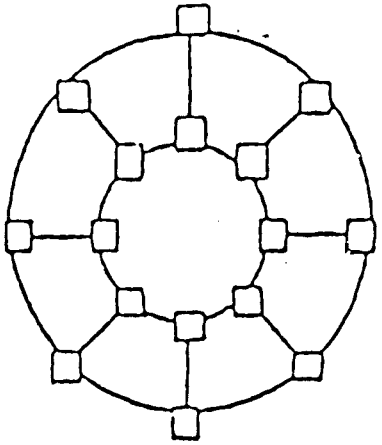
CHINESE



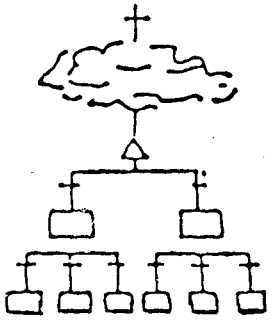
WOMAN'S LIB



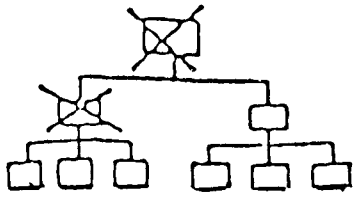
UNITED NATIONS



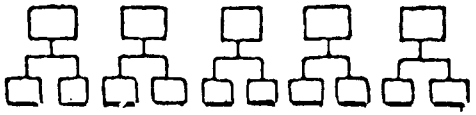
VATICAN



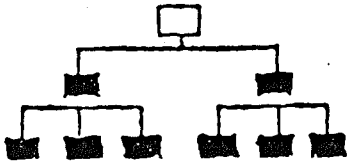
LATIN AMERICAN



ITALIAN

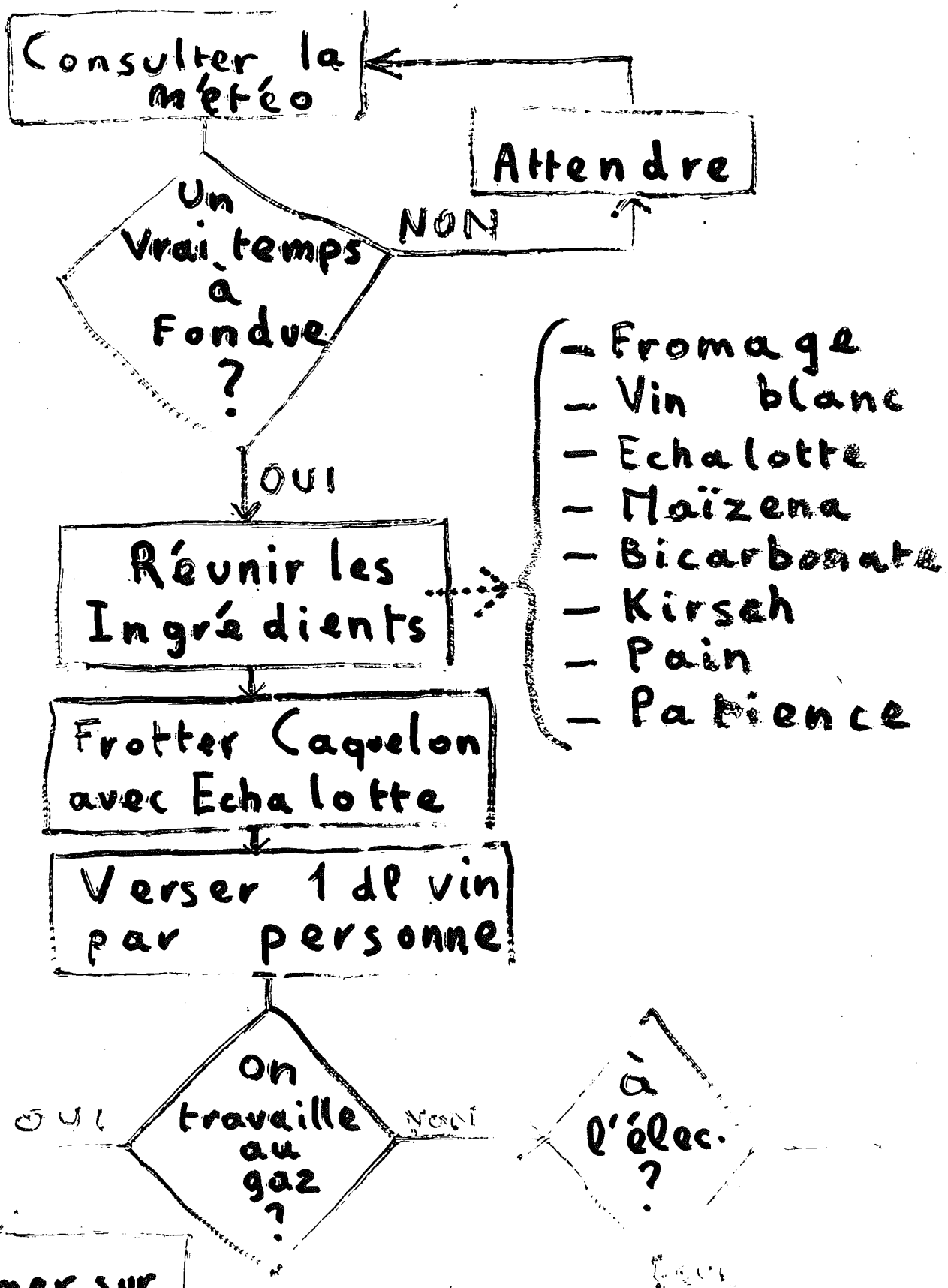


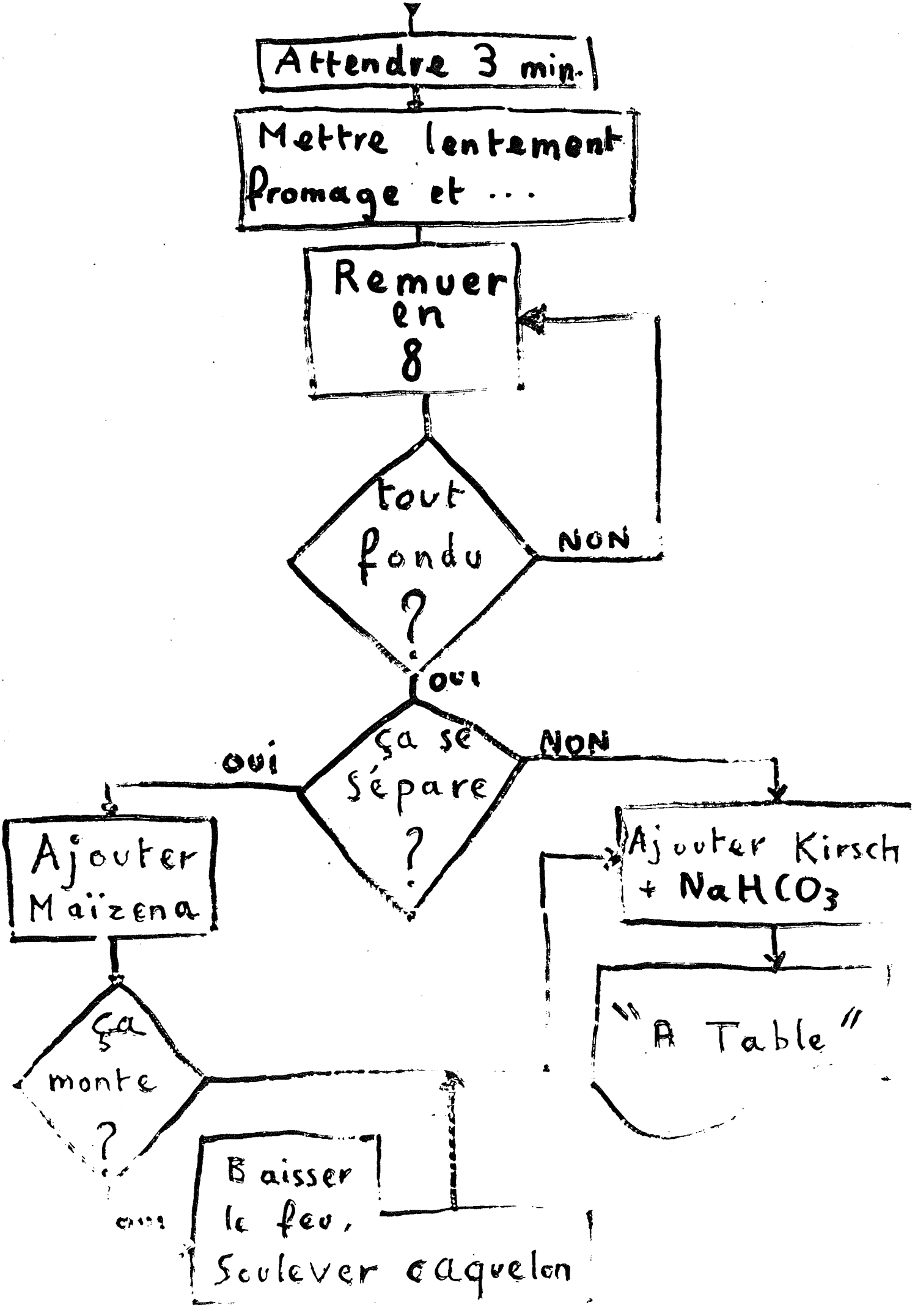
RHODESIA



.C.F.F.

Computers For Fondues.




Attendre 3 min.
Mettre lentement fromage et ...
Remuer en 8
tout fondu ?
NON
OUI
ça se sépare ?
OUI
NON
Ajouter Maïzena
Ajouter Kirsch + NaHCO3
"A Table"
ça monte ?
OUI
Baisser le feu, Soulever caquelon

CHAPITRE 1

ALGORITHMES ET ORDINATEURS

1.1.—ALGORITHMES ET ORGANIGRAMMES

Qu'est-ce qu'un algorithme ? Un *algorithme* est une suite d'instructions pour exécuter un processus en pas à pas. Une recette d'un livre de cuisine est un parfait exemple d'algorithme. La préparation d'un plat compliqué est divisée en étapes simples que chaque personne connaissant la cuisine peut comprendre. Un autre bon exemple d'algorithme se trouve être la chorégraphie d'un ballet classique. Une danse compliquée peut être divisée en une suite de pas de danse et de figures de ballet. Le nombre de ces pas de base et de figures est très faible mais, en les accolant de diverses manières, on peut imaginer une variété infinie de danses.

