

UNIVERSITE D ALGER
INSTITUT DE LINGUISTIQUE ET DE PHONETIQUE

TRAITEMENT DE TEXTES ARABES ANCIENS

– LE CORAN –

ESSAI DE LEXICOLOGIE AUTOMATISEE
REALISATION DES OUTILS SOFTWARES

MEMOIRE
DE MAGISTER EN SCIENCES DU LANGAGE

OPTION : TRAITEMENT FORMEL DU LANGAGE
ET INFORMATIQUE LINGUISTIQUE

Presenté par : Mr LAMINE ABDAT

Devant le Jury :

Mr A. HADJ SALAH

President

Mr R. BUSA

Rapporteur

Mr A. REY

Examineur

Mr M. HARKAT

Examineur

NOVEMBRE 1985



جامعة بوبكر بلقايد * تلمسان *
كلية الآداب و اللغات
مكتبة اللغات الأجنبية

Inscrit sous le N° 00238
Date: 06.02/2002

Je dédie ce travail tout particulièrement à la mémoire
de mon père, pour le sacrifice de sa vie au service de Dieu.

A ma défunte soeur qui fût toujours à mes côtés.

A ma femme et à mes enfants pour m'avoir soutenu
moralelement sans reserves.



J'adresse ma reconnaissance et ma considération à l'appui indéfectible et au précieux concours de Monsieur le professeur A. HADJ SALAH, Directeur de l'Institut de Linguistique et de phonétique d'Alger.

A Monsieur le professeur R. BUSA pour sa contribution très distinguée et son dévouement indiscutable à la réussite de mon séjour en Italie tant au niveau technique qu'administratif.

De même, je ne saurais oublier Monsieur ROEST CROLLIUS, recteur de l'Université Grégorienne de Rome, ainsi que Monsieur BOLOGNISI, Directeur du groupe GIRCS de l'Université Catholique de Milan, pour leur aide très importante mise à mes services.

Mes remerciements s'adressent également aux Docteurs LUCIA VIAN et PAOLO BRANCA.

Par ailleurs, je tiens à exprimer toute ma gratitude au groupe de recherches de l'ILP et notamment Monsieur Tahar MILLA, pour toute son aide et son assistance dans l'accomplissement de mes travaux.

Je remercie vivement Messieurs TIRECHE et SAADOUNE, pour tous les moyens qu'ils ont bien voulu mettre à ma disposition pour la réalisation de cette thèse.

Enfin, que tout le monde sache que mon travail, est le résultat et l'aboutissement de tous les encouragements et du concours qui m'ont été apportés.

III-3/ Analyse

III-3-1/ Construction des records élémentaires.	33
III-3-1-1/ Principe du programme d'analyse	34
III-3-1-2/ Contrôle de l'analyse	36
III-3-1-3/ Description de l'output de la phase d'analyse	36

III-3-2/ Tri alphabétique

III-3-2-1/ Principe du tri	40
III-3-2-2/ Format des records input tri.	41
III-3-2-3/ Enregistrement et restitution d'un texte.	44
III-3-2-4/ Contrôle du tri	45

III-3-3/ Récapitulatif des formes 49

III-3-3-1/ Schèmes de la récapitulation	50
III-3-3-2/ Le fichier des formes graphiques.	51
III-3-3-3/ Le fichier index-contraire.	54
III-3-3-4/ Récapitulatif des formes.	55

III-4/ Lemmatisation

III-4-1/ Présentation	56
III-4-2/ Les schèmes générateurs du Ism et du verbe	59

III-4-3/ Ebauche d'un menu pour faire la lemmatisation	
III-4-3-1/ Assignation de lemmes aux formes.	66
III-4-3-2/ Codes morphologiques.	67
III-4-3-3/ Formes qui appartiennent à plusieurs lemmes.	69
III-4-4-/ Traitement informatique de la lemmatisation	72
III-5/ Récapitulatif.	79
IV- Ebauche d'un traitement de texte en temps réel.	84
V- Conclusion.	89

A- ANNEXE

1/- Programme d'Analyse - ANAL	92
2/- Programme Mélange - MEL	93
3/- Programme de préparation des listes pour la lemmatisation.	94
4/- Programme d'introduction des lemmes en temps reel.	95
5/- Programme de génération des listes.	96
6/- Bibliographie	97

LISTES

- 1/- Liste des LEMMES par ordre alphabétique
- 2/- Liste des FORMES par ordre alphabétique
- 3/- Liste des FORMES par ordre de fréquence décroissante
- 4/- Liste des STATISTIQUES

SYSTEME DE TRANSCRIPTION

d. ض
 t. ط
 d. ظ
 ع
 g. غ
 f. ف
 q. ق
 k. ك
 l. ل
 m. م
 n. ن
 h. ه
 w. و
 y. ي

ب
 ت
 ث
 ج
 د
 ذ
 ر
 ز
 س
 ش
 ص
 ض

Voyelles : = : a -- : u | : i
 Voyelles longues : تا : a و : u ي : i
 Articles : al

I/ PRESENTATION

Depuis plusieurs siècles, les philologues et autres chercheurs ont traité manuellement les textes anciens, index lexicographiques, dictionnaires et autres, ce qui demandait un temps de travail énorme ainsi qu'un nombre important de personnes sans oublier les risques d'erreurs dus à la manipulation de très grandes quantités d'informations.

Nous citerons à titre d'exemple, le " Dictionnaire Indexé des mots du CORAN " du Docteur Fuad 'Abd - al - Bāqī, qui a été élaboré manuellement.

Sa rédaction a nécessité à elle seule près de sept années de travail (du 7 aout 1938 au 15 fevrier 1945), mises à part les années de recherche.

C'est seulement après la seconde guerre mondiale que l'ordinateur commence à être employé dans le domaine de la linguistique. C'est le début de la " Linguistique Computationnelle " .

- Les essais de traduction automatique de A - J - BOOTH sont de 1946 (1) .
- En 1949, le professeur R - BUSA de l'Université de MILAN, utilisa un ordinateur IBM pour la réalisation du fameux " Index - Thomisticus " de Saint Thomas d'Aquin en 56 volumes (plus de 100 millions d'occurrences).
- Dès 1958, à Besançon, M. QUEMADA commençait à recenser le vocabulaire français sur l'ordinateur Gamma 60 de H.BULL.

1) Cf .La pratique des ordinateurs dans la critique des textes
(no 579 p : 286)

L'utilisation de l'ordinateur s'est généralisée dans plusieurs centres de linguistique tels que :

- Le Centre pour l'Automatisation de l'Analyse Linguistique (CAAL - MILAN).
- Le Centre de Traitement Electronique des Documents (CETEDOC - 1968) de l'Université Catholique de Louvain (1).
- L'Institut de Linguistique et de Phonétique (ILP) d'Alger, dirigé par le professeur A - Hadj - Salah.
- Le Laboratoire de Linguistique Computationnelle (CNR) du professeur A - Zampolli (Pise - Italie).
- Le Centre de Recherche en Linguistique du CNRS (Nancy 1960) équipé de machines électroniques pour la réalisation du " Trésor de la Langue Française ".

Afin que l'ordinateur puisse être utilisé par des non spécialistes, des langages informatiques ont été développés tels que :

- LEXICO : qui est un langage informationnel destiné à aider les lexicographes à emmagasiner, à éditer, à faire concorder, à subordonner et à standardiser l'information pour les dictionnaires et autres produits de l'analyse lexicale (2).

Lexico a été écrit en langage Fortran (18.000 lignes codes) sur un Univac bi-processeur (1110).
La version 2 de Lexico a été terminée en juillet 1976 après cinq années de travail.

1) Cf. La pratique des ordinateurs dans la critique des textes
(no 57) pp 163 - 168)

2) Cf. Qemada, Cahiers de lexicologie (no 30 / 1977)

- PASCAL : Langage de haut niveau, orienté "Traitement de textes".
Son utilisation est particulièrement simple, et facile à apprendre.
Il a été développé par Niklaus Wirth, de l'école polytechnique fédérale de Zurich en 1971.

- PROLOG : Langage de la "cinquième génération".
C'est un langage simple et puissant, il permet la manipulation des informations sous forme symbolique, utilise des mécanismes de "raisonnement" analogues à l'intelligence artificielle, et favorise l'interactivité.
Il existe actuellement plusieurs interpréteurs de ce langage (Prolog II, Prologue / P, Foll : First Order Logic Language).
Prolog est né en 1970 dans les laboratoires du groupe d'Intelligence Artificielle - GIA - de l'Université de Marseille.

Les travaux de lexicologie et lexicographie, avec l'aide d'un ordinateur, sont réalisés depuis plus de trente ans déjà.
On peut citer par exemple :

- Le Dictionnaire Machine de l'Italien (DMI) qui a été réalisé par un groupe de linguistes et d'informaticiens de la division linguistique du CNUCE sous la direction du professeur A - Zampolli.
Ce dictionnaire fait l'inventaire des formes de l'italien, auxquelles toute information phonétique, morphologique, lexicale, syntaxique et sémantique est associée. Ce qui permet de faire la lemmatisation automatique (1).

- Le Système de traitement automatique de l'anglais médical développé aux " National Institutes of Health" (2) est opérationnel depuis plusieurs années.
Ce système automatisé d'analyse sémantique de données en langue naturelle comprend l'acquisition d'informations textuelles, un mécanisme d'analyse grammaticale, la construction et l'interrogation du dictionnaire.
Un système analogue a été établi pour le français médical.

1) Cf. A Zampolli, The Italian Dictionary projet.

2) Cf. Qemada, Cahiers de lexicologie, (no 30/1977 pp, 47 - 74)

- Le Trésor de la langue Française (dictionnaire de la langue du 19e et du 20e siècles), a été réalisé par 150 chercheurs et techniciens en moins de sept ans et ceci à l'aide de l'ordinateur.

Aujourd'hui, grâce à l'existence de terminaux en langue arabe, il n'est plus nécessaire de recourir à la transcription latine pour faire du traitement automatisé des textes arabes.

Avec l'aide du professeur A- Hadj-Salah (Directeur de l'institut de linguistique et de phonétique d'Alger) et du professeur R-BUSA (Directeur du centre de Gallarate MILAN), nous nous proposons d'appliquer la linguistique computationnelle aux textes arabes anciens, en prenant pour base la théorie du Professeur Hadj-Salah (théorie néo - Khalilienne).

Le but de ce travail consiste, d'une part à écrire tous les outils software pour permettre le traitement des textes Arabes, et d'autre part à appliquer ces outils afin d'obtenir un "Dictionnaire machine du texte arabe " sur ordinateur .

Dans notre cas, le texte choisi est le livre sacré de l'Islam (LE CORAN) qui est une source inépuisable de recherches tant sur le plan linguistique, que sur le plan scientifique.

La réalisation du dictionnaire machine du CORAN sur ordinateur permettra aux philologues d'effectuer leurs recherches avec rapidité et fiabilité.

Ce travail serait incomplet si nous ne parlions pas du traitement de textes en temps réel: le linguiste n'aura plus à faire ses recherches sur des tonnes de papier, mais entamera un dialogue avec l'ordinateur (recherche automatique du contexte d'un mot par exemple).

Notre étude sera structurée de la façon suivante :

- 1/- Présentation linguistique du texte coranique
- 2/- Outils informatiques pour le traitement de textes
- 3/- Traitement du texte coranique (Résultats).

Pour éviter toute ambiguïté, les logiciels de traitement de textes que l'on va écrire seront différents de ceux fournis par un constructeur d'ordinateurs. Ces logiciels permettent à l'utilisateur d'enregistrer un texte, de le relire et de le corriger avant impression. Par contre, les notres vont à partir d'un texte déjà enregistré et corrigé, générer un lexique complet de ce texte qui pourra être consulté par un philologue dans le cadre de ses recherches de façon automatique et rapide.

II/ PRESENTATION LINGUISTIQUE DU TEXTE CORANIQUE

II-1/ Le texte coranique

La révélation du CORAN (Al - Qur'ān) au Prophète de l'Islam Muhamed s'est faite d'une manière fragmentée entre 612 (JC) environ et 632 (JC), d'abord à la Mekke puis à Medine.

Dès qu'une révélation était transmise au Prophète, ses compagnons la notaient soigneusement.

Parmi ses nombreux secrétaires, on citera notamment (1) :

- Mu'ād ibn Jabāl
- 'Ubay ibn Ka'b
- Zaid ibn Iābit

Après la mort du Prophète (632/16H), le premier calife de l'Islam Abu Bakr, sur l'initiative de 'Umar, fit procéder à la constitution du corpus coranique qui devait grouper l'ensemble des recueils individuels.

A partir de là fût effectuée la première recension du Coran.

C'est d'abord au Médinois Zayd ibn Iābit qu'incomba la charge de cette grande et délicate mission.

Le premier recueil fût conservé par le calife Abu Bakr, puis par son successeur 'Umar, qui le confia à sa mort à sa fille Hafsa.

Vingt ans plus tard, le 3ème calife 'Utmān (644 - 656 JC), ayant été frappé par les divergences articulatoires et morphologiques d'origine régionale ou tribale, décida de rédiger une nouvelle recension du corpus coranique à partir de celui d'Abu Bakr.

1) Cf. R. Blachère, Introduction au Coran, p.110
et
S. Kechrid, Al Qur'an al karim, p. XIX

La commission de rédaction était composée de (1) :

- Zayd ibn Iābit
- ʿAbd - Allah ibn Az-zubayr
- Saʿid ibn al-ʿĀs
- ʿAbd - al - Raḥmān ibn al Hārit

Ce deuxième recueil est depuis, la seule recension admise par les Musulmans (Muṣḥaf ʿUtmān = la Vulgate de Othman).

1) Cf. R. Blachère, Introduction au Coran , p. 53

II-2/ Forme linguistique du texte Coranique

Malgré cette deuxième réforme, on ne pouvait lire correctement le Coran qu'à la condition expresse de le posséder par coeur, car il était transcrit en une écriture consonantique, sans indication de voyelles ni de points diacritiques.

Ce n'est que sous le règne du calife 'Umayyad 'Abd - al - Malik ibn Maruān (685 - 705) que l'on plaça les signes diacritiques, toujours en se basant sur la tradition orale la plus accréditée.

A la fin du 3ème siècle de l'Hégire, un spécialiste de la reproduction orale coranique, Ibn Muḡāhid, réussit à recenser les types de reproductions les plus répandus à son époque. Ce sont :

- Médine : Nafi' ibn Na'im (mort en 169H / 785)
- Mekke : 'Abdu Allah ibn Kaṭir (mort en 120H / 737)
- Bassora : Abu 'Amr ibn al 'Alā (mort en 70H / 689)
- Damas : 'Abdu Allah ibn 'Amr (mort en 118h / 736)
- Coufa : 'Acim ibn Abi-n-Najud (mort en 127H / 744)
- Hamza ibn Habib (mort en 156H / 772)
- Al - Kisai ('Ali ibn Hamza) (mort en 189H / 804)

Ces sept systèmes de reproduction furent désormais considérés comme canoniques.

Des améliorations graphiques ont été apportées (on citera celui d'AL-Ḥalil Ibn Ahmad, le grand linguiste arabe et maître du célèbre Sibawayh, qui mit au point le système graphique des voyelles et autres signes diacritiques), et ce afin que le texte coranique soit doté d'un diacritisme complet qui évite les confusions de lettres de même forme.

1) Cf. R. Blachère Introduction au Coran, p. 53

II-3/ Les divisions du texte coranique

Le texte coranique comporte 114 chapitres ou Sourates (Sūrat) de longueur différente.

La plus longue comporte 286 versets (Sourate II El-Baqara), et la plus courte comporte 3 versets (Sourate 108 EL-Kawtar).

Dans chaque sourate, nous retrouvons :

- Un titre
- Des indications générales sur le lieu de la révélation (Medine ou la Mekke).
- Le nombre de versets qui la composent.

Chaque sourate est divisée à son tour en versets (Ayāt). La longueur des versets est variable (1 page à 2 mots), ainsi que leur nombre (de 3 à 286).

Certaines sourates débutent par des sigles parfois composés de 2 à 3 lettres et qui sont nommés (1) Al - Fawatih (les liminaires) ou Al - Hurūf al Muqatta'a (t) (lettres séparées).

La classification des sourates se présente selon un ordre de longueur décroissante. Elle ne suit pas l'ordre chronologique de la révélation des versets.

La classification des chapitres (Sourates) ainsi que la numérotation des versets sont l'oeuvre du Prophète lui-même inspiré par Celui qui l'a envoyé comme "grâce et miséricorde à toute l'humanité".

L'ordre des versets suit les faits et les événements de la communauté Musulmane.

Le texte a été également partagé en 60 " Hizb " (fragments).

1) Cf. R. Blachère, Introduction au Coran, p.130

III/ OUTILS INFORMATIQUES

III-1/ Introduction

Avec l'avènement de l'ordinateur, et profitant des différentes possibilités méthodologiques offertes par l'informatique, les recensements intégraux d'énormes quantités d'informations peuvent être faits aisément, et ce avec exactitude, cohérence, maniabilité et rapidité.

La manipulation de ces énormes quantités d'informations s'expliquent par le fait que le système d'échantillonnage ne peut être appliqué chaque fois que la sémantique est en question.

Ces premiers outils informatiques nous mèneront à avoir un texte "Machine Readable Form" (M.R.F) qui sera compréhensible par l'ordinateur.

Le professeur BUSA soutient que : " Lors du transfert sur bande magnétique des lettres et ponctuations, le texte n'est pas encore en M.R.F. "

A son avis, un texte n'est en Machine Readable form, que lorsqu'on a produit :

- le texte analysé par mots individuels lemmatisés
- le fichier des formes lemmatisées avec le résumé de leurs données quantitatives dans le texte.
- un autre fichier, celui des lemmes avec leurs codes (typologiques, etc...) et les résumés des données quantitatives de toutes les formes de chacun.

Un pareil texte en M.R.F doit être considéré comme point de départ pour n'importe quelle recherche philologique. Ce n'est pas l'étape finale, mais le début de la recherche.
Afin d'obtenir un texte en M.R.F on doit procéder à :

- 1/ L'enregistrement
- 2/ L'analyse
- 3/ La lemmatisation

III-2/ Enregistrement

Ce chapitre se compose de trois parties:

- Préparation (Preediting)
- Enregistrement
- Correction

III-2-1/ Préparation

III-2-1-1/ Choix des parties et des données à enregistrer.