De même, les algorithmes exécutés par un ordinateur peuvent mettre en jeu des millions de pas élémentaires, tels que des additions et des soustractions dans un calcul mathématique compliqué. Ainsi, au moyen d'algorithmes, un ordinateur peut contrôler le développement d'une usine ou coordonner, dans une compagnie aérienne, les réservations reçues à partir de bureaux dans tout le pays. Les algorithmes de ces processus à large échelle sont, naturellement, très complexes, mais ils sont construits à partir de morceaux tels que ceux de l'exemple que nous allons considérer à présent.

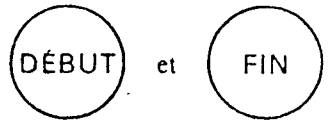
Si nous *pouvons* inventer un algorithme pour une application donnée, nous pouvons ordinairement le faire aussi de différentes manières. Voici un algorithme de tous les jours : le changement d'une roue crevée

1. Soulever la voiture avec le cric.
2. Dévisser les écrous.
3. Enlever la roue.
4. Mettre la roue de secours.
5. Revisser les écrous.
6. Descendre la voiture.

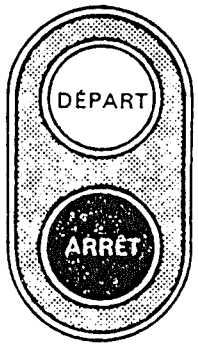
Nous pourrions ajouter de multiples autres détails à cet algorithme. Nous pourrions lui inclure les étapes de la sortie du matériel du coffre, la mise en place du cric, la dépose des enjoliveurs de roue, et le desserrage des écrous avant de monter la voiture sur le cric. Pour des algorithmes de description de processus mécaniques, il est généralement mieux de décider quels détails

inclure. Cependant, les étapes que nous avons écrites seront idéales pour rendre l'idée d'organigramme. Quand nous étudierons les algorithmes mathématiques, nous aurons à être plus précis.

Un *organigramme* est un diagramme qui représente un algorithme. Dans la figure 1.1, nous voyons un organigramme de l'algorithme du pneu crevé. Les



dans l'organigramme nous font penser aux boutons utilisés pour démarrer ou arrêter une machine. Chaque instruction de l'organigramme est enfermée



dans un cadre ou une « case ». Comme nous le verrons bientôt, la forme du cadre indique le type d'instruction qu'il contient. Un cadre rectangulaire signifie un ordre de faire un travail quelconque.

Pour exécuter la tâche décrite dans l'organigramme, nous commençons au bouton de départ et nous suivons les flèches, de case en case, en exécutant les instructions lorsque nous arrivons dessus.

Après avoir dessiné l'organigramme, nous regardons toujours pour voir si nous pouvons l'améliorer. Par exemple, dans l'organigramme du pneu crevé, nous avons négligé de vérifier si la roue de secours était à plat. Si elle est crevée, nous ne changerons pas le pneu et nous appellerons un garage. Ceci demande une décision entre deux possibilités d'action. Pour cela, nous introduisons une nouvelle forme de cadre dans notre organigramme.

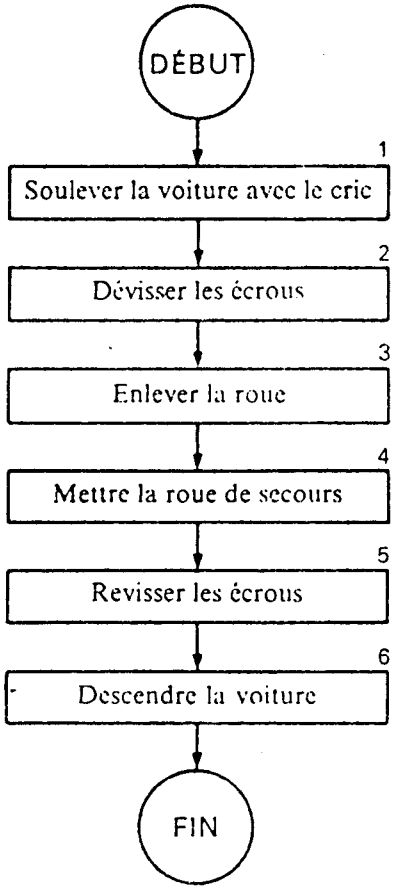
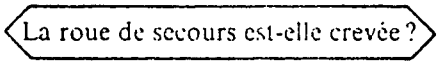
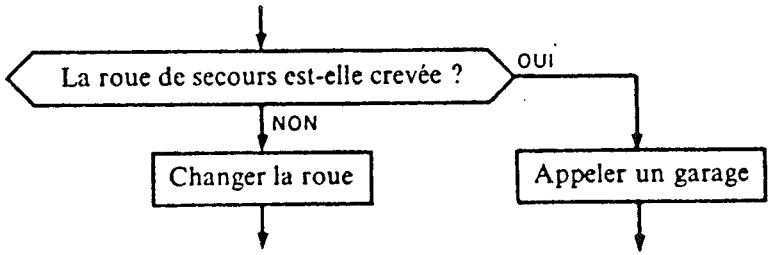


FIG. 1.1. — Premier organigramme de la crevaison.

A l'intérieur, nous écrivons une déclaration à la place d'une commande.

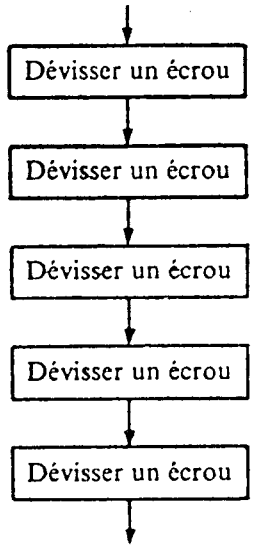


On appelle ceci une *case de test* et nous lui donnerons deux sorties appelées OUI et NON. Après vérification de la validité ou de la fausseté de la déclaration, nous choisissons la sortie adéquate et exécutons ensuite le travail demandé.

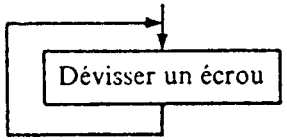


Par l'insertion de ce fragment d'organigramme dans la figure 1.1, nous obtenons l'organigramme de la figure 1.2.

Il existe une autre amélioration intéressante possible. L'instruction de la case 2 de l'organigramme actuel se présente comme un certain nombre de répétitions du même travail. Pour montrer le détail à ajouter, nous pourrions remplacer la case 2 par

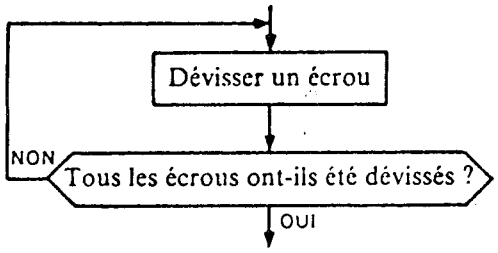


La lourdeur de cette instruction répétée peut être éliminée en introduisant une *boucle*.



Quand nous quittons cette case, nous trouvons que la flèche nous mène tout droit en arrière pour refaire le même travail. Aussi, nous sommes retenus dans une boucle sans fin, étant donné que nous n'avons prévu aucun chemin de sortie pour se diriger vers un nouveau travail. Pour mettre fin à cette situa-

on, nous nous servons d'une case de test comme ci-dessous :



Si on remplace la case 2 de l'organigramme par cela et que l'on effectue même opération sur la case 5, nous obtenons le résultat final décrit dans figure 1.3.

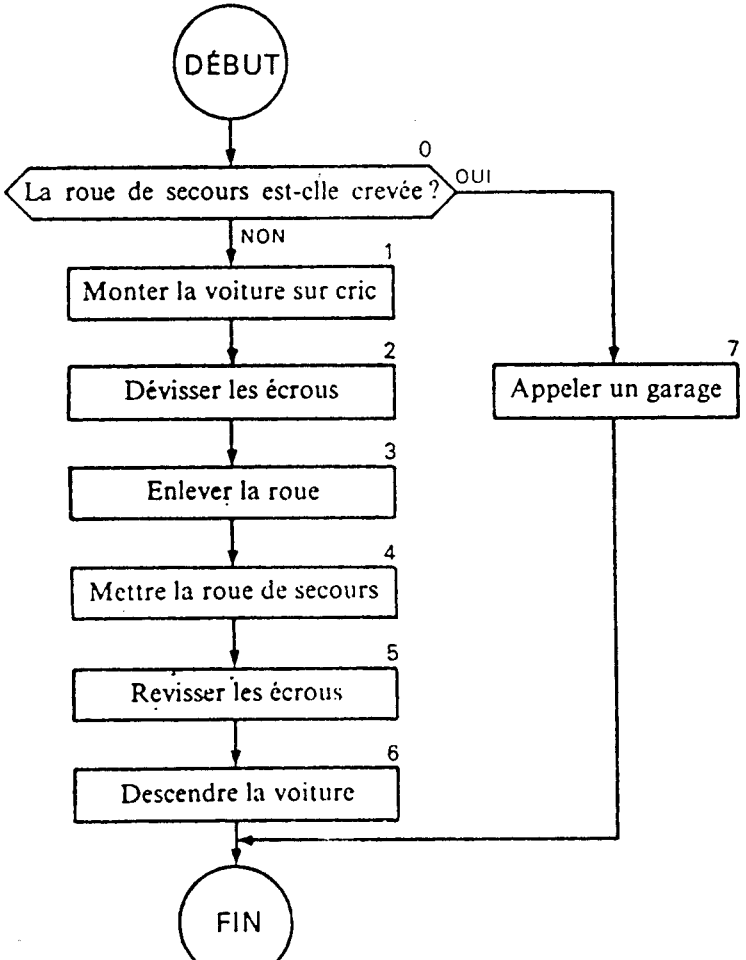


FIG. 1.2. — Second organigramme de la crevaison.