Une fois le choix du texte fait, il faut calculer approximativement le nombre total des mots (pour avoir une idée de la grandeur du texte). Pour une recherche lexicologique minimale, il est nécessaire d'avoir entre 30000 et 100000 mots.

Cette préparation consiste à préciser dans le texte :

- les parties à enregistrer et celles qui ne le sont pas (en-tête, préface, introduction, index, notes, titre courant, etc...)

- les deux types fondamentaux des données à enregistrer (type référence/ type texte). La référence est conventionnelle; en général il en existe deux types:

- référence naturelle ou logique, qui codifie la division existante dans le texte (partie, tome, chapitre, etc....)
- référence physique typographique qui codifie volume, page, ligne dans la page.

Il est à noter que lors de la saisie du texte sur machine, chaque code spécifiant le type de données (référence ou texte) doit être inscrit en début de ligne .

III-2-1-2/ Description du système graphémique du texte

Il faut inventorier les graphèmes qui constituent les "mots" et les "ponctuations".

Graphèmes mots

les différents caractères qui peuvent constituer le mot sont les lettres, chiffres, pro-lettres, et les en-lettres.

Pro-lettres: sont des caractères qui occupent un espace propre et qui sont différents des lettres (tels que trait d'union et apostrophe pour les mots composés).

En-lettres : sont des caractères qui se trouvent dans le même espace que la lettre (tels que accent ou voyelle en arabe).

On pourrait également inclure parmi les en-lettres des qualités typographiques telles que : caractères gras, italiques, soulignés, éparpillés, majuscules, etc....

Graphèmes ponctuations

Il faudrait également inventorier le type de ponctuation et leur groupement pour connaître le nombre maximum de ponctuations qui se suivent.

Parmi les ponctuations, l'informaticien inclut toujours l'espace (blanc).

Graphèmes équivoques

Parfois, il est difficile de différencier une pro-lettre d'une ponctuation telle que:

- trait d'union "hyphen" qui est une pro-lettre
- et
- tiret "dash" qui est une ponctuation.

Graphèmes sous-entendus

Il existe parfois dans le texte des points de ponctuation sous-entendus, à la fin d'un titre par exemple .
Après " et cetera" (etc) le point sera considéré comme point d'abréviation et comme point de ponctuation en même temps.

Pour la recherche automatique des limites des phrases signalées par les points, il est préférable d'enregistrer au moins les points sous-entendus.

Autres graphèmes

Il existe souvent dans un texte des informations ou signes supplémentaires pour indiquer par exemple:

-Citations

-Dialogues (pour signaler les interlocuteurs dans les pièces de théâtre par exemple)

- Signes typologiques dans les éditions critiques (mots illisibles d'un manuscrit, ou partiellement recopiés)

Il faudrait alors décider de les coder ou non.

Groupement des graphèmes

A partir de cet inventaire des différents graphèmes, il faut préciser le système graphémique de l'enregistrement.
A chaque graphème, il faut lui assigner un caractère informatique (code) et sa touche correspondante.
Les mono-codes sont évidemment à préférer aux bi-codes.

Les graphèmes seront groupés en tableaux:

- graphèmes mots:

- . chiffres
- . lettres
- . pro-lettres
- . en-lettres.

- graphèmes "extérieurs" au mot :

- . ponctuations
- . codes typologiques
- . commandes
- . etc....

Il faudrait également systématiser la présence des espaces (un espace après chaque mot, chaque ponctuation, chaque code)

III-2-1-3/ Système graphémique du texte Coranique

Les éditeurs, pour conserver l'orthographe coranique (celle du Coran de 'Uthman), ont dû rajouter des traits spécifiques (ou signes) de lecture.

On y trouve (1) :

signe point النقطة المستديرة

Indique que le mot commence par un 'Alif non prononcé.
La position du point par rapport au 'alif nous indique la voyelle qui suit:

Si le point se trouve sur le 'Alif, alors la voyelle qui suit est un "a".

تَتَوْنَ الَّذِي

Si le point se trouve au dessous du 'Alif, alors la voyelle qui suit est un "i".

تَسْتَعِينِي ! هِدَانَا

Si le point se trouve au milieu du 'Alif, alors la voyelle qui suit est un "u".

أَنْ أَشْكُرُ

Lors de l'enregistrement du Coran, le signe " point " et le " 'Alif " ont été remplacés par le 'Alif.

ا ← { ا ا ا

1) Cf. M Said Cherifi, Le texte coranique, annexe A

Signe "Zéro-rond" الصفير المستدير

قَالُوا	قَالُوا
أَوْ تَلِيكَ	أَوْ تَلِيكَ
نَبِيَّيْ	نَبِيَّيْ

Il se trouve toujours sur les voyelles longues (و ا ي) et indique que le caractère correspondant ne doit pas être "prononcé" (caractère en plus)

Ce signe est supprimé lors de l'enregistrement .

Signe "Zéro oval" الصفير المستطير

Se trouve sur le 'Alif , et indique qu'une liaison est à faire lors de la prononciation.

أَنَا خَيْرٌ

Comme le signe précédent, il a été supprimé

Sukun sur le "nun" avec "šada" sur le caractère "و" et "ي"

Indique une gémiation du "nun"

مَنْ يَقُولُ ← مَنْ يَهْوُلُ

Comme la "šada" dans ce cas précis, est utilisée uniquement en lecture, elle a été supprimée lors de l'enregistrement.

Sukun sur le "ط" avec "Šada" sur le caractère "ت"

Pour indiquer qu'il y a gémiation partielle du "ط" sur "ت"

أَطَّطُ ← أَطَّطُ

La "Šada" sur le "ت" a été également supprimée lors de l'enregistrement.

Petites lettres se trouvant sur les caractères

Elles indiquent qu'elles n'ont pas été inscrites dans le texte coranique de 'Utmān. Il a fallu lors de l'enregistrement, écrire ces petites lettres dans leurs positions correspondantes.

exemple :

كُتِبَتْ ← كَتَبَتْ

تَلُّوْا ← تَلُّوْا

'Alif sur le "و" et le "ي"

Indique qu'à l'origine, le mot était écrit avec le "و" ou le "ي" (conservation de l'ancienne orthographe); actuellement ces caractères (وي) ont été remplacés par le 'alif.

الصَّوْرَةُ ← الصَّوْرَةُ

التَّوْرِيَّةُ ← التَّوْرِيَّةُ

L'orthographe de la Hamza (4)

Signe notant l'attaque ou la détente vocalique (').
Dans le texte coranique, la hamza a été inscrite sur la ligne même et sans support.
Par contre dans les textes modernes, ce signe en initiale ou à l'intérieur d'un mot a une voyelle a, u, i comme support.

Signe " ~ "

Indique un allongement de la voyelle, il est utilisé uniquement en lecture.

Signe " — " qui accompagne le 'alif

Sa position par rapport au 'alif détermine la voyelle qui le suit.

اوتِي - اوتِي

امِي - امِي

اِذَا - اِذَا

Signe



Indique la fin d'un verset et donne son numéro dans la sourate

Signe



Indique le début et la fin d'une marque de hizb soit (1/8, 1/4, 1/2, 1)

III-2-1-4 Préparation du texte Coranique

Les seuls signes du texte coranique qui ont été gardés et que nous avons appelés "ponctuation" sont:

- marque Hizb : définie par le caractère #
- marque Soujoud: définie par le caractère *

Les graphèmes mots sont constitués par:

- 36 lettres de l'alphabet (consonnes)

آ ا ا و ئ ء ا ب ت
ث ج ح خ د ذ ر ز س
ش ص ض ط ظ ع غ
ف ق ك ل م ن ه و ي لا

- 10 chiffres (0 à 9)

0 1 2 3 4 5 6 7 8 9
٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

- 13 voyelles

ـَ ـِ ـُ ـِـ ـِـ ـِـ
ـِـ ـِـ ـِـ ـِـ ـِـ

(voir annexe pour le codage informatique).

III-2-2/ Enregistrement

Le traitement du texte doit se faire sur un gros ordinateur du fait de la quantité importante d'informations. Il est donc nécessaire que le matériel de saisie utilisé soit compatible avec l'ordinateur pour le traitement.

la mémorisation (ou l'enregistrement) du texte se fait dans la plupart des cas, soit sur cartouche (capacité en M octets), soit sur disquette 8 pouces ou 5 pouces (capacité de 250 k octets à 1000 k octets)

Pour de gros travaux, il est possible aujourd'hui de recourir à l'enregistrement automatique par lecteur optique (à l'Université d'Oxford, Mme Susan HOCKEY utilise un "kurzweil" qui est un type de périphérique optique). Pour l'arabe, aucune tentative n'a été faite dans ce domaine (1984) du fait de l'inexistence de matériel optique en arabe.

Il est préférable et même nécessaire que le philologue enregistre lui même une partie du texte (1 à 2 volumes min) afin de déterminer avec exactitude les différents graphèmes utilisés.

Actuellement, avec les micro-ordinateurs, la saisie peut se faire aisément et n'importe où, à condition que les disquettes soient utilisables sur un gros ordinateur.

Mais, faire appel également aux micro-ordinateurs pour le traitement du texte reste chose difficile voire impossible du fait de la faible capacité de mémorisation et de la lenteur dans le temps de traitement (Recherche, Tri et Impression).

Malgré ces difficultés, nous avons utilisé un micro-ordinateur VT-104 de la société Vidéoton (Honrie) ayant 4 postes de travail indépendants de 64K octets de mémoire chacun, et de 20M octets de mémoire auxiliaire (2 disques amovibles et 2 disques durs), pour faire l'enregistrement et le traitement de textes arabes. (Matériel acquis récemment par l'institut de linguistique d'Alger).

L'enregistrement devient trop lourd quand le nombre des bi-codes augmente, mais il est toujours possible de transformer ensuite les mono-codes et les bi-codes dans la série maximum des mono-codes EBCDIC (256 codes différents).

Toutes les en-lettres seront enregistrées comme des caractères indépendants après la lettre avec laquelle elles sont combinées dans le texte. Cela est conséquent pour plusieurs raisons:

- Ne pas alourdir l'enregistrement avec autant de codes qu'il n'y a de combinaisons possibles entre lettres et en-lettres.
- Faciliter le repérage et la correction des erreurs de l'enregistrement.
- Pouvoir distinguer dans le tri, caractères du premier niveau et du deuxième niveau (en arabe, le tri s'effectue d'abord sur les consonnes, ensuite sur les voyelles).

Cette distinction des en-lettres sera gardée pendant toute la durée du traitement de texte qui va suivre, mais l'impression finale pour l'édition des listes définitives mettra les en-lettres à leurs places.

b/ Ponctuation

la ponctuation peut se trouver n'importe où dans le texte mais doit être comprise entre deux blancs (espaces).
On trouve :

- marque de Hizb : représentée par le caractère "#"
(1/8, 1/4, 1/2, 1)

- marque de soujoud : représentée par le caractère "*" "

c/ Texte

Le texte ne doit jamais commencer en colonne 1 (ou en position 1 de la ligne d'enregistrement) pour ne pas le confondre avec une référence .

Chaque mot du texte doit être compris entre deux espaces. Les lignes textes sont de longueurs variables et sont enregistrées également sur des records de longueurs variables.

سورة آل عمران من مدينته و آياتها ٢٠٠

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
 ١ اللَّهُ لَا إِلَهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ ٢ نَزَّلَ عَلَيْكَ الْكِتَابَ
 بِالْحَقِّ مُصَدِّقًا لِمَا بَيْنَ يَدَيْهِ وَأَنْزَلَ التَّوْرَةَ وَالْإِنْجِيلَ ٣ مِنْ
 قَبْلُ هُدًى لِلنَّاسِ وَأَنْزَلَ الْفُرْقَانَ إِنَّ الَّذِينَ كَفَرُوا بِآيَاتِ اللَّهِ
 لَهُمْ عَذَابٌ شَدِيدٌ وَاللَّهُ عَزِيزٌ ذُو انْتِقَامٍ ٤ إِنَّ اللَّهَ لَا
 يَخْفَى عَلَيْهِ شَيْءٌ فِي الْأَرْضِ وَلَا فِي السَّمَاءِ ٥ هُوَ الَّذِي
 يُصَوِّرُكُمْ فِي الْأَرْحَامِ كَيْفَ يَشَاءُ لَا إِلَهَ إِلَّا هُوَ الْعَزِيزُ
 الْحَكِيمُ ٦ هُوَ الَّذِي أَنْزَلَ عَلَيْكَ الْكِتَابَ مِنْهُ آيَاتٌ
 مُحْكَمَاتٌ هُنَّ أُمُّ الْكِتَابِ وَأُخَرُ مُتَشَابِهَاتٌ فَأَمَّا الَّذِينَ فِي قُلُوبِهِمْ
 زَيْغٌ فَيَتَّبِعُونَ مَا تَشَابَهَ مِنْهُ ابْتِغَاءَ الْفِتْنَةِ وَابْتِغَاءَ تَأْوِيلِهِ وَمَا يَعْلَمُ
 تَأْوِيلَهُ إِلَّا اللَّهُ وَالرَّاسِخُونَ فِي الْعِلْمِ يَقُولُونَ ءَأَمَّنَّا بِهِ كُلٌّ مِنْ عِنْدِ
 رَبِّنَا وَمَا يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ ٧ رَبَّنَا لَا تُزِغْ قُلُوبَنَا بَعْدَ
 إِذْ هَدَيْتَنَا وَهَبْ لَنَا مِنْ لَدُنْكَ رَحْمَةً إِنَّكَ أَنْتَ الْوَهَّابُ ٨



REFERENCE SOURATE

س 3

سُورَةُ آلِ عِمْرَانَ مَدَنِيَّةٌ
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
اِنَّمَا

1 ا

اللَّهُ لَا إِلَهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ

2 ا

نَزَّلَ عَلَيْكَ فِي الْكِتَابِ الْوَحْيَ مُتَسَدِّدًا لِيَمَّا يَتَذَكَّرُ
وَأَنْزَلَ الْقُرْآنَ آيَةً وَالْآيَاتُ تُحْمَلُ

3 ا

مِنْ عَمَلٍ مُدْرِكٍ لِنِقْمَتِهِ وَأَنْزَلَ الْفُرْقَانَ إِنَّ الْفُرْقَانَ

4 ا

كُنُفَرًا وَمِيقَاتٍ اللَّهُ لَنُحْمَلُهُمْ وَعَذَابُهُمْ شَدِيدٌ وَاللَّهُ عَزِيزٌ
ذُو انبِعَادٍ #

#

إِنَّ اللَّهَ لَا يُحْمَلُ عَلَيْهِمْ فِي الْآدَامِ وَاللَّهُ فِي
السَّمَاءِ

5 ا

فَوَالَّذِينَ يَبْتُغُونَ دُكْمًا فِي الْآدَامِ كَمَثَلِ بَشَارٍ لَا إِلَهَ
إِلَّا هُوَ الْعَزِيزُ الْحَكِيمُ

6 ا

MARQUE DE HIZB

REFERENCE VERSET

ENREGISTREMENT DU TEXTE CORANIQUE

III-2-2-2/ Corrections

Le professeur BUSA après trois corrections à la lecture (de 100% du texte) sur 10,5 millions de mots a encore trouvé 1600 caractères erronés dus à des erreurs d'enregistrement.

En effet, deux lignes entières n'ont pas été enregistrées du fait de "l'homoteleton".

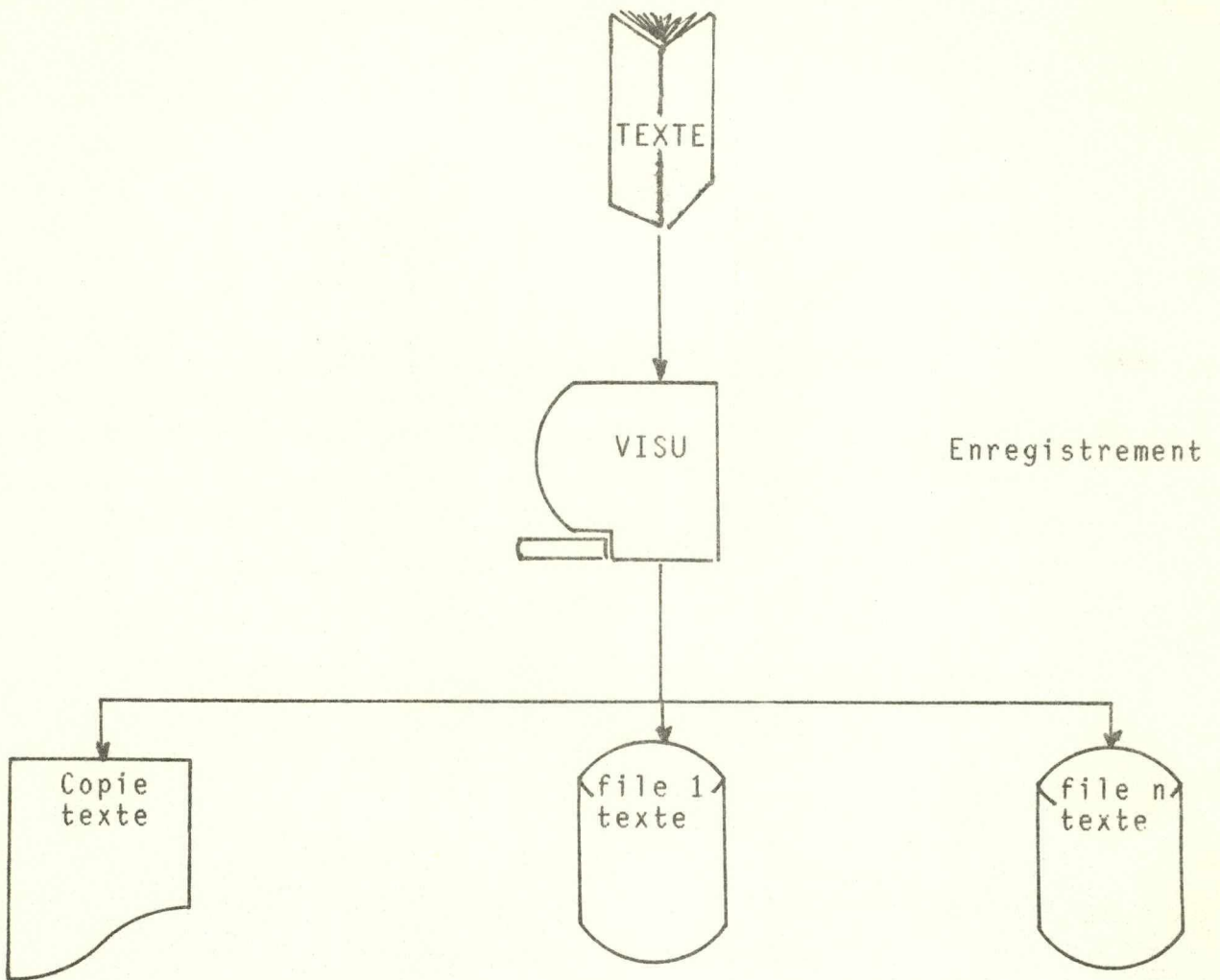
Cette erreur a été répétée trois fois par trois personnes différentes.

C'est pour cela que le nettoyage de l'input doit être fait minutieusement (20% du temps pour l'enregistrement et 80% du temps pour la correction).

Méthodes de correction

Les méthodes de correction d'un texte enregistré dans la mémoire d'un ordinateur peuvent se présenter ainsi :

- Soit faire enregistrer deux fois et par deux personnes différentes le même texte, et comparer les deux textes par programme .
- Soit comparer le texte et la liste de l'enregistrement par la même personne, ou par deux personnes; dans ce cas il faudrait établir des règles de prononciations .



supports externes
de mémorisation
(cartouches de 5M octéts)

Enregistrement / Correction.

III-3/ Analyse

Elle a pour but d'introduire la lemmatisation, et en outre elle permet de découvrir les erreurs de l'enregistrement.

L'analyse comporte 3 phases:

- Construction des records élémentaires
- Tri alphabétique
- Récapitulatif des " formes".

III-3-1/ Construction des records élémentaires

La génération (ou construction, ou dérivation) des records élémentaires divise l'input qui est groupée par lignes écran de 80 caractères en autant de records qu'il y a "d'éléments" dans le texte.

Les éléments du texte sont:

- les mots, ponctuations, codes, et commandes
- records de références

Chaque record doit contenir en sortie

- un élément
- le code du type de l'élément
- une référence.

III-3-1-1/ Principe du programme d'analyse

Le programme commence par explorer la première position de chaque record de l'input pour déterminer le type de record (Référence, mot ou ponctuation)

- Si le premier caractère est un sin " س " ou 'alif " ا " alors tous les caractères se trouvant après le sin et le 'alif et jusqu'au blanc seront transférés en output sans aucun traitement, en lui ajoutant un code pour spécifier que c'est un record référence et un numéro de record.

- Si le premier caractère est un blanc, il faudra alors balayer chaque caractère l'un après l'autre et les comparer avec les tableaux en mémoire afin d'en déterminer le type.

Les chiffres, lettres, pro-lettres, et en-lettres iront composer un record mot précédé du code mot.

S'il s'agit d'une ponctuation ou d'un code typologique ou d'une commande, on aura autant de records en sortie mono ou bi-codes, mais chacun avec le code de son type.

(voir en annexe algorithme et programme)

NB1 - Tous les records mots en output auront la même longueur commandée soit par le mot le plus long ou par la référence la plus longue (en arabe, la longueur maximum est de 26 caractères).
Par contre, les records références et ponctuations auront des longueurs variables.
il est indispensable d'ajouter à chaque record:

- une zone indiquant le nombre de caractères inscrits
- la numérotation ascendante continue du record

NB2 - La prolifération (translation) des références aux records texte sera exécutée de façon complète et définitive seulement dans les opérations finales de la lemmatisation. Mais, parfois, la nature du texte et de son premier traitement demande que l'on prolifère au moins de façon réduite la référence de chaque élément (Sourate, verset, Hizb pour le Coran). Cela nous permettra de retrouver les mots dans le textes imprimé et enregistré pour une éventuelle correction.

III-3-1-2/ Controle de l'analyse

Pour le contrôle de l'exécution de l'analyse, on pourrait demander la liste des records construits.
Pour ne pas avoir une liste trop importante, on devra imprimer en comprimant les records en horizontale, et se rappeler que chaque record possède une zone donnant le nombre de caractères inscrits.

La fin de liste contiendra:

- le nombre de records en input
- le nombre de records références
- le nombre de records ponctuations
- le nombre de records mots
- le nombre de records en output

III-3-1-3/ Description de l'output de la phase d'analyse

Les records élémentaires seront construits à partir des records obtenus à la phase enregistrement, ils sont de deux types:

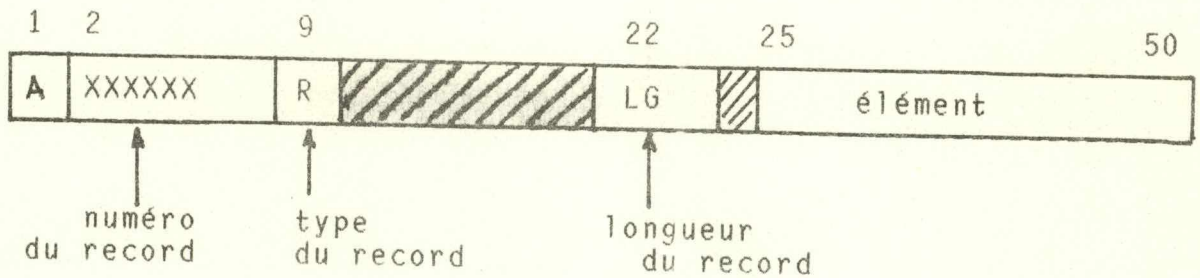
- les records références
- les records textes (mots et ponctuations)

Chaque record construit aura :

- un numéro de progression
- un code spécifiant son type (Référence, mot ou ponctuation)
- une référence (Sourate + verset + Hizb)
- un nombre spécifiant sa longueur (26 caractères pour les mots arabes, y compris les voyelles).
- l'élément.

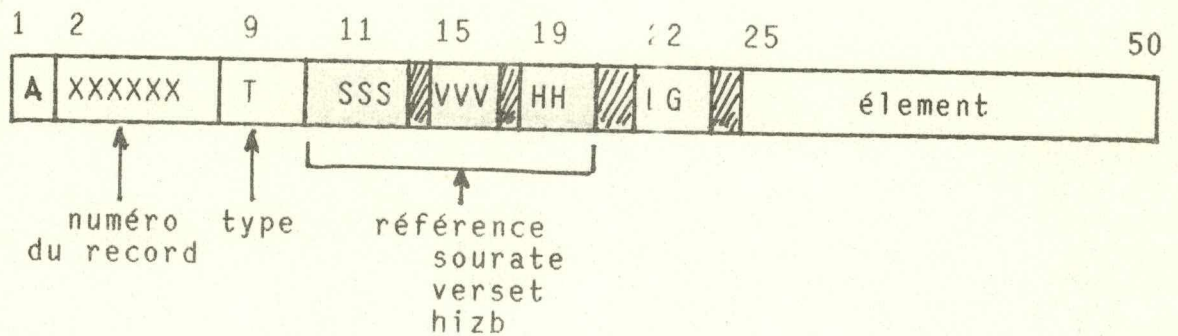
a - Records références

Ils sont copiés sur le fichier output, tels qu'on les trouve à l'input sans aucun traitement.



b - Records ponctuations et mots

Ils sont construits à partir du record texte :



P= ponctuation
M= mot

ELEMENT

LG*HH*VVV*SSS*T*N-REC

ELEMENT	LG	HH	VVV	SSS	T	N	REC
مس 3	3					R	1
سُورَةُ	7	5	0	3	M		2
آل	3	5	0	3	M		3
عِمْرَانُ	9	5	0	3	M		4
مَدْيَنَةُ	10	5	0	3	M		5
ا 1	3					R	6
بِسْمِ	6	5	1	3	M		7
اللّٰه	6	5	1	3	M		8
الرَّحْمٰنِ	10	5	1	3	M		9
الرَّحِیْمِ	9	5	1	3	M		10
اٰلِیْمٍ	6	5	1	3	M		11
ا 2	3					R	12
اللّٰه	6	5	2	3	M		13
اٰلِیْمٍ	2	5	2	3	M		14
رَبِّهِ	6	5	2	3	M		15
رَبِّهِ	4	5	2	3	M		16
مُو	4	5	2	3	M		17
الْحَمْدُ	6	5	2	3	M		18
الْمَشْوُومُ	9	5	2	3	M		19
ا 3	3					R	20
رَبِّهِ	6	5	3	3	M		21
عَلِیْمِكِ	8	5	3	3	M		22
الْکِتَابِ	9	5	3	3	M		23
بِالْحَقِّ	8	5	3	3	M		24
مُحَمَّدًا	9	5	3	3	M		25

لِمَا	5	5	3	3	M	26
لَمَّا	6	5	3	3	M	27
بَدَانَا	8	5	3	3	M	28
وَا	2	5	3	3	M	29
اَنْزَلْ	8	5	3	3	M	30
التَّوْبَةَ	11	5	3	3	M	31
وَا	2	5	3	3	M	32
الْاَنْجِيلِ	10	5	3	3	M	33
ا	4	3			R	34
سِوَا	4	5	4	3	M	35
فَمِثْلُ	6	5	4	3	M	36
مُدَّيْنِ	5	5	4	3	M	37
لِلنَّاسِ	8	5	4	3	M	38
وَا	2	5	4	3	M	39
اَنْزَلْ	8	5	4	3	M	40
الْفُرْقَانَ	11	5	4	3	M	41
اِنَّ	4	5	4	3	M	42
التَّوْبَةَ	8	5	4	3	M	43
كُفْرًا	8	5	4	3	M	44
بِآيَاتِ	8	5	4	3	M	45
اللَّهِ	6	5	4	3	M	46
لَهُمْ	6	5	4	3	M	47
عَذَابِ	7	5	4	3	M	48
هُدًى	7	5	4	3	M	49
وَا	2	5	4	3	M	50
اللَّهِ	6	5	4	3	M	51
عَرَبِ	7	5	4	3	M	52

د' و	3	5	4	3	M	53
الشمس	10	5	4	3	M	54
#	1	5	4	3	P	55
ا 5	3				R	56
لر	4	5	5	3	M	57
الله	6	5	5	3	M	58
لا	2	5	5	3	M	59
مخفي	7	5	5	3	M	60
عليه	8	5	5	3	M	61
شيء	5	5	5	3	M	62
في	3	5	5	3	M	63
الار	7	5	5	3	M	64
و	2	5	5	3	M	65
لا	2	5	5	3	M	66
في	3	5	5	3	M	67
السماء	9	5	5	3	M	68
ا 6	3				R	69
مو	4	5	6	3	M	70
الذي	6	5	6	3	M	71
بموتكم	12	5	6	3	M	72
في	3	5	6	3	M	73
الار	10	5	6	3	M	74
كيف	6	5	6	3	M	75
بشاه	7	5	6	3	M	76
لا	2	5	6	3	M	77
لله	6	5	6	3	M	78
للا	4	5	6	3	M	79

كلمة	4	5	6	3 M	80
المرادفات	9	5	6	3 M	81
الحكيم	9	5	6	3 M	82

RECORDS MOTS = 74
 RECORDS PONCTUATIONS = 1
 RECORDS REFERENCES = 7
 RECORDS INPUT = 13
 RECORDS OUTPUT = 82

FICHER ELEMENTS

III-3-2/ Tri alphabétique

Il faudrait noter que pour les textes arabes, le sens de la lecture ou l'écriture se fait de droite à gauche; par contre l'écriture en mémoire se fera toujours de gauche à droite.

Pour la lemmatisation, le tri intéresse seulement les mots, donc les références, ponctuations, codes et commandes n'interviennent pas dans le tri.

III-3-2-1- Principe du tri

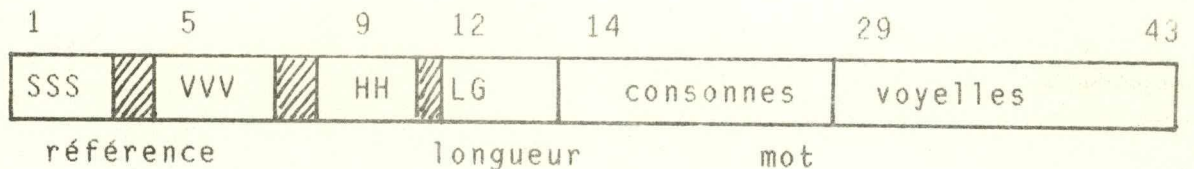
Le tri s'effectuera à partir des enregistrements créés dans la phase "construction des records élémentaires". Il sera nécessaire avant le tri d'effectuer par programme, la dispersion des consonnes et des voyelles sur les records du type mot, car on devra trier ces derniers d'abord sur les consonnes et ensuite sur les voyelles.

En output, les records auront le même format que ceux trouvés à l'input, à la seule différence qu'ils seront triés par ordre alphabétique normal et que leur numéro de progression suivra l'ordre alphabétique.

Une fois triés, ces records serviront d'input pour la phase récapitulation.

III-3-2-2/ Format record input tri

Le format des données à trier est le suivant:



Les exigences de l'impression sont que toutes les en-lettres doivent être composées avec une lettre dans son même espace (voyelle en arabe, accents en français, italien).

Afin d'écrire la forme telle qu'elle a été enregistrée, il faudra transformer l'output tri en output tri écriture normale. (voir programme tri et mélange en annexe).

NB1 - Lors de la dispersion des consonnes et des voyelles, certains caractères posent des problèmes, ce sont :

- Le lam 'alif (ل) qui est représenté par un seul code machine, devant être remplacé par les caractères lam (ل) et alif (ل) pour obtenir deux codes différents.
- La šada et le tanwin (َ , ِ , ِ) qui sont représentés par deux codes machines alors qu'en réalité, il ne doit y avoir qu'un seul code. Pour cela il aura fallu avant le tri, recoder les caractères suivants :

- ا a été remplacé par le caractère 1
- آ a été remplacé par le caractère 2
- أ a été remplacé par le caractère 3

SSS	VW	HH	LG	Voyelles	Consonnes
3	0	5	7		سورة
3	0	5	3		آل
3	0	5	9		عمران
3	0	5	10		مدينة
3	1	5	6		بسم
3	1	5	6		الله
3	1	5	10		الرحمن
3	1	5	9		الرحيم
3	1	5	6		ألم
3	2	5	6		الله
3	2	5	2		لا
3	2	5	6		له
3	2	5	4		لا
3	2	5	4		هو
3	2	5	6		الحي
3	2	5	9		القيوم
3	3	5	6		نزل
3	3	5	8		عليك
3	3	5	9		الكتاب
3	3	5	8		بالحق
3	3	5	9		مصدقاً
3	3	5	5		لما
3	3	5	6		بين
3	3	5	8		يديه
3	3	5	8		و
3	3	5	8		أنزل
3	3	5	11		التوراة
3	3	5	2		و
3	3	5	10		الانجيل
3	4	5	4		من
3	4	5	6		قبل
3	4	5	5		هدى
3	4	5	8		للناس
3	4	5	2		و
3	4	5	8		أنزل
3	4	5	11		الفرقان
3	4	5	4		إن
3	4	5	8		الدين
3	4	5	8		كفروا
3	4	5	8		بآيات
3	4	5	6		الله
3	4	5	6		لهم
3	4	5	7		عذاب
3	4	5	7		شديد
3	4	5	2		و
3	4	5	6		الله
3	4	5	7		عزيم
3	4	5	3		ذو
3	4	5	10		انتقام

3	5	5	4	ل	ل
3	5	5	6	الله	ل
3	5	5	2	ل	ل
3	5	5	7	يخفي	ل
3	5	5	8	عليه	ل
3	5	5	5	شيء	ل
3	5	5	3	في	ل
3	5	5	7	الارض	ل
3	5	5	2	و	ل
3	5	5	2	ل	ل
3	5	5	3	في	ل
3	5	5	9	السماء	ل
3	6	5	4	هو	ل
3	6	5	6	الذي	ل
3	6	5	12	بصوركم	ل
3	6	5	3	في	ل
3	6	5	10	الارحام	ل
3	6	5	6	كيف	ل
3	6	5	7	بشاء	ل
3	6	5	2	ل	ل
3	6	5	6	رنا	ل
3	6	5	4	رنا	ل
3	6	5	4	هو	ل
3	6	5	9	العزير	ل
3	6	5	9	الحكيم	ل

FICHER DES MOTS NON TRIE

3	4	5	6		فيل
3	4	5	8		كفروا
3	6	5	6		كيف
3	6	5	2		لا
3	5	5	2		لا
3	5	5	2		لا
3	2	5	2		لا
3	4	5	8		لباس
3	3	5	5		لما
3	4	5	6		لهم
3	0	5	10		مدنية
3	3	5	9		مصدقاً
3	4	5	4		من
3	3	5	6		نزل
3	4	5	5		عدي
3	6	5	4		هو
3	6	5	4		هو
3	2	5	4		هو
3	3	5	2		و
3	5	5	2		و
3	4	5	2		و
3	4	5	2		و
3	3	5	2		و
3	5	5	7		يخفي
3	3	5	8		يديه
3	6	5	7		يشاء
3	6	5	12		بصوركم

FICHER DES MOTS TRIES

III-3-2-3/ Enregistrement et restitution d'un texte

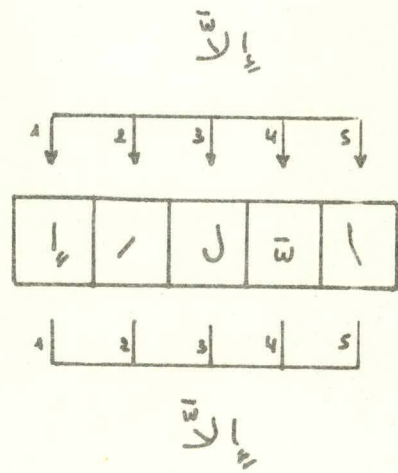
L'enregistrement du texte arabe dans la mémoire de l'ordinateur se fait de gauche à droite. A chaque lettre de l'alphabet lui correspond un code informatique (ou code machine) qui est une suite de " 0 " et " 1 ". Cette suite est appelée octet ou byte en informatique.

Par exemple, l'enregistrement et la restitution du mot " 'illa" montrent la difficulté de l'opération.