Après avoir suivi la formation de l'organigramme de la crevaison, essayez vous-mêmes d'en inventer un. Dans l'algorithme de l'exercice suivant, vous trouverez probablement des tests et des boucles. Il y a différentes manières de dessiner l'organigramme de cet algorithme, aussi, il est probable que vous en proposerez différentes représentations.

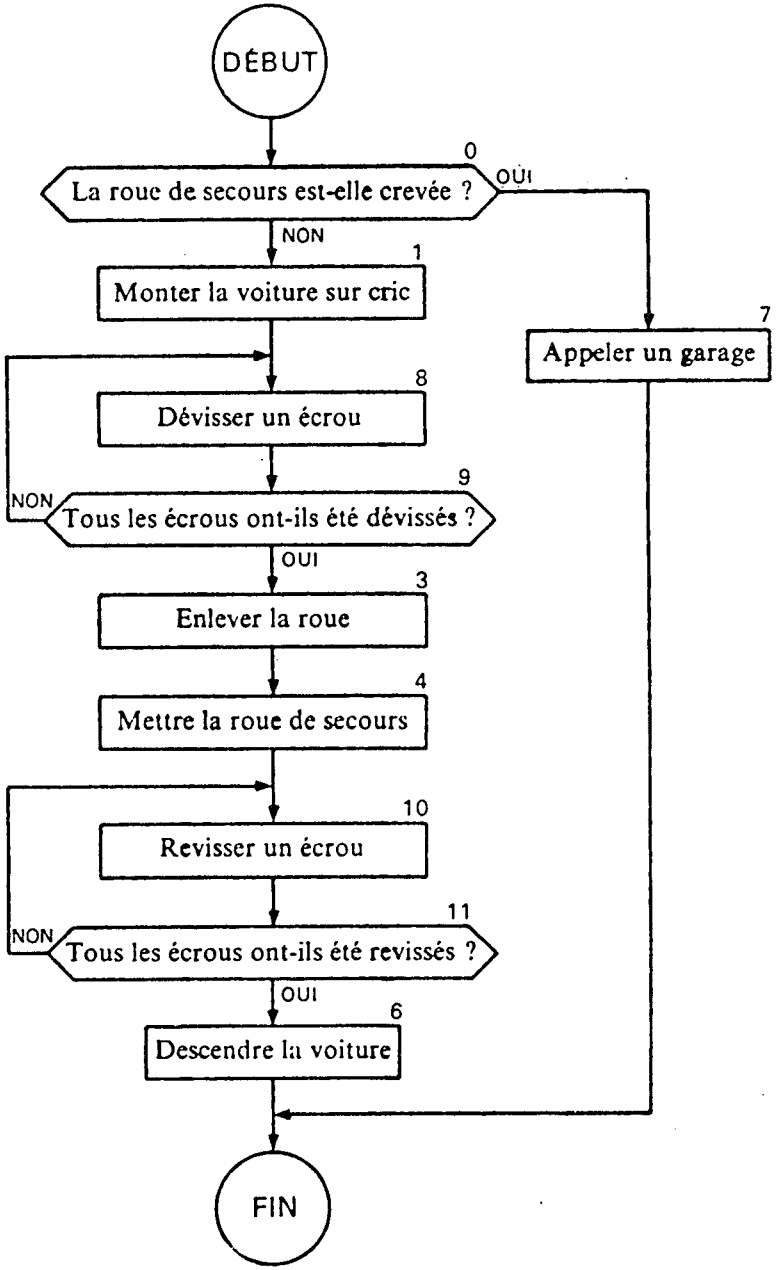
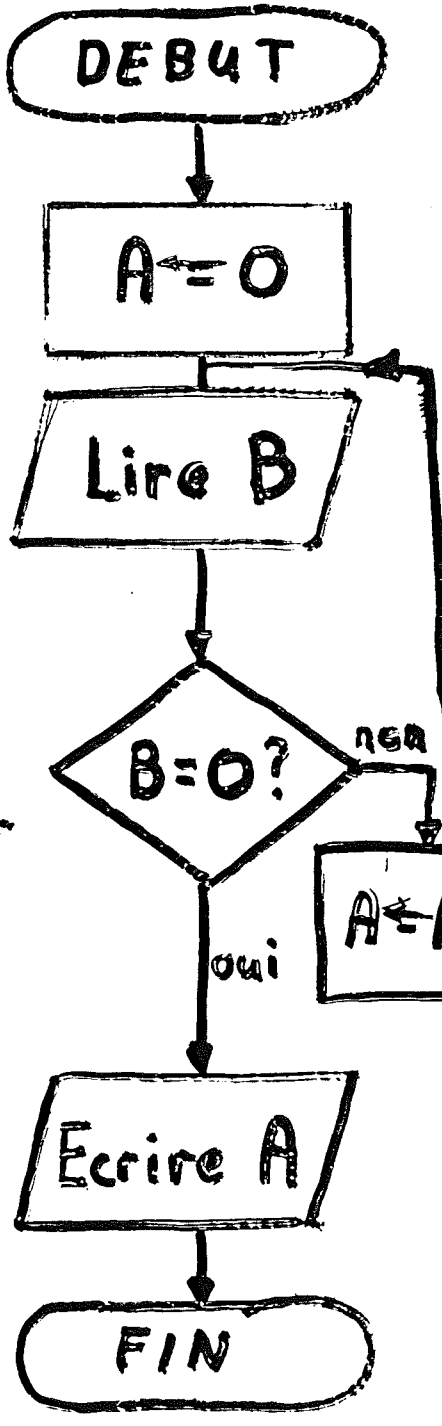


FIG. 1.3. — Organigramme final de la crevaison.