- Le mot est stocké en mémoire de l'ordinateur sous forme d'une suite de cinq lettres (voir figure)

- Lors de l'enregistrement la lettre " ﻻ " lam-alif est séparée par l'ordinateur en lam et alif bien que l'écriture arabe impose que le lam et alif quand ils sont voisins doivent être remplacés par une ligature lam-alif " ﻻ ". Et réciproquement, lors de la restitution du texte, si l'ordinateur rencontre le lam et le 'alif voisins, il les remplace par le lam-alif.

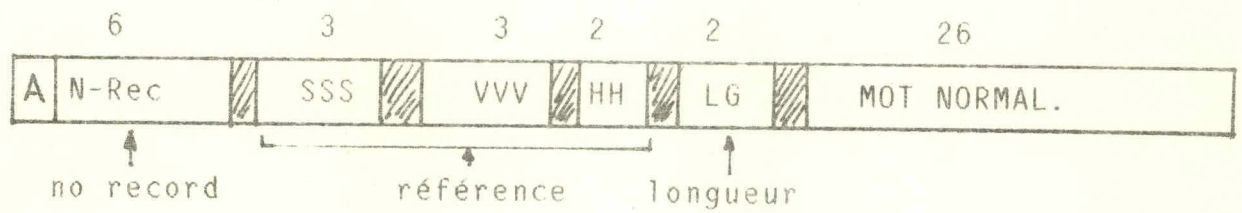
- Après l'assemblage de toutes les formes, l'ordinateur affiche le mot sur l'écran ou l'imprime sur le papier de droite à gauche.



III-3-2-4/ Contrôle du tri

Le contrôle du tri ne demande qu'une liste du premier et du dernier record trié et le total des records en output.

Format des records en output



Ainsi on a déjà produit trois fichiers :

- 1/ Fichier de l'enregistrement
- 2/ Fichier des records élémentaires en ordre du texte
- 3/ Fichiers des records mots triés en ordre alphabétique avec toutes les références.

ELEMENT

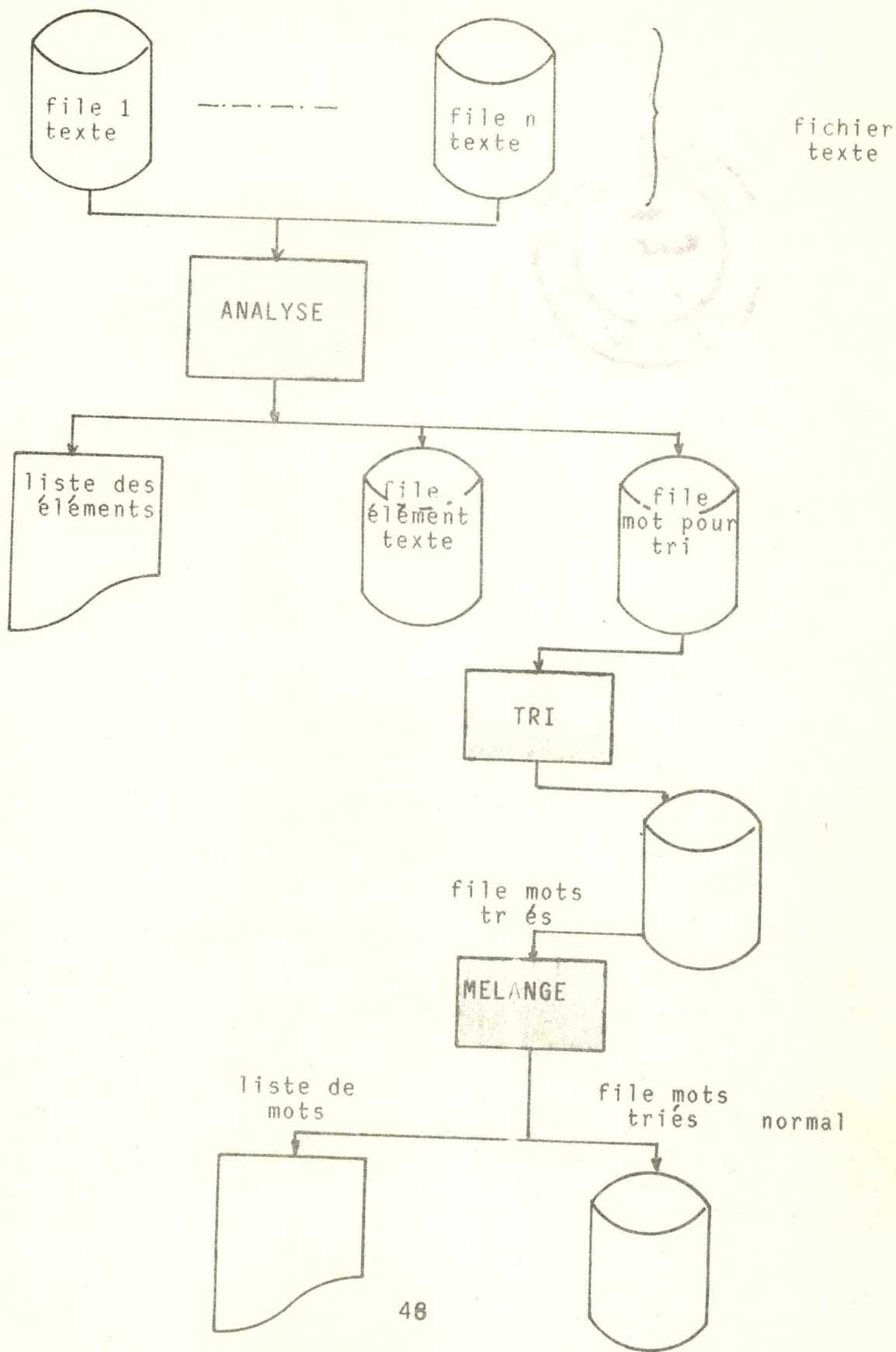
LG*HH*VVV*SSS*N-REC

ELEMENT	LG	HH	VVV	SSS	N-REC
آل	3	5	0	3	1
آل 1	6	5	1	3	2
آل 1	8	5	4	3	3
آل 1	8	5	3	3	4
آل 1	4	5	6	3	5
آل 1	4	5	2	3	6
آل 1	6	5	6	3	7
آل 1	6	5	2	3	8
آل 1	4	5	4	3	9
آل 1	4	5	5	3	10
آل 1 حَام	10	5	6	3	11
آل 1 صِد	7	5	5	3	12
آل 1 نَجْمِل	10	5	3	3	13
آل 1 التَّوَدَاة	11	5	3	3	14
آل 1 الحَكْرِم	9	5	6	3	15
آل 1 الحَوْد	6	5	2	3	16
آل 1 القَدْرِي	6	5	6	3	17
آل 1 القَدْرِيْن	8	5	4	3	18
آل 1 الرِّقْحَمُون	10	5	1	3	19
آل 1 الرِّقْحَمِيم	9	5	1	3	20
آل 1 السَّمَام	9	5	5	3	21
آل 1 العَرَبِيْر	9	5	6	3	22
آل 1 الفُرْقَان	11	5	4	3	23
آل 1 العَمَشُوْم	9	5	2	3	24
آل 1 الكِتَاب	9	5	3	3	25
آل 1 اللّٰه	6	5	5	3	26

اللَّه'	6	5	2	3	27
اللَّه'	6	5	4	3	28
اللَّه'	6	5	4	3	29
اللَّه'	6	5	1	3	30
الْمُعْتَمِد	10	5	4	3	31
بِآيَاتِ	8	5	4	3	32
بِالْحَقِّ	8	5	3	3	33
بِسْمِ	6	5	1	3	34
بِمَنْ	6	5	3	3	35
ذ' و	3	5	4	3	36
سُورَةٍ	7	5	0	3	37
شَدِيدٌ	7	5	4	3	38
شَجِيهٌ	5	5	5	3	39
عَذَابٌ	7	5	4	3	40
عَزِيزٌ	7	5	4	3	41
عَلَيْكَ	8	5	3	3	42
عَلَيْهِ	8	5	5	3	43
عَمْرَانٌ	9	5	0	3	44
فِي	3	5	5	3	45
فِي	3	5	5	3	46
فِي	3	5	6	3	47
فَمِنْ	6	5	4	3	48
كَمَرُوا	8	5	4	3	49
كَيْفَ	6	5	6	3	50
لَا	2	5	6	3	51
لَا	2	5	5	3	52
لَا	2	5	5	3	53

لَيْلَتَانِ	2	5	2	3	54
لَيْلَتَانِ	8	5	4	3	55
لَيْلَانِ	5	5	3	3	56
لَيْلِمُ	6	5	4	3	57
لَيْلَتُهُ	10	5	0	3	58
لَيْلَتَانِ	9	5	3	3	59
لَيْلَانِ	4	5	4	3	60
لَيْلَانِ	6	5	3	3	61
لَيْلَانِ	5	5	4	3	62
لَيْلَانِ	4	5	6	3	63
لَيْلَانِ	4	5	6	3	64
لَيْلَانِ	4	5	2	3	65
لَيْلَانِ	2	5	3	3	66
لَيْلَانِ	2	5	5	3	67
لَيْلَانِ	2	5	4	3	68
لَيْلَانِ	2	5	4	3	69
لَيْلَانِ	2	5	3	3	70
لَيْلَانِ	7	5	5	3	71
لَيْلَانِ	8	5	3	3	72
لَيْلَانِ	7	5	6	3	73
لَيْلَانِ	12	5	6	3	74

REGROUPEMENT CONSONNES VOYELLES



III-3-3/ Récapitulatif des formes

Dans cette partie, on entre dans le METALANGAGE car nous entendons par "Forme" chaque type de mots identiques.

Un mot peut être identique à la forme selon les signifiants et les signifiés, ou seulement selon l'un des deux.

Exemple :

- . DĪN : C'est une forme graphiquement identique, mais sémantiquement ce sont deux mots différents

DĪN (1): pratiquer une religion

DĪN (2): contracter une dette

- . 'AQT et WAQT (ou 'ALLAH et RABB)

Pourraient être considérés sémantiquement identiques mais graphiquement différents.

La récapitulation, ici, aboutit aux formes graphiquement identiques.

La lemmatisation s'occupera d'en examiner l'identité sémantique ou son contraire.

III-3-3-1/ Schème de la récapitulation

Input: On a les mots en ordre alphabétique, on prend le premier mot et on le transfère en sortie avec un "1" dans le compteur.

On compare ensuite le mot suivant au mot précédent; s'ils sont identiques, il suffit d'incrémenter le compteur sans aucun transfert en sortie; s'ils sont différents il faut recopier le record précédent en sortie, et on ouvre un autre record avec une remise à zéro du compteur.

A la fin on aura autant de formes différentes, chacune avec sa fréquence (le nombre de mots identiques).

Output: Chaque record en sortie comportera :

- un numéro ascendant continu des formes
- la (les) Forme(s) (voir 3-3-2)
- la fréquence globale de la forme
- la référence de quelques mots appartenant à cette forme

Le but de ces références est d'avoir la possibilité de chercher les mots dans leurs contextes.

En effet, les formes de basses fréquences sont rares; parmi elles on pourra découvrir comme erreurs les formes impossibles.

Le nombre de références est fonction de la longueur du record en output.

Au niveau de la récapitulation, il va de soi qu'en arabe, quand le texte est vocalisé, la reconnaissance de l'identité graphique se base sur les consonnes et les voyelles.

III-3-3-2/ Le fichier des formes graphique

Chaque zone forme contient trois éléments:

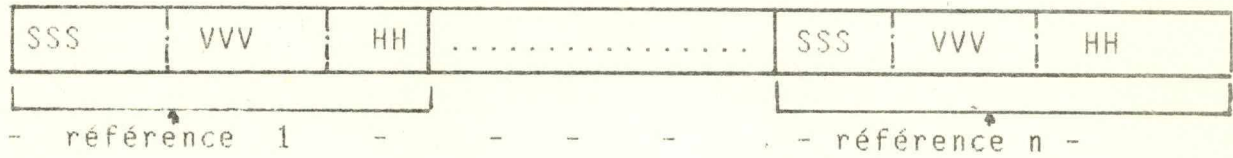
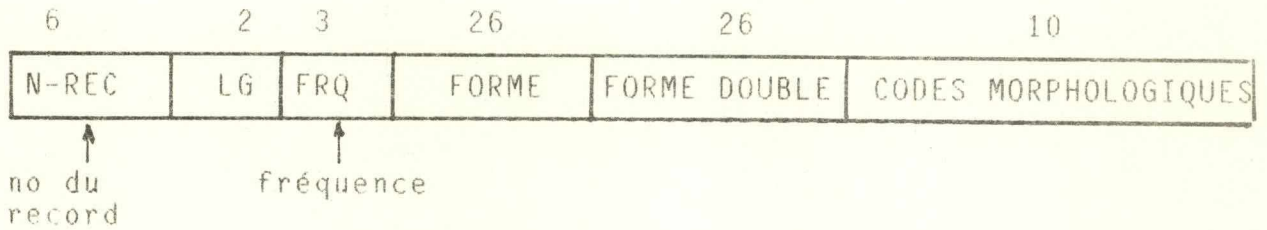
- Forme
- le nombre de caractères
- le code initial de la majuscule si elle existe, quand il est nécessaire de préciser que la forme a une majuscule (n'existe pas pour l'arabe).

Serviront d'input les records obtenus dans la phase de tri .

En output, chaque record obtenu dans la phase récapitulation devra comporter les informations suivantes:

- le numéro de progression du record
- le nombre de caractères occupés par la forme
- la zone forme du mot arabe cadrée à droite
- une copie de la zone forme qui sera utilisée dans la lemmatisation
- une zone réservée aux codes morphologiques
- la fréquence globale de la forme
- les premières références des premiers mots appartenant à la forme.

Format du record



On aura ainsi produit un autre fichier qui contiendra les formes graphiques avec leur fréquence et quelques détails de leurs références (toutes les références se trouvent dans le fichier des mots).

Contrôle de l'output

Il y a deux possibilités pour effectuer le contrôle de l'output :

- soit avoir la liste complète du fichier qui sera par ordre alphabétique normal de la forme.

- soit avoir la liste partielle pour le contrôle de l'opération de récapitulation.

Dans les deux listes, le total de toutes les fréquences est nécessaire, ce qui nous permettra de contrôler s'il correspond au nombre de mots en input.

On peut demander également une liste des fréquences de plusieurs façons :

- Par nombre décroissant des fréquences suivi par le nombre des formes qui les vérifie sans écrire aucune forme.

- Par ordre de fréquences décroissantes de toutes les formes qui ont une fréquence au dessus d'un certain seuil (10 pour 1000 mots), et au-dessous du seuil, le résumé numérique du nombre de formes vérifiant chaque fréquence; ou chaque groupement de fréquences (90-99/80-99/...)

On pourrait aussi demander la liste complète de toutes les formes après les avoir triées suivant la fréquence décroissante d'une part et de l'alphabet d'autre part.

Dans tous les cas, la liste des formes est obligatoire pour faire la lemmatisation.

Cette liste doit :

- 1/ Laisser de l'espace entre chaque forme pour y faire par la suite des rajouts à la main
- 2/ Etre en ordre alphabétique des formes
- 3/ Contenir deux zones marquées pour y mettre le lemme et les codes morphologiques.

III-3-3-3/ Le Fichier Index-contraire

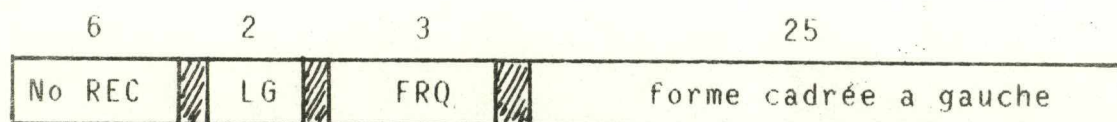
Parfois il est nécessaire d'écrire les formes alignées à droite afin :

- D'étudier les terminaisons des mots
- D'effectuer la recherche des mots composés avec un même final
- De permettre également la segmentation morphématique qui découpe les formes en constantes radicales et morphèmes.

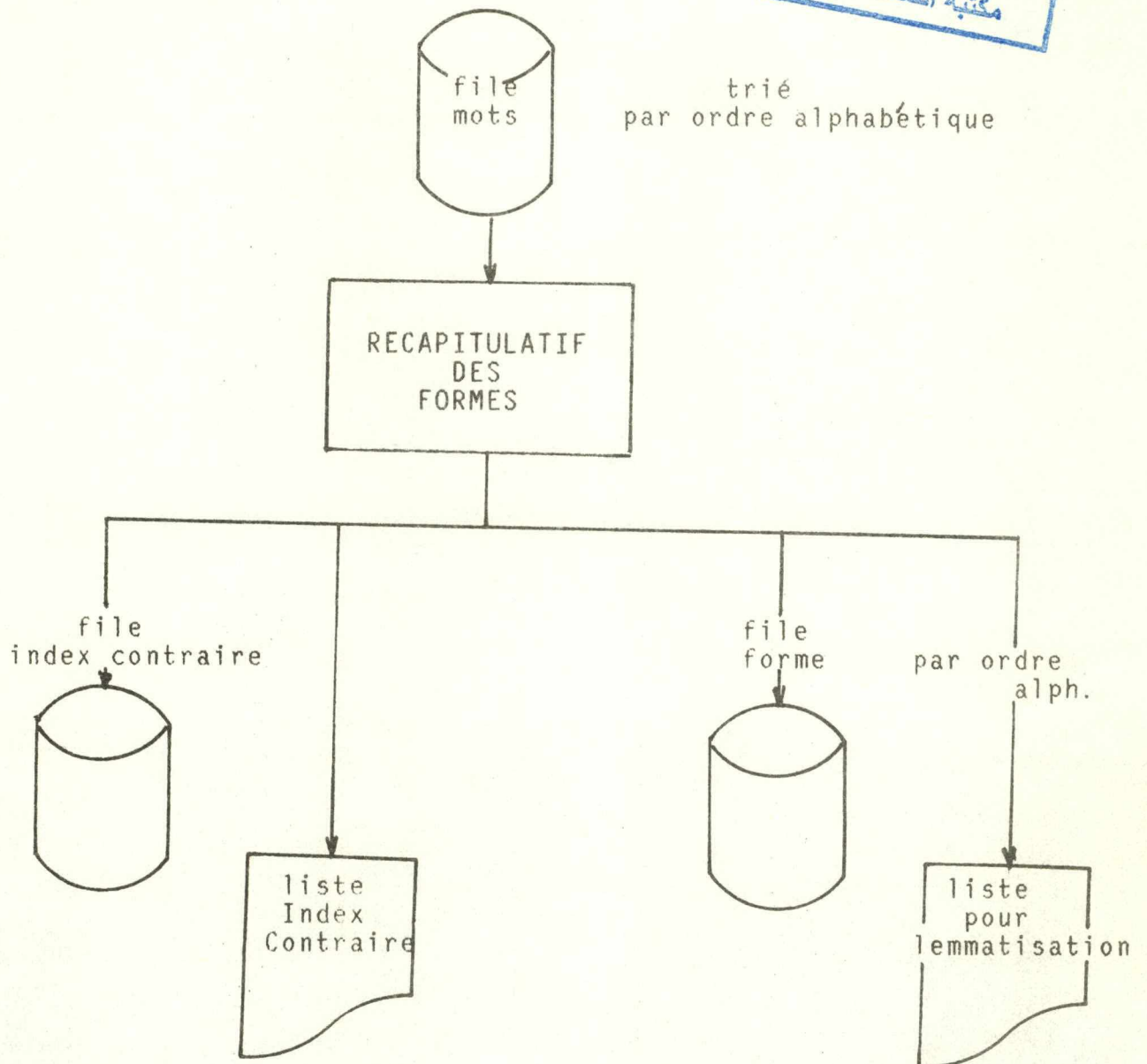
Le fichier index-contraire sera construit à partir des records obtenus dans la phase tri.

IL comportera les éléments suivants:

- le numéro du record
- la longueur de la forme
- la fréquence globale de la forme
- la forme cadrée à gauche



Les références des mots appartenant a cette forme ne sont pas obligatoires, car on les trouve dans le fichier des lemmes. En output, on obtiendra une liste triée par ordre alphabétique de la forme.



R é c a p i t u l a t i f d e s f o r m e s

III-4/ La lemmatisation

III-4-1/ Présentation

Dans la phase analyse, il y a eu découpage du texte en unités graphiques séparées par deux blancs (mots informatiques). A partir de là, il y a eu regroupement des mêmes unités graphiques en formes sans tenir compte du contexte. A chaque forme est associée une occurrence (nombre de mots de même identité graphique).

Dans un texte, il s'avère qu'un même mot subit des variations qui altèrent sa consistance graphique mais sans en changer son sens. Ces variations sont dues dans la plupart des cas aux flexions.

Donc, à chaque mot sont associées une ou plusieurs formes. Il est alors indispensable de regrouper les formes d'un même mot en une seule entité. Cette entité représentera pour nous le lemme.

Dans ce qui suit, nous allons essayer de donner la définition du lemme et présenter également l'opération de lemmatisation.

a /- Définition du lemme

Le lemme d'une façon générale est le mot "titre" ou "entrée" dans un dictionnaire, où sont représentées toutes les flexions de ce mot. Ce mot titre, signifie l'unité centrale du sens du mot, indépendamment de ces variations flexibles.

Donc la fonction du lemme est double; elle permet de :

- Représenter toutes les flexions des différentes formes
(mot titre ou entrée)
- Unifier le sens du mot (unité de base).

b /- La lemmatisation

Le mot lemmatisation est dérivé du mot lemme pour signifier l'ensemble des opérations qui permettent de réunir toutes les formes sous leur lemme respectif.

En général, il existe la lemmatisation morphologique et la lemmatisation syntaxique.

Lemmatisation morphologique

Elle ne tient pas compte du contexte, et s'applique aux formes récapitulées graphiques.

La lemmatisation morphologique est " granulaire ", car elle ne peut porter que sur des mots individuels, l'un après l'autre. Elle est aussi " initiale " car on commence par elle, on peut l'enrichir, l'approfondir et la spécifier.

Lemmatisation syntaxique

Elle considère toutes les couches sémantiques que le contexte ajoute ou précise au mot. Elle tient compte du contexte; c'est pour cela que cette lemmatisation demande en premier lieu une concordance. Elle a l'avantage d'être complète et riche, mais par contre elle est lourde et demande beaucoup de temps.

Principe de la lemmatisation

La lemmatisation a toujours été faite d'une façon :

- Conventiennelle
- Instinctive
- Non standardisée
- Différenciée.

(cf. Lemmatisation du professeur R. Busa)

Il est nécessaire de s'appuyer sur un dictionnaire standard qui servira pour toute la lemmatisation du texte. En principe et pour plus de simplicité, le plus grand nombre de flexions doit être groupé sur le plus petit nombre de lemmes.

L'opération de lemmatisation est confrontée à plusieurs problèmes concernant l'unité du lemme.

En effet, quand des mots identiques ont des sens différents, doit-on faire un lemme ou deux ?

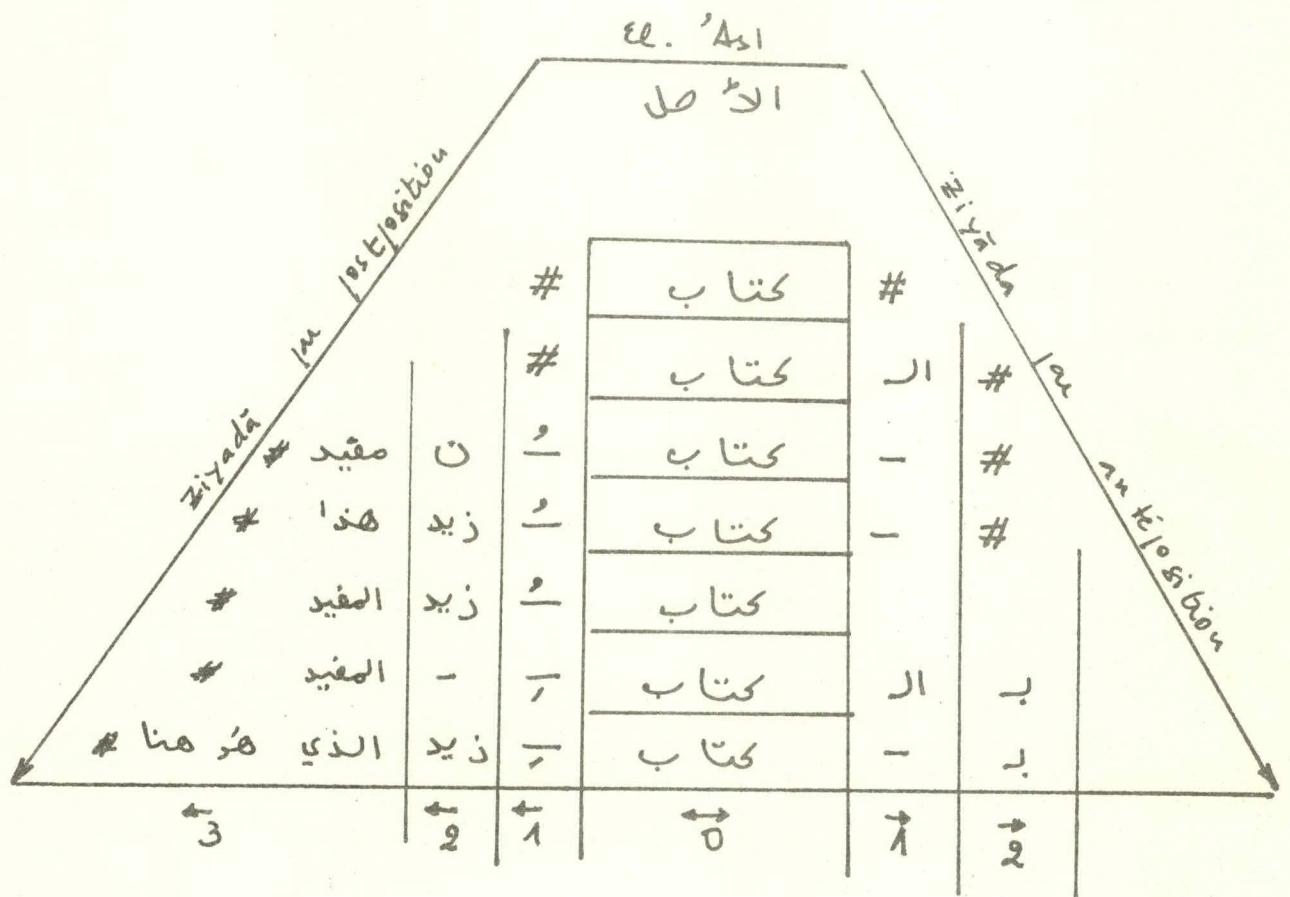
Afin de répondre à tous ces problèmes, et selon l'expérience du professeur R. Busa, nous allons donner une ébauche d'un premier menu pour faire la lemmatisation en général.

Mais au préalable, il est indispensable de connaître la structure du mot en arabe. Pour cela, nous allons présenter les modèles de construction (Schèmes) du Ism et du verbe, et cela en nous basant sur la théorie du professeur Hadj-Salah.

III-4-2 / Les schèmes générateurs du ism et du verbe

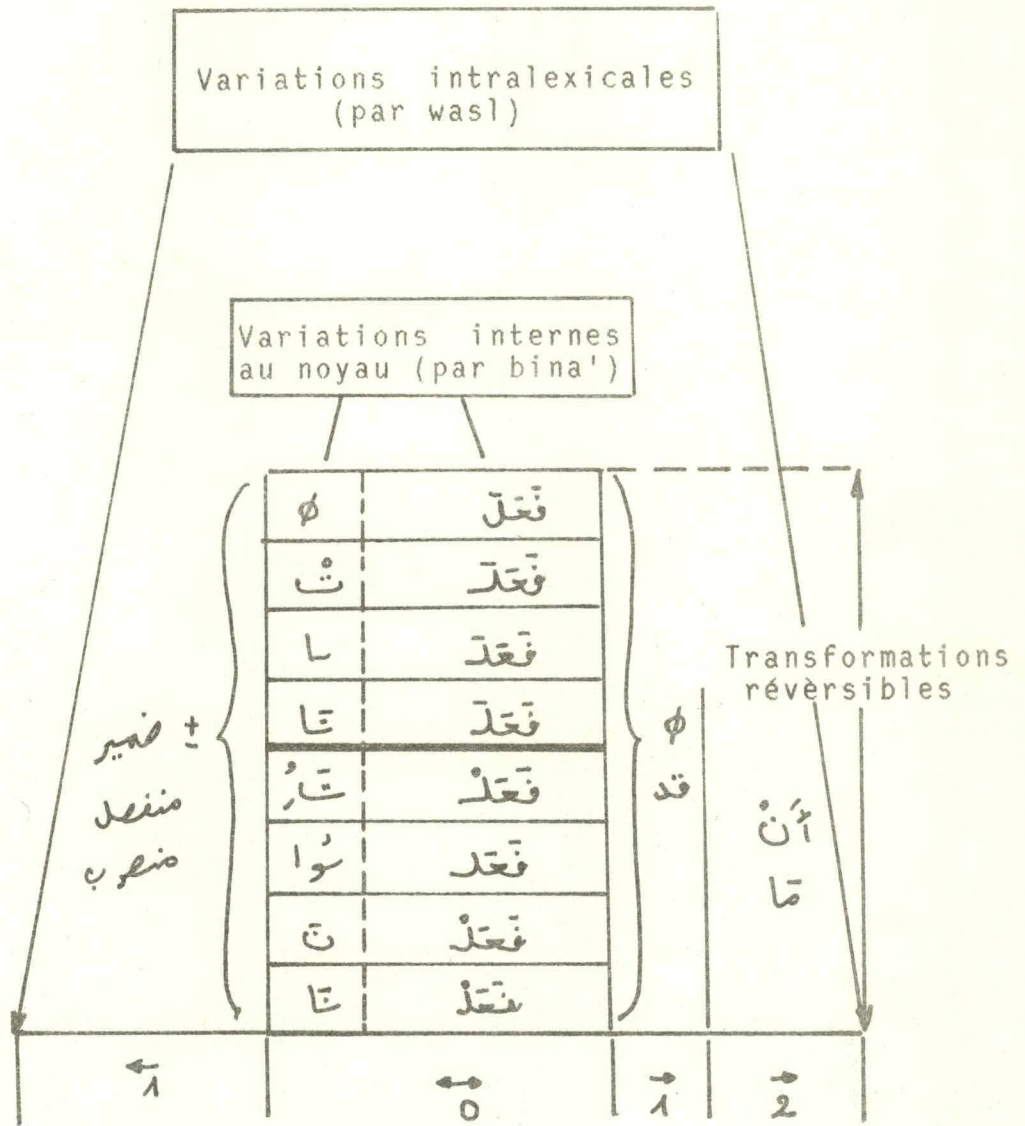
Il existe 1 schème pour le ism et 3 schèmes pour le verbe (accompli, inaccompli, impératif)

- Modèle du Ism (El-Ism) - (1)



1) Cf. A.Hadj-Salah, Linguistique Arabe et linguistique générale, pp. 669 - 677 (tome II)

- ↔ 0 = Mawdi⁶ du ism, élément central ou noyau de la lexie
- 1 = Mawdi⁶ du déterminant identifiant (harf al-ta⁶rif)
- 2 = Mawdi⁶ de l'exposant annectif (harf al-ğarr)
- ← 1 = Mawdi⁶ de la désinence casuelle
- ← 2 = Mawdi⁶ du déterminant par annexion directe (mudāf ilayh)
c'est aussi le mawdi⁶ du tanwin
- ← 3 = Mawdi⁶ du déterminant caractérisant.
Ce déterminant peut être :
- une SIFA dérivée (adjectif codérivé du verbe)
- ou
- un MAWSŪL avec sa sila (relatif + séquence introduite
par ce relatif)
- ou
- un démonstratif
- ou
- tout autre item auquel on peut conférer le rôle de caractérisant.

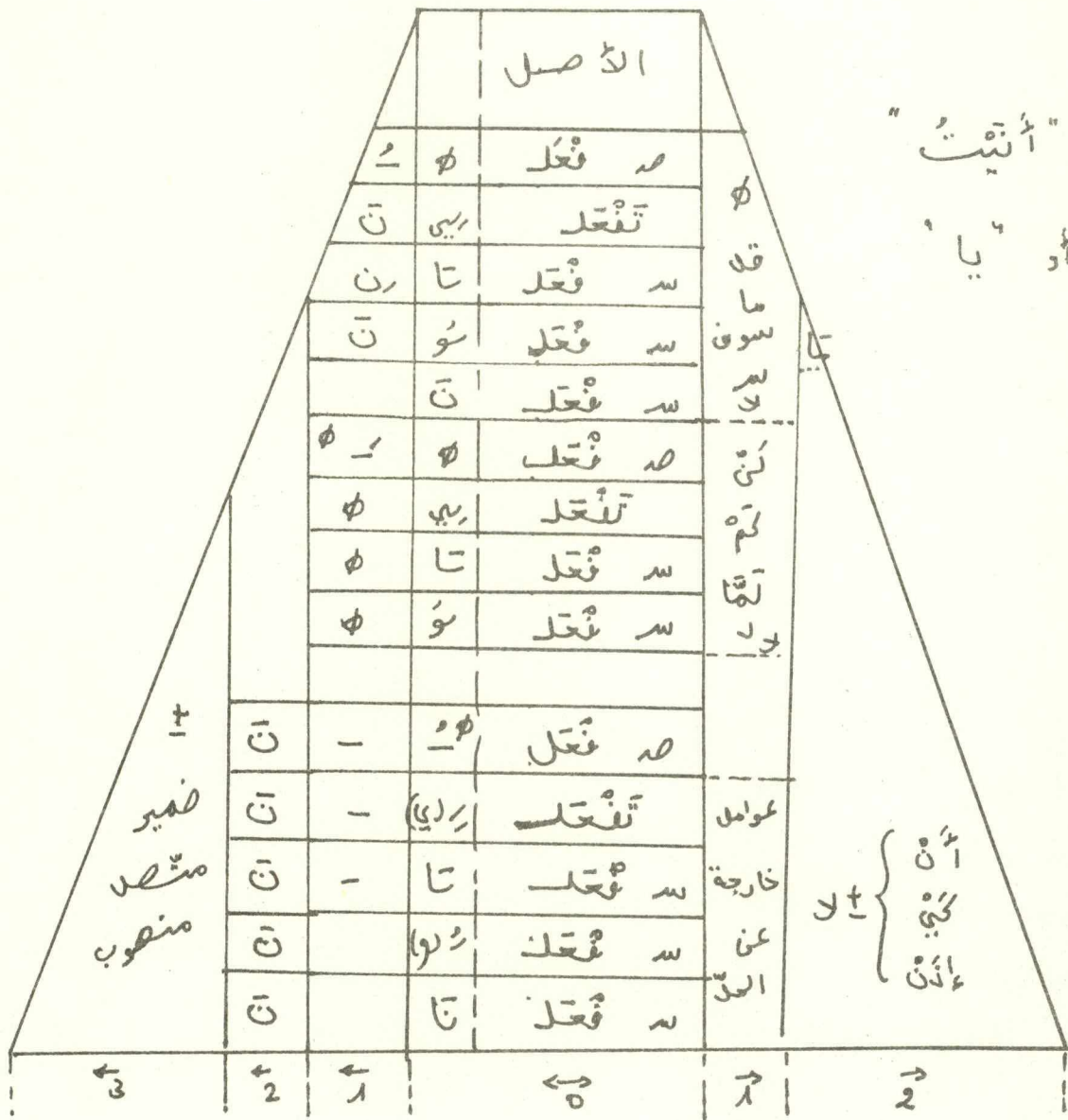


1) Cf. A.Hadj Salah,

Linguistique Arabe et linguistique générale, pp. 693 - 695 (tome II)

- ↔ 0 = Mawdi^c de la lexie verbale (noyau)
 Asl interne à ce noyau est le ḡā'ib : Fa'ala
 (3ème pers. mas. sing.)
- 1 = Situation du discours (lieu)
- 2 = Est le lieu où apparaissent " an " ou " mā ".
 Ceux sont des convertisseurs (objets qui permettent de
 relier un verbe à un autre verbe).
- ← 1 = Est le lieu où fonctionnent les ajouts pronominaux qui
 déterminent l'identité du régime du verbe.