Somme de plusieurs nombres

Enoncé :

Lire des nombres et les additionner jusqu'à ce que le dernier nombre introduit vale zéro. Imprimer alors la valeur de la somme et s'arrêter.

Algorithme	Organigramme	Programme
Mettre le total à zéro.		100 A = 0
Lire un nombre.		110 INPUT B
Le nombre lu vaut-il zéro? Si non, additionner le nombre au total et recommencer.		120 IF B = 0 GOTO 150
Si oui, imprimer le total.		130 A = A + B 140 GOTO 110 150 PRINT A
Arrêter le programme.		160 END

Programme

```
100  A = 0
110  INPUT B
120  IF B = 0 GOTO 150
130  A = A + B
140  GOTO 110
150  PRINT A
160  END
```

Exercice 1.1

Déterminer un organigramme représentant la recette suivante :

« La route pavée » de Tante Lucie

Ingrédients :

1 tasse de noix broyées,
125 g de chocolat à croquer ordinaire,
250 g de pâte de guimauves coupées en deux,
3 tasses de sucre,
$\frac{1}{2}$ tasse de lait bouilli,
$\frac{1}{2}$ tasse de sirop de blé,
1 cuillerée de vanille,
125 g de beurre,
$\frac{1}{2}$ cuillerée de sel.

Mettre le lait, le sirop de blé, le sucre, le chocolat et le sel dans une casserole assez grande et chauffer à grand feu, en remuant sans arrêt jusqu'à ébullition. Réduire le feu et, en entretenant l'ébullition, remuer jusqu'à ce qu'une goutte de cette mixture jetée dans un verre d'eau froide forme une bille molle. Retirer de la flamme et laisser refroidir 10 minutes. Battre le beurre et la vanille jusqu'à un mélange homogène. Ajouter les noix. Répartir les guimauves coupées en deux dans un plat contenant le beurre cuit. Verser le sirop sur les guimauves. Laisser refroidir 10 minutes. Couper en carrés et servir.

1.2. — UN ALGORITHME NUMÉRIQUE

Nous sommes maintenant prêts à étudier un algorithme tiré d'un calcul mathématique. Comme premier exemple, considérons le problème d'une série de termes faisant partie de la suite de Fibonacci :

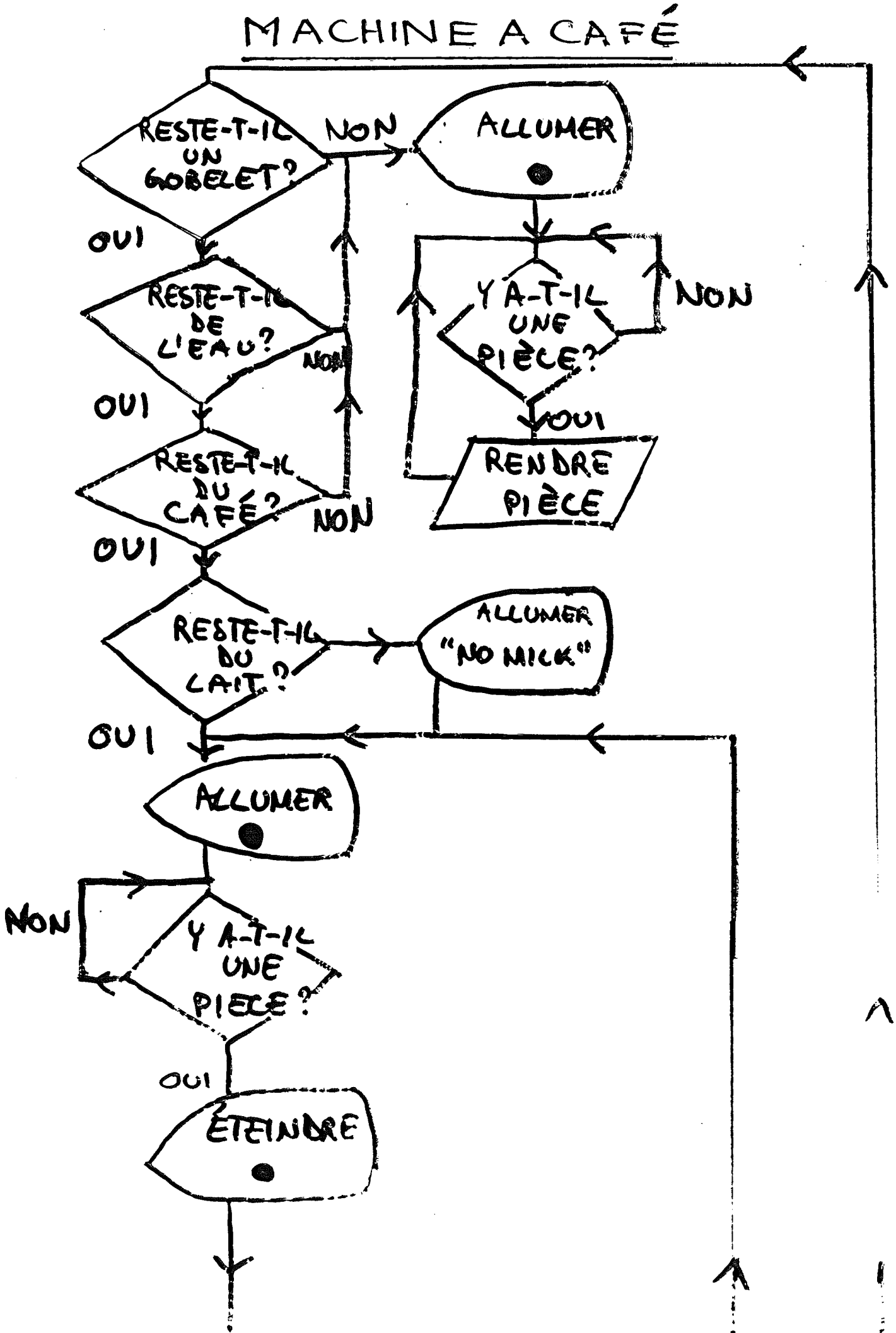
$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$$