Modèle du Verbe Inaccompli (1)



1) Cf. A.Hadj Salah, Linguistique Arabe et linguistique générale, pp. 699 - 701 (tome II)

- ↔ 0 = L'unité centrale. 'Asl interne à ce noyau est : Ca-F⁶al Ø
(qui représente la 1ère pers. sing., la 2ème pers. mas. sing. et la 3ème pers. sing.)
- ← 1 = Est celui de la marque flexionnelle du Mudāri⁶
- ← 2 = Est celui de la marque " -nna " qui dénote une assertion énergique (= nūn al-tawktīd).
- ← 3 = Ajouts pronominaux.
- 1 = Est le lieu où apparaissent les exposants qui doivent ajouter une précision à la valeur initialement indéterminée de l'inaccompli.
- 2 = Est le lieu où apparaissent des " convertisseurs ".

Modèle du Verbe Impératif (1)

		ϕ	فَعَلْ	(أ)
		بِي	فَعَلْ	(أ)
		تَا	فَعَلْ	(أ)
		وَا	فَعَلْ	(أ)
		نَ	فَعَلْ	(أ)
	نَ	ϕ	فَعَلْ	(أ)
	نَ	بِي	فَعَلْ	(أ)
	نَ	تَا	فَعَلْ	(أ)
	نَ	وَا	فَعَلْ	(أ)
	نَ	!	فَعَلْ	(أ)

Ajouts
pronominaux

nūn
al tawkid

flexion
u'xiote ١٥٥

pronom

Verbe (Asl interne à un noyau est

(i) F° al (2^{em} pers. mas. sing)

Combinés

1) Cf. A. Hadj Salah, Linguistique Arabe et linguistique générale, pp. 702 - 710 (tome II)

III-4-3/Ebauche d'un menu pour faire de la lemmatisation

La lemmatisation consiste à :

- assigner à chaque forme son lemme
- assigner à chaque lemme des codes morphologiques.

Le problème est: que faut-il faire quand une forme appartient à plusieurs lemmes?

III-4-3-1/ Assignation de lemmes aux formes

L'assignation de lemmes aux formes consiste à :

- 1/ Ajouter dans la zone qui a été réservée par le programme de récapitulation des formes, le mot du lemme.
- 2/ Ajouter, si l'on veut, le code numérique qui correspondrait au mot du lemme.
On peut imaginer deux façons pour obtenir le code numérique :
 - . Assigner à chaque lettre alphabétique un numéro (classement alphabétique)
 - . Ou assigner à chaque lemme un indicatif nom, verbe, invariable, etc.....

III-4-3-2/ Codes morphologiques

Il faut se contenter d'un "minimum suffisant" pour ne pas se retrouver avec des listes illisibles.

En général, on trouve:

- Catégorie des noms

أنواع الأسماء

Qui regroupe:

- Noms propres qui indiquent une entité singulière (Algérie, Italie, etc....).

الجزائر

- Tous les dérivés proches avec des désinences et des terminaisons fixées (Algérie, Algériens,...)

الجزائري

- Noms communs

ذئب

- Les formes dérivées

.tous les pronominaux (subjonctifs, adjectifs, autres).

ضربوا؛ ها؛

.le nom indéclinable.

أمس

Catégorie des invariables

بِاسْمِ مَبْنِي

On trouve :

- Prépositions

حروف الجر

بِ رَضِيَ

- Conjonctions et autres adverbes

حروف العطف وَ

هَذَا ثُمَّ وَ

الظروف

- Autres invariables

خلات الأخرى

هَذَا

Catégorie des flexions verbales

الأفعال

ذَهَبُوا

Exemple

FORME	LEMME	CODE
I لِذَهَابٍ I	I ذَهَبَ I	I principal I
I لِذَهَابٍ I	I ذَهَبَ I	I secondaire I

III-4-3-3/ Formes qui appartiennent à plusieurs

lemmes

Une forme peut appartenir à plusieurs lemmes pour plusieurs raisons:

- A cause de la composition instable et stable

كُتِبَ : composé instable

مَلْعُونٌ بِهِ : composé stable

- A cause de l'homographe

Verbe
 (من ...) superlatif
 أَحْسَنُ

a/ Formes composées instables et stables

Les formes composées instables de mot ne sont ici jamais lemmatisées comme telles; chacun de leurs composants va avoir son propre lemme.

La solution consiste à multiplier entièrement cette forme autant de fois qu'il y a de composants.

Il faut fixer également des règles afin que les mots du texte soient lemmatisés selon les lemmes que l'on a codifiés comme principaux.

On peut marquer par exemple avec un seul code l'existence d'un pronom sujet sans le déterminer, ou bien il faudrait codifier celui parmi les 14 en arabe (ce raisonnement est valable ainsi pour les autres catégories).

Pour les formes composées instables, il faudrait se référer au dictionnaire.

Exemple

حَسْرَةَ مَوْتٍ { -----> 1 forme si le mot existe dans le dictionnaire.
 -----> 2 formes (حَسْرَةَ / مَوْتِ) si le mot n'existe pas.

b/ Homographies actuelles

Nous appelons "homographies" toutes les formes qui sont épelées de la même manière mais qui appartiennent à plus d'un lemme.

Appliquée à l'informatique, l'homographie est une graphie identique avec des lemmes différents.

Exemple

Nom

فَدَان

Verbe

دَهَبَ عَلَيَّ إِلَى

إِسْم

فِعْل

En général dans le discours, il n'y a pas de formes homographes, il y a des "distorsions" du langage.

Ce cas est appliqué aussi aux "stylèmes" (effets de style), mais en principe, il n'y a pas de mots ambigus dans la phrase.

Il y a plusieurs niveaux et plusieurs sources de cause d'homographie:

- Homographie entre nom propre et nom commun

exemple :

Amīn

أَمِين

En général, tous les noms propres en arabe sont des noms communs

- Parfois il y a une homographie entre les différentes valeurs logiques du même mot à l'intérieur des phrases différentes.

MOT 1 - est fonction substantivée dans phrase 1

MOT 2 - est fonction adjectivale dans phrase 2

Nom

Adjectif

>

عَرَبِيٌّ

L'homographie on la trouve dans les formes ; mais si on la trouve dans les lemmes (toutes les formes en seraient homographes), on doit distinguer le (les) lemme(s) graphiquement identique avec un code numérique différent ou bien avec un autre code.

Dans un texte arabe les formes homographes seront beaucoup plus nombreuses si le texte n'est pas vocalisé.

L'identité des consonnes radicales ne suffit pas encore à reconnaître les formes comme homographes.

c/ Homographies possibles

Parmi les formes récapitulées d'un texte, il y a celles qui sont des homographies actuelles et des autres qui sont des homographies possibles.

Les premières existent uniquement dans un seul texte (même mot à plusieurs lemmes différents); par contre pour les deuxièmes, elles ne s'adressent qu'à plusieurs textes et suivant le contexte (le mot 1 a un sens unique dans un texte 1, mais le même mot peut avoir un autre sens dans un texte 2).

Le problème est : comment reconnaître qu'une forme a l'état isolé ou hors contexte est une forme homographe possible ? Pour cela:

- Quand on lemmatise sur les formes, il faudrait toujours se demander si cette forme est homographe pour éviter les erreurs d'interprétations.
- En théorie, pour chaque forme il faudrait lire le contexte avant la lemmatisation (pour le texte coranique, le contexte se résume au verset).

III-4-4/ Traitement informatique de la lemmatisation

A partir des listes de lemmatisation obtenues dans la phase de récapitulation, il faudrait rajouter manuellement le lemme et les codes morphologiques de chaque forme dans les emplacements prévus pour cela.

La lemmatisation a été faite de la manière suivante :

Invariables : lemme = invariable sans aucune modification.

Noms : pour les noms propres, le lemme a été inscrit sans préfixe et sans désinence.

الجزائر ← جزائر

pour les formes dérivées, le lemme est la forme du nom au singulier (pour les adjectifs, seul le masculin a été représenté).

جميلة ← جميد
كتاب ← كتابان
كتب

- Verbes : toutes les formes sont représentées à la 3ème personne du singulier de l'imparfait.

ذهب ← ذهبوا
يتذهبون
ما ذهبنا

Remarque : Pour éviter l'homographie entre le lemme du verbe et celui du nom, on a supprimé la désinence casuelle du nom.

Codes morphologiques

Les codes morphologiques retenus sont :

- les Invariables

- . prépositions
- . conjonctions
- . adverbes irréductibles
- et autres ----- G

- Flexions nominales

- . noms propres ----- A
- . noms communs et dérivées ----- N

- Flexions verbales ----- V

- Homographie

- . oui ----- H
- . non ----- blanc

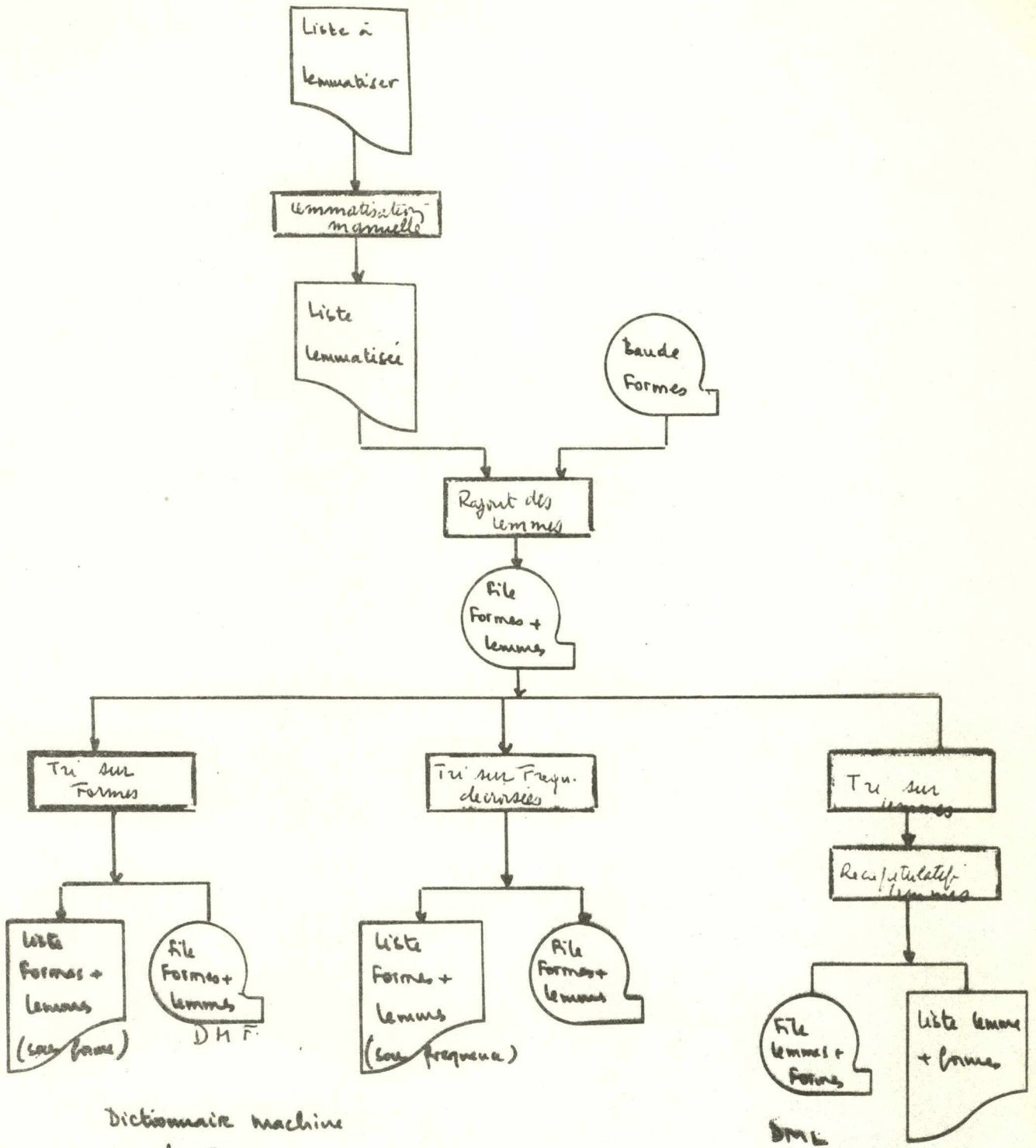
- Inconnu

- . Pour certaines formes se trouvant au début des
sourates, et qui n'ont pas pu être identifiées
----- X

Cette opération manuelle terminée, il faudra apporter tous les rajouts dans le fichier des formes à lemmatiser par l'intermédiaire de l'écran de visualisation (Visu)

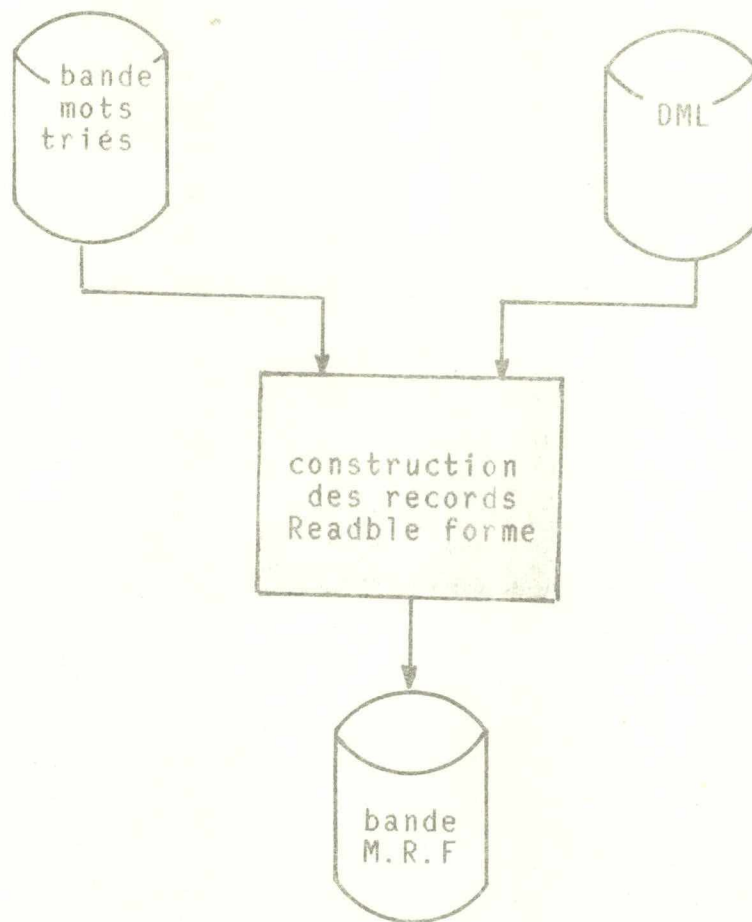
(cf: programme de lemmatisation en Annexe.)

Un tri sera alors effectué sur la position des lemmes afin d'obtenir une liste de tous les lemmes avec leurs formes.



Dictionnaire machine des Formes

DML
Dictionnaire machine des lemmes



mots, ponctuations, codes

Les listes composant le système lexical

Pour avoir un logiciel de traitement de texte complet il serait souhaitable de pouvoir générer des listes qui constitueront le système lexical, telles que :

- liste des lemmes
- liste des formes par ordre alphabétique
- liste des formes par ordre de fréquence décroissante
- liste des variantes graphiques
- liste des lemmes ordonnés contraire
- liste des formes ordonnées contraire
- liste des homographes
- quantités (toutes les statistiques lexicales).

Références			
# 5	0	3	
# 5	1	3	
# 5	3	3# 5	4 3
# 5	2	3# 5	6 3
# 5	2	3# 5	6 3
# 5	5	3# 5	4 3
# 5	6	3	
# 5	5	3	
# 5	3	3	
# 5	3	3	
# 5	6	3	
# 5	2	3	
# 5	6	3	
# 5	4	3	
# 5	1	3	
# 5	1	3	
# 5	5	3	
# 5	6	3	
# 5	4	3	
# 5	2	3	
# 5	3	3	
# 5	5	3	
# 5	4	3# 5	2 3
# 5	1	3# 5	4 3
# 5	4	3	
# 5	4	3	

mot	Fréquence	Longueur
آل	1 3	1
آل نام	1 6	2
آل نزال	2 8	3
آل ربه	2 4	4
آل ربه	2 6	5
آل ربه	2 4	6
آل ربه حاتم	1 10	7
آل ربه من	1 7	8
آل نجميل	1 10	9
آل الشوراه	1 11	10
آل الحكيم	1 9	11
آل الحية	1 6	12
آل القدي	1 6	13
آل القديس	1 8	14
آل الرهمن	1 10	15
آل الرحيم	1 9	16
آل القائم	1 9	17
آل المرير	1 9	18
آل المرهقان	1 11	19
آل العيشوم	1 9	20
آل العيشاب	1 9	21
آل الله	1 6	22
آل الله	2 6	23
آل الله	2 6	24
آل العرشان	1 10	25
آل العرشان	1 8	26

	# 5	3	3		بِالْحَقِّ	1	8	27	
	# 5	1	3		بِسْمِ	1	6	28	
	# 5	3	3		بِئْمَنَ	1	6	29	
	# 5	4	3		ذُو	1	3	30	
	# 5	0	3		سُورَةُ	1	7	31	
	# 5	4	3		شَدِيدًا	1	7	32	
	# 5	5	3		شَهِيءًا	1	5	33	
	# 5	4	3		عَذَابًا	1	7	34	
	# 5	4	3		عَذَابًا	1	7	35	
	# 5	3	3		عَلَيْكَ	1	8	36	
	# 5	5	3		عَلَيْكُمْ	1	8	37	
	# 5	0	3		عِمْرَانَ	1	9	38	
# 5	6	3	# 5	5	3	5	3	39	
	# 5	4	3		عَمَلًا	1	6	40	
	# 5	4	3		كَفَرًا	1	8	41	
	# 5	6	3		كَيْفًا	1	6	42	
# 5	5	3	# 5	5	3	5	3	43	
	# 5	4	3		لِلنَّاسِ	1	8	44	
	# 5	3	3		لِيَا	1	5	45	
	# 5	4	3		لَهُمْ	1	6	46	
	# 5	0	3		مَدَنِيَّةً	1	10	47	
	# 5	3	3		مُتَدَانًا	1	9	48	
	# 5	4	3		مِنْ	1	4	49	
	# 5	3	3		نَرًا	1	6	50	
	# 5	4	3		مُدَى	1	5	51	
# 5	2	3	# 5	6	3	5	6	3	52
					مَوْ	3	4		

# 5	4	3# 5	5	3# 5	3	3	و	5	2	53
				# 5	5	3	بَعَثَ	1	7	54
				# 5	3	3	بَدَأَ	1	8	55
				# 5	6	3	بَشَأَ	1	7	56
				# 5	6	3	بِصْوَرٍ	1	12	57

NOMBRE DE MOTS = 7.400000E+01

LISTE A LEMMATISER

Références					Code Morphologique	Lemme	Forme	Fréquence			
								Longueur			
8	22	4#	8	11	4	N	أَب	أَبَاؤُكُمْ	6	10	1
16	70	/#	15	148	6	N	أَب	أَبَاؤُنَا	7	9	2
24	109	11#	13	104	5	N	أَب	أَبَاؤُمُ	3	10	3
		#	2	133	2	N	أَب	أَبَائِكَ	1	8	4
		#	26	23	13	N	أَب	أَبَائِهِمْ	1	10	5
19	23	9#	3	200	2	N	أَب	أَبَاؤَكُمْ	2	10	6
13	104	5#	3	170	2	N	أَب	أَبَاؤَنَا	5	9	7
		#	24	38	12	N	أَب	أَبَاؤِي	1	7	8
		#	29	26	17	V	أَبَى	أَبَى	1	3	9
15	165	6#	12	48	5	V	أَبَى	أَبَاكُمْ	3	8	10
		#	20	75	9	V	أَبَى	أَبَانَا	1	6	11
23	63	11#	23	28	11	V	أَبَى	أَبَائِي	2	7	12
5	258	2#	4	251	2	V	أَبَى	أَبَاهُ	2	6	13
7	170	3#	8	180	3	V	أَبَى	أَبَائِهِمْ	6	8	14
18	190	7#	18	190	7	V	أَبَى	أَبَائِنَا	2	9	15
		#	24	31	12	V	أَبَى	أَبَاتُ	1	5	16
3	200	2#	3	201	2	V	أَبَى	أَبِينَا	5	6	17
2	83	2#	1	43	2	V	أَبَى	أَبَاؤَا	10	5	18
		#	25	54	12	V	أَبَى	أَبَائِي	1	7	19
		#	25	66	12	V	أَبَى	أَبَاؤُهُ	1	7	20
		#	9	25	4	V	أَبَى	أَبَاؤُنِي	1	8	21
		#	22	88	10	V	أَبَى	أَبَاتُ	1	7	22
		#	17	144	7	V	أَبَى	أَبَائِكَ	1	9	23
11	12	5#	4	233	2	V	أَبَى	أَبَائِهِمْ	3	9	24
8	19	4#	4	229	2	V	أَبَى	أَبَائِنَاؤُنِي	3	14	25
		#	18	189	/	V	أَبَى	أَبَائِنَا	1	10	26
		#	25	101	12	V	أَبَى	أَبَائِي	1	10	27

References	Lemme Secondaire 1	Code morphologique	Lemme Principal	Formes			
	#15 134 6	N	آت	آت	1	5	1
	#27 85 15	J	---	آت	1	9	2
		V	آت	آت	2		
	#29 26 17			آت	1	3	1
15 165	6#12 48 5			آتكم	3	8	2
	#20 75 9			آتانا	1	6	3
23 63	11#23 28 11			آتاني	2	7	4
5 258	2# 4 251 2			آتاه	2	6	5
7 170	3# 8 180 3			آتام	6	8	6
18 190	7#18 190 7			آتاما	2	9	7
	#24 31 12			آتت	1	5	8
3 200	2# 3 201 2			آتنا	5	6	9
2 83	2# 1 43 2			آتوا	10	5	10
	#25 54 12			آتوني	1	7	11
	#25 66 12			آتوه	1	7	12
	# 9 25 4			آتومن	1	8	13
	#22 88 10			آتنت	1	7	14
	#17 144 7			آتنتك	1	9	15
11 12	5# 4 233 2			آتنتم	3	9	16
8 19	4# 4 229 2			آتنتمومن	3	14	17
	#18 189 7			آتنتنا	1	10	18
	#25 101 12			آتنتلي	1	10	19

Fichier des Lemmes

INDEX-CONTRAIRE

*FRQ*LG*N-REC

آل	1 3	1
الأم	1 6	2
الأنزل	2 8	3
الآيات	2 4	4
الآية	2 6	5
الآيات	2 4	6
الآيات الحام	1 10	7
الآيات من	1 7	8
الآيات جميل	1 10	9
الآيات راحة	1 11	10
الحكيم	1 9	11
الحق	1 6	12
التدري	1 6	13
التدريس	1 8	14
الرحمن	1 10	15
الرحيم	1 9	16
السماء	1 9	17
المرير	1 9	18
الفرقان	1 11	19
الغيثوم	1 9	20
الكتاب	1 9	21
الله	1 6	22
الله	2 6	23
الله	2 6	24
السماء	1 10	25
بيانات	1 8	26
بالحق	1 8	27

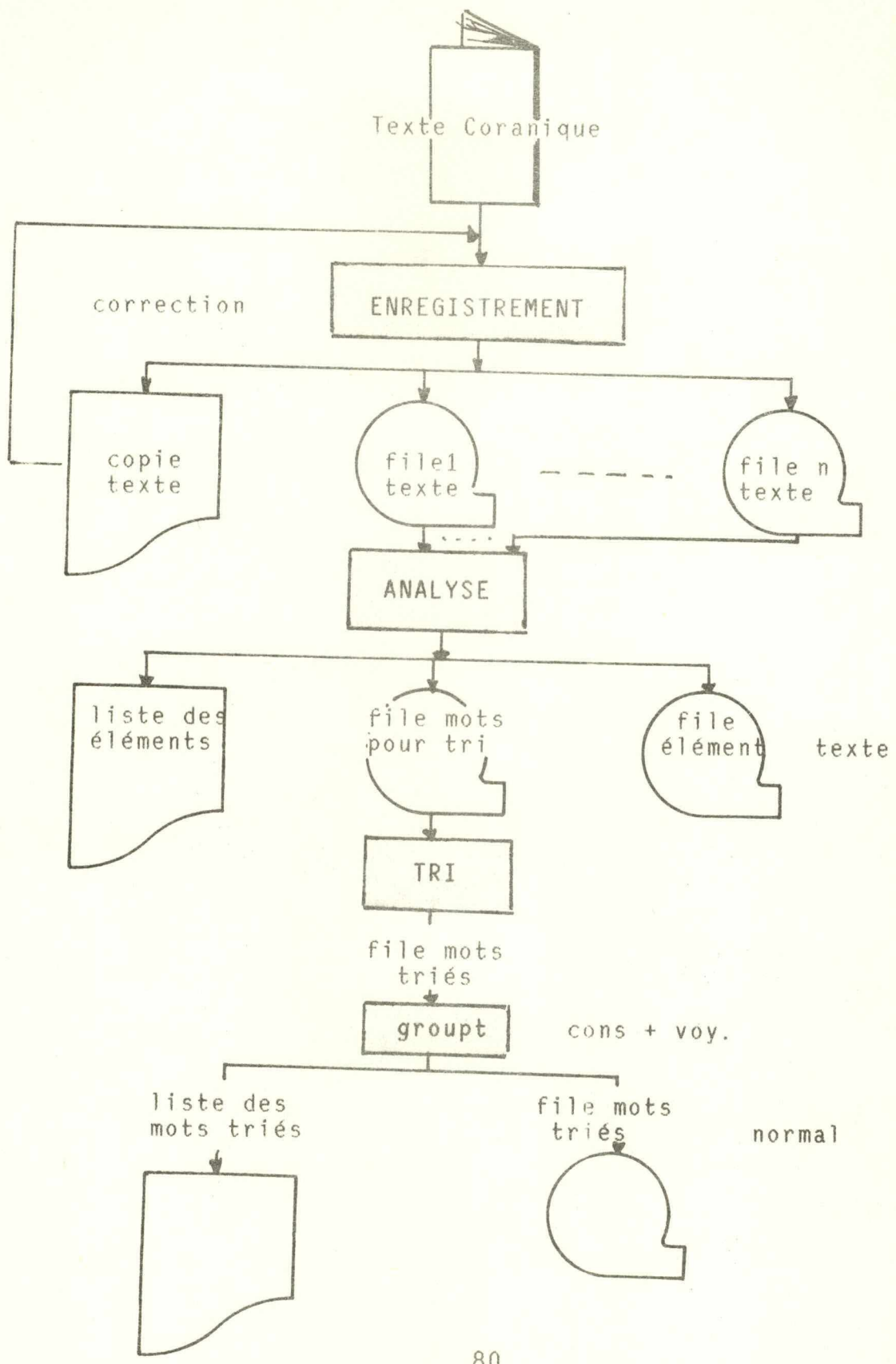
بِسْمِ	1 6	28
بِئْتِنَ	1 6	29
زُ و	1 3	30
سُورَةَ	1 7	31
سُدْرِ بَدِّ	1 7	32
شَيْءٍ	1 5	33
عَنْ أَبِي	1 7	34
عَزْرِي	1 7	35
عَلَيْكَ	1 8	36
عَلَيْهِ	1 8	37
عَمْرَانِ	1 9	38
فِي	3 3	39
قَبْلُ	1 6	40
كَفَرُوا	1 8	41
كَيْفَ	1 6	42
لَا	4 2	43
لِيُنْقِصَ	1 8	44
لِيَمَّا	1 5	45
لَهُمْ	1 6	46
مَدَّ نَيْتَهُ	1 10	47
مُمَدِّقًا	1 9	48
مِنْ	1 4	49
نَرَقَلْ	1 6	50
مُدَّي	1 5	51
مَوْ	3 4	52
وُ	5 2	53
بِغَفَى	1 7	54

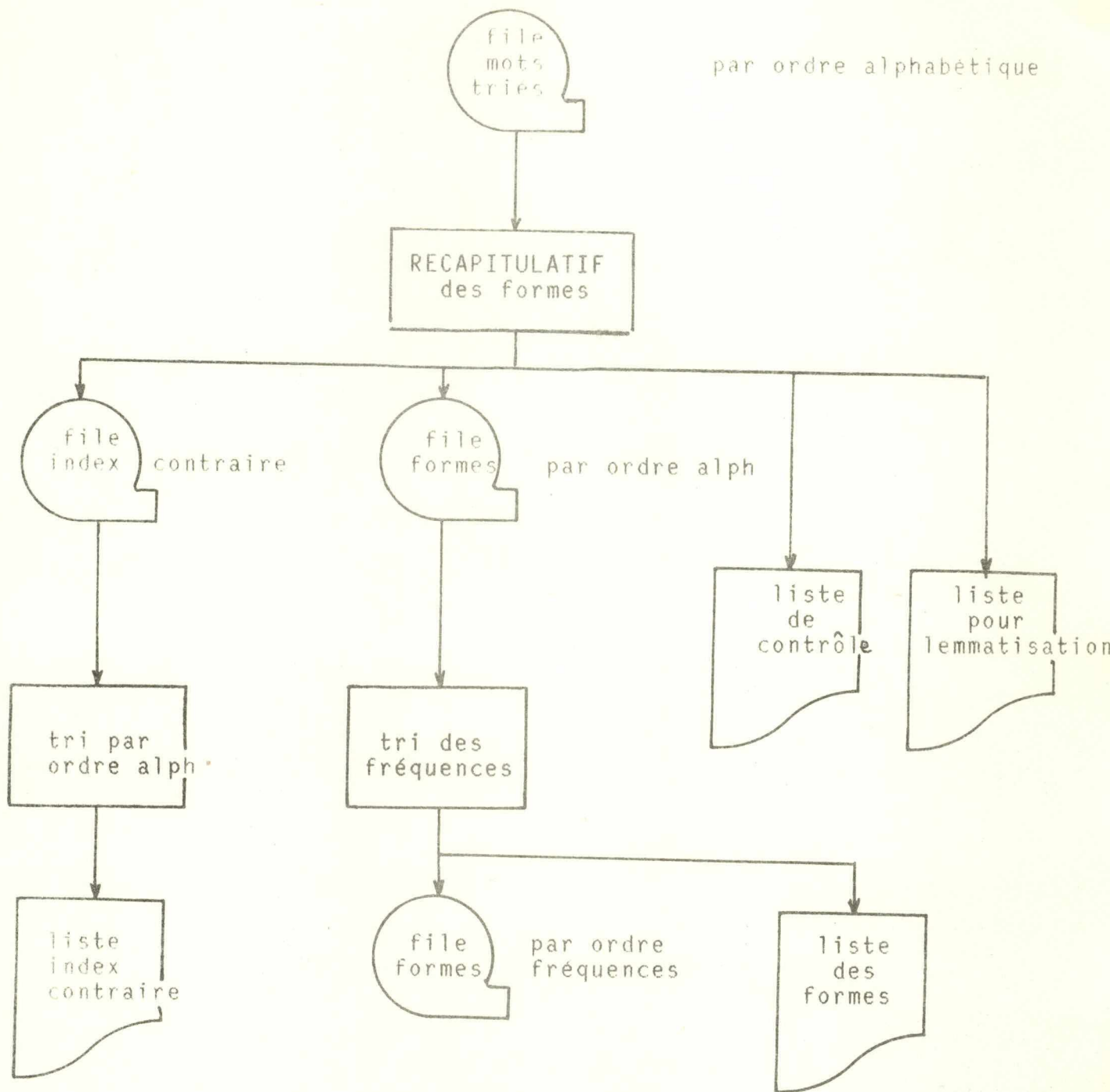


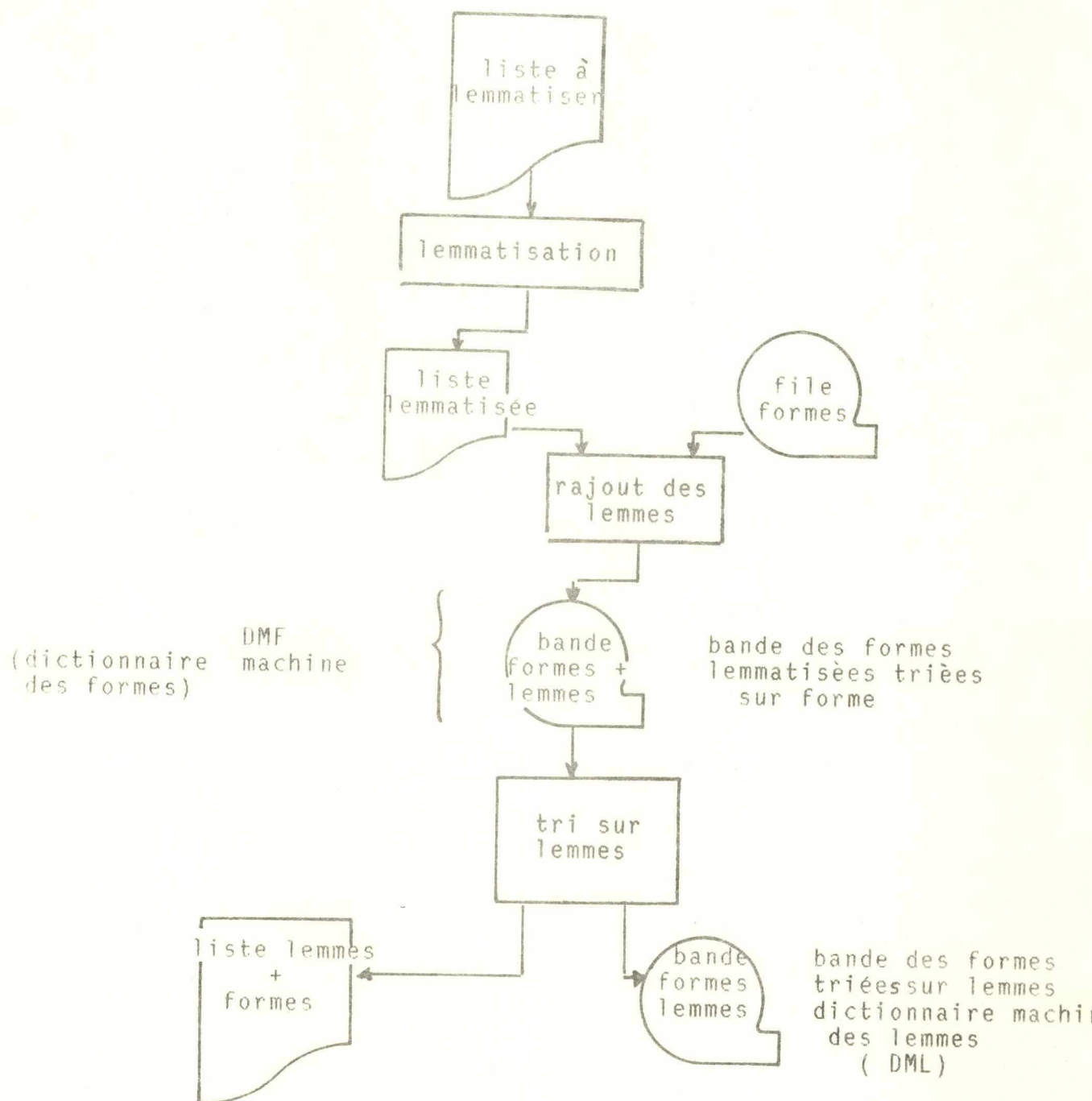
بند بنه	1 8	55
بشاور	1 7	56
بمبوتو (بم)	1 12	57

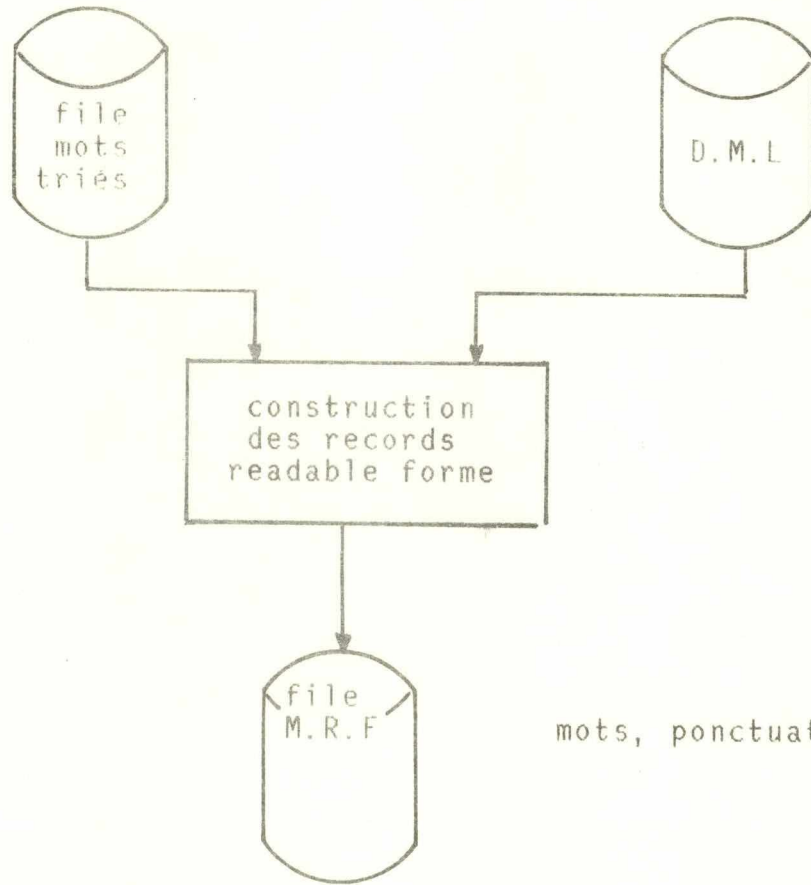
NOMBRE DE MOTS = 7.400000E+01

R E C A P I T U L A T I F
-----o-----









mots, ponctuations

IV/ EBAUCHE D'UN TRAITEMENT DE TEXTE EN TEMPS REEL

Pour éviter l'édition de gros volumes de concordances par exemple, ce qui demanderait des coûts très élevés, on est amenés aujourd'hui à réaliser sur ordinateur des banques de données qui pourront être consultées par des philologues en temps réel pour plusieurs usages, tels que :

- linguistique (Etude des langues, traduction automatique, étude des thésaurus,..)
- recherche de l'information (économique, politique, etc..)