Dans cette suite (ou liste de nombres), les deux premiers termes sont 0 et 1. Ensuite, les termes sont générés suivant la règle suivante : chaque nombre de cette suite est la somme des deux précédents. Vérifier donc que ceci est vrai. Ainsi, le terme suivant le dernier de la liste ci-dessus est

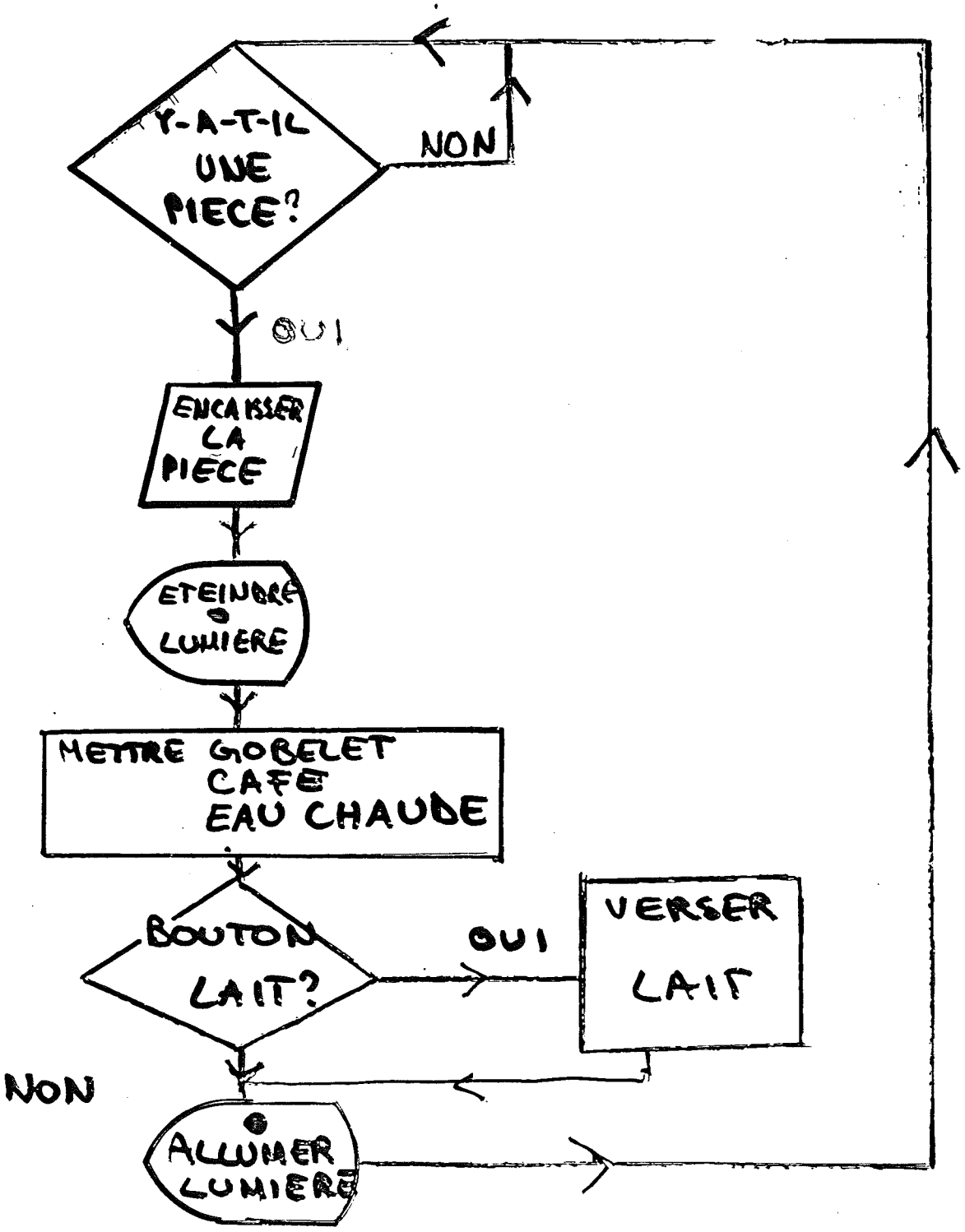
$$34 + 55 = 89$$

Evidemment, nous pouvons continuer la génération des termes de la suite, l'un après l'autre, aussi longtemps qu'on le désire. Mais, si on désire écrire un algorithme sur ce processus (tel qu'un ordinateur puisse l'exécuter par exemple), nous devons être plus précis dans nos hypothèses. Examinons de plus près ce processus.

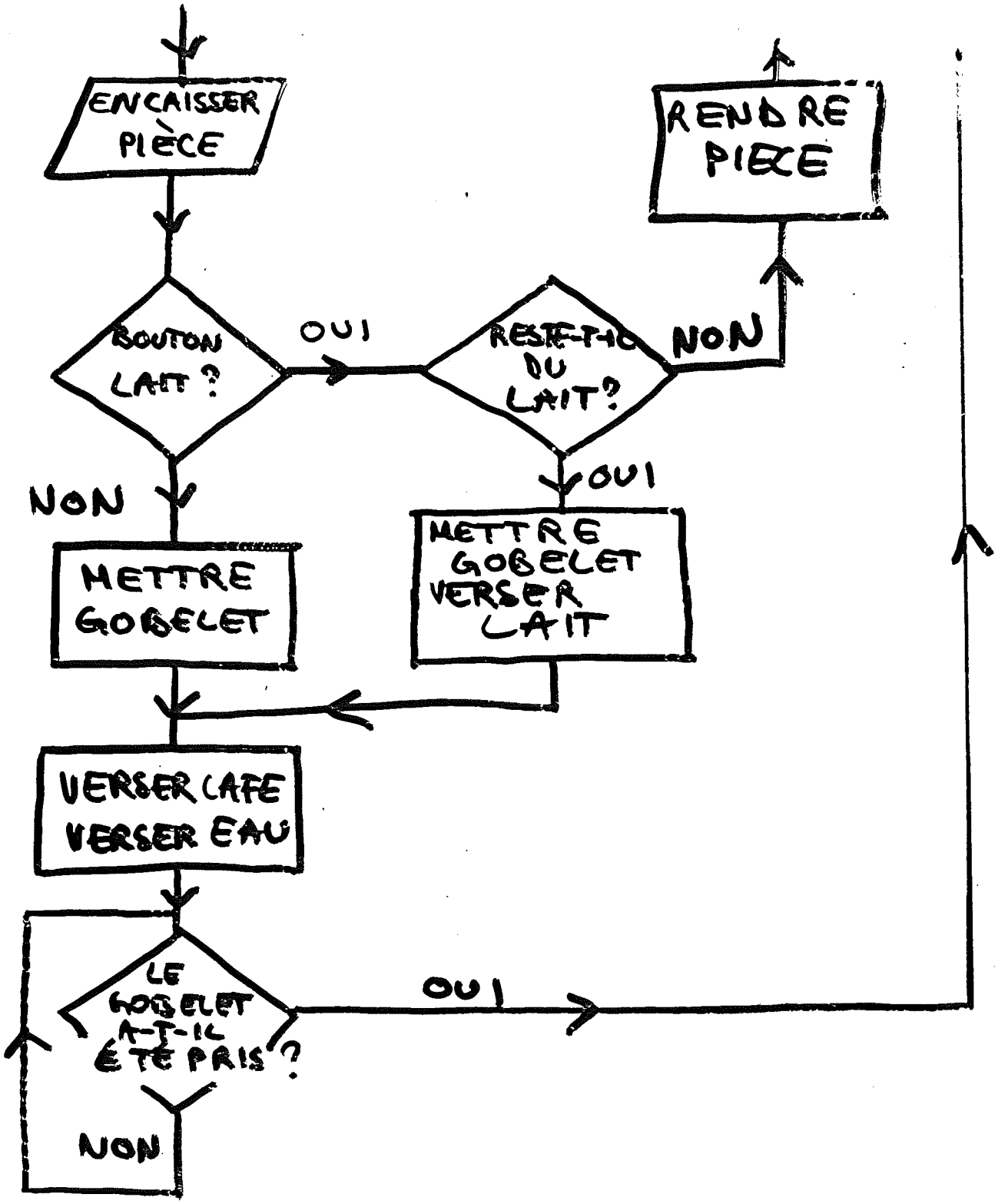
MACHINE A CAFÉ



MACHINE À CAFÉ



MODELE A.


ENCAISSER PIÈCE
RENDRE PIECE
BOUTON LAIT ?
OUI
RESTE-T-IL DU LAIT?
NON
NON
OUI
METTRE GOBELET
METTRE GOBELET VERSER LAIT
VERSER CAFE VERSER EAU
LE GOBELET A-T-IL ÉTÉ PRIS ?
OUI
NON

MODÈLE B