Dans notre cas, ces bases de données vont permettre l'étude des thésaurus qui consiste, à partir du lemme ou de la forme, retrouver le mot dans son contexte.

Avant d'aborder le principe de la recherche automatique, il est indispensable de connaître comment vont être enregistrer les données (le texte) dans la mémoire externe (disques, bandes, etc) et de ce fait déterminer la méthode de consultation correspondante.

La façon d'enregistrer un texte, dépend du langage de programmation utilisé (puissance du compilateur), ainsi que du type de mémoires disponibles.

En général, il existe trois types d'organisations, qui sont :

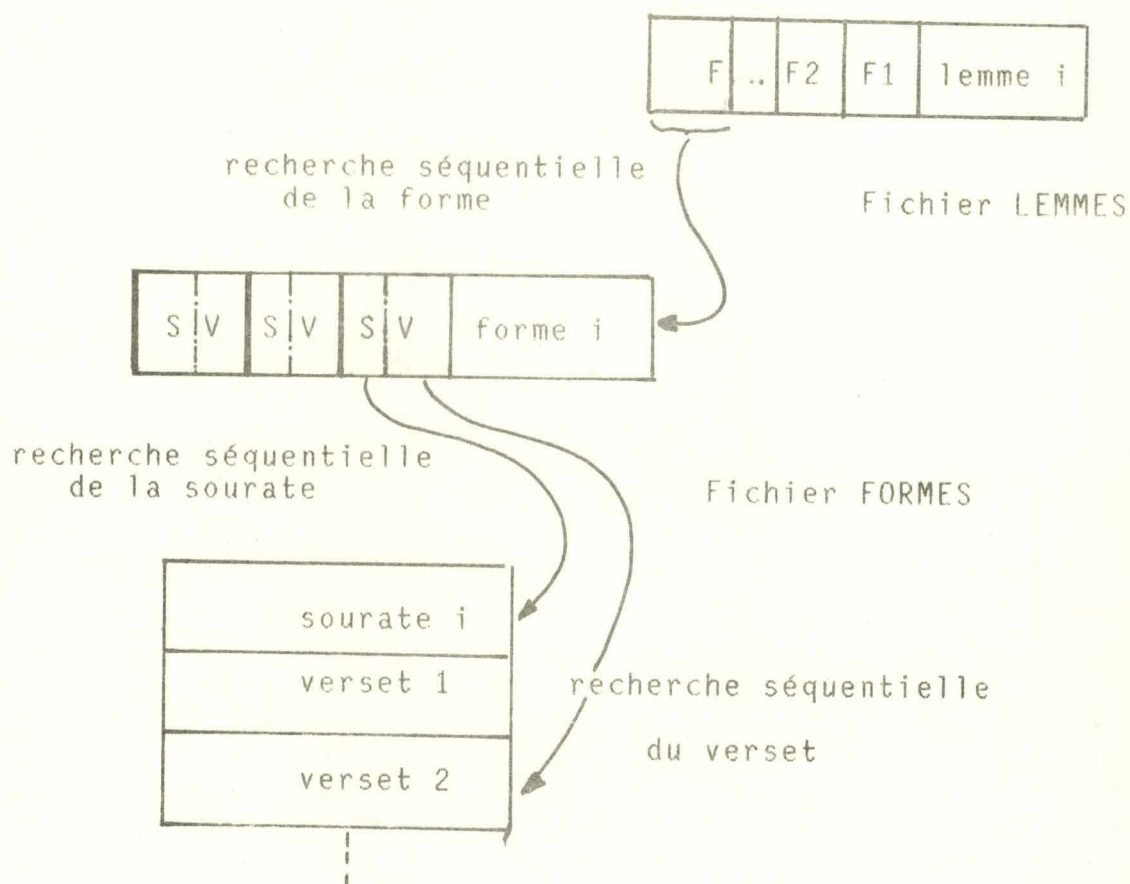
- Organisation séquentielle
- Organisation directe
- Organisation séquentielle indexée.

Pour chaque organisation, nous allons donner la structure des enregistrements, afin de retrouver le contexte d'un mot.

IV-1/ Principe de la recherche automatique

Organisation séquentielle

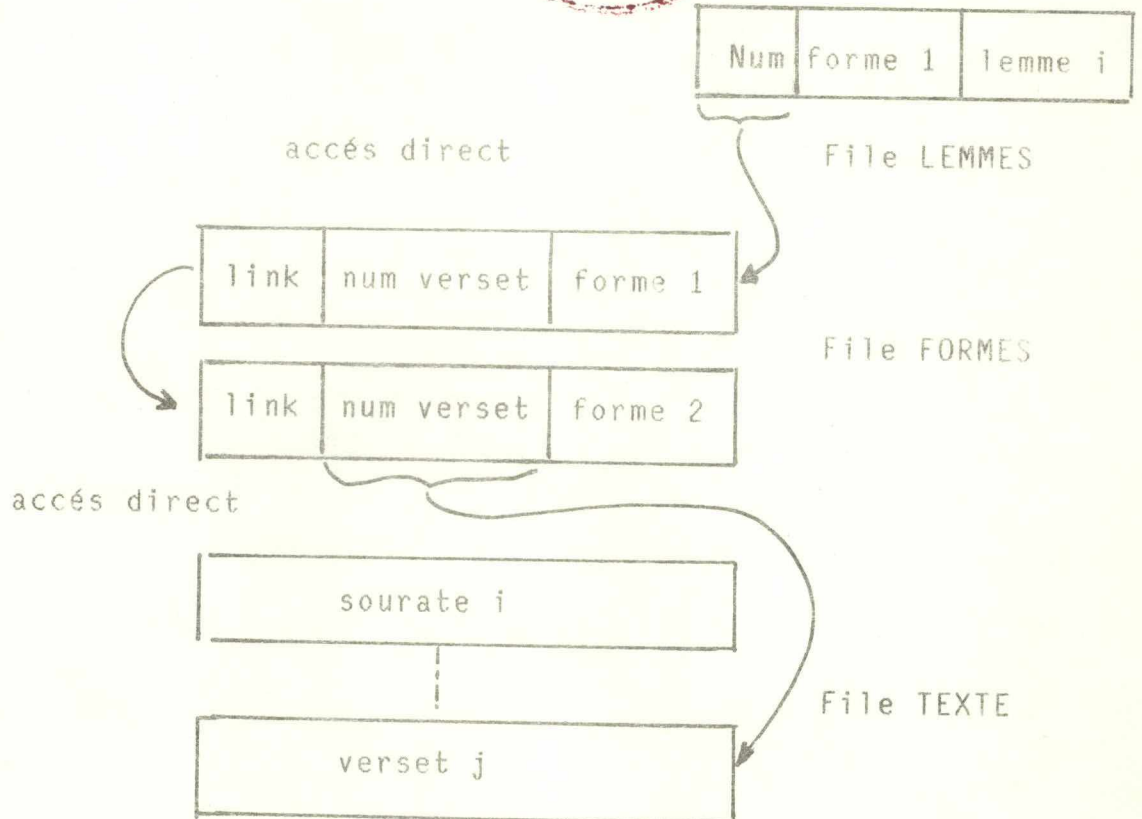
Les sourates composant le texte coranique, seront enregistrées les unes après les autres sur le disque. La consultation des versets est réalisée selon l'ordre physique; c'est à dire, pour consulter le nième verset, la lecture des n-1 versets qui le précèdent est obligatoire.



C'est une organisation coûteuse en temps de traitement. La consultation des versets est très lente (le temps de réponse, augmente avec la quantité de données enregistrées).

Organisation directe

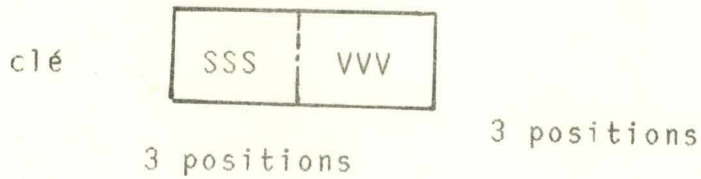
A chaque record, le système lui associera un numéro d'enregistrement, qui sera transparent à l'utilisateur, et permettra de retrouver d'une façon directe, le contexte du mot. La référence sourate, verset, et hizh permettra la recherche du mot uniquement dans le texte coranique.



Cette organisation permet une consultation rapide de contexte. Elle sera utilisée dans le cas où l'organisation indexée n'est pas permise.

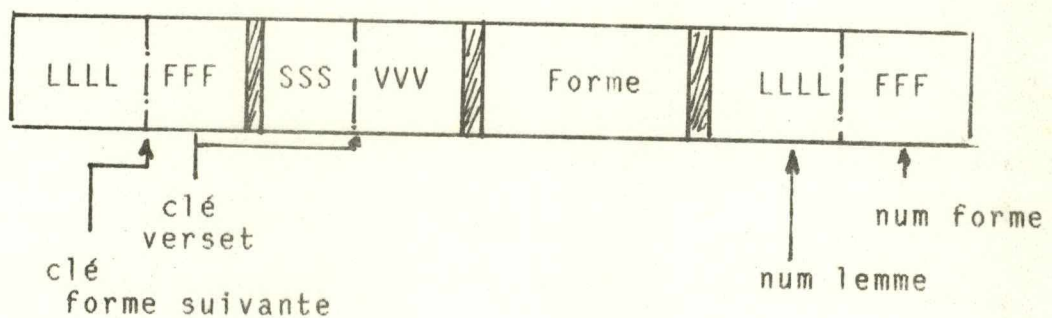
Organisation indexée

Elle permet d'accéder à un enregistrement le plus directement possible, sans avoir ni à balayer tout le fichier comme dans le cas d'une organisation séquentielle, ni à gérer les chainages (links) comme dans le cas d'une organisation directe. Chaque enregistrement est repéré par un indicatif ou clé. Pour le fichier sourate, la clé sera composée du numéro de la sourate, et du numéro du verset.



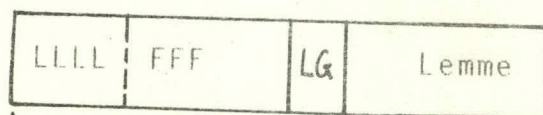
Le record forme comportera :

- la clé d'accès composée du numéro verset et du numéro de la forme
- la forme
- l'adresse de la forme suivante, appartenant au même lemme.

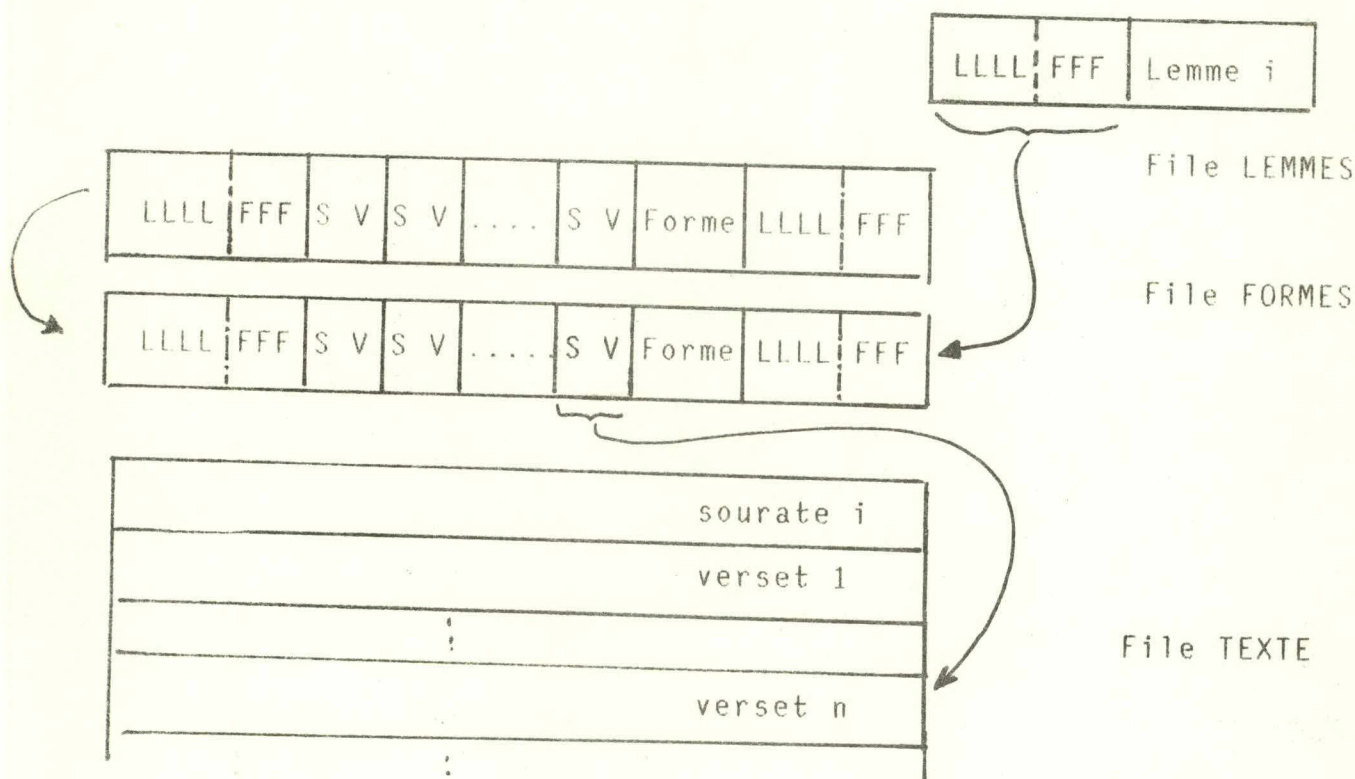


Le record lemme comportera :

- le lemme
- la clé de la première forme
- le nombre de formes appartenant à ce lemme



référence de la première forme



La gestion des clés est à la charge du système, ce qui permet une consultation rapide et sans erreurs.

Cette organisation nous permet de rechercher d'une manière simple et rapide, à l'aide d'un menu, le contexte d'un mot à partir soit du lemme principal, soit du lemme secondaire 1 ou 2, soit enfin de la forme.

V- Conclusion.

Par ce travail, nous avons voulu d'une part montrer qu'il était possible d'utiliser l'ordinateur pour le traitement de textes Arabes sans faire de transcodage (avec entrée et sortie en Arabe) ; et d'autre part évaluer l'importance de l'outil informatique pour les lexicologues, par la réalisation d'un dictionnaire machine du texte coranique.

Cette recherche a été surtout limitée par l'outil informatique, en effet, l'enregistrement et le traitement ont été faits sur un micro-ordinateur de 4 postes de 64 k octets de mémoire centrale.

Il a fallu subdiviser notre texte (30 premiers Hizbs du texte coranique) de 50.000 mots en sous - textes, afin de pouvoir les traiter par cette machine.

Le temps de traitement d'un sous - texte varie de 30 mn à 1 heure

Si la saisie a été facilitée par le système (Analyse contextuelle pour la génération de caractères arabes suivant leurs positions dans le mot), il n'en a pas été de même avec les imprimantes. Actuellement seules les imprimantes à aiguilles (Matricielles) ou à marguerite sont utilisées, car elles offrent une qualité d'impression de l'arabe vocalisé acceptable ; mais l'inconvénient reste le temps d'impression (80 à 160 caractères par seconde).

Pour donner une idée de cette lenteur, il a fallu près de 3h pour imprimer les listes formes (soit 9.000 formes) avec 4 imprimantes en parallèle.

Il serait utile de parler d'une manière succincte, du problème de l'évaluation du projet, ou d'une manière générale de l'utilisation de l'ordinateur en lexicologie.

La manière la plus simple pour déterminer ce coût, consiste à calculer le montant des frais, le temps et les efforts qu'on a dépensés pendant le procédé automatique (enregistrement, traitement, impression). Il serait intéressant de les comparer aux mêmes types de frais engagés en procédant à la recherche, à l'analyse, à la classification et à la description des données, le tout réalisé d'une manière traditionnelle (manuelle).

En ce qui concerne la portabilité du système, aucun problème ne se pose pour les programmes, car ils ont été écrits en langage évolué qui est le Pascal, dont le compilateur existe aujourd'hui sur tous les ordinateurs. Par contre pour les données, si le code des caractères arabes ne correspond pas à celui du système VT - 20, un transcodage sera alors nécessaire, car il n'existe pas actuellement un code standard regroupant tous les caractères arabes (voyelles et consonnes).

Pour conclure, il faudrait considérer cette étape comme le premier chaînon d'un travail plus complexe qui demanderait pour cela de gros moyens matériels (gros ordinateurs avec plusieurs postes de travail bilingues et surtout avec imprimantes bilingues rapides - photocomposeuse - etc...), ainsi qu'une équipe pluridisciplinaire regroupant des lexicologues, des informaticiens, et des philologues.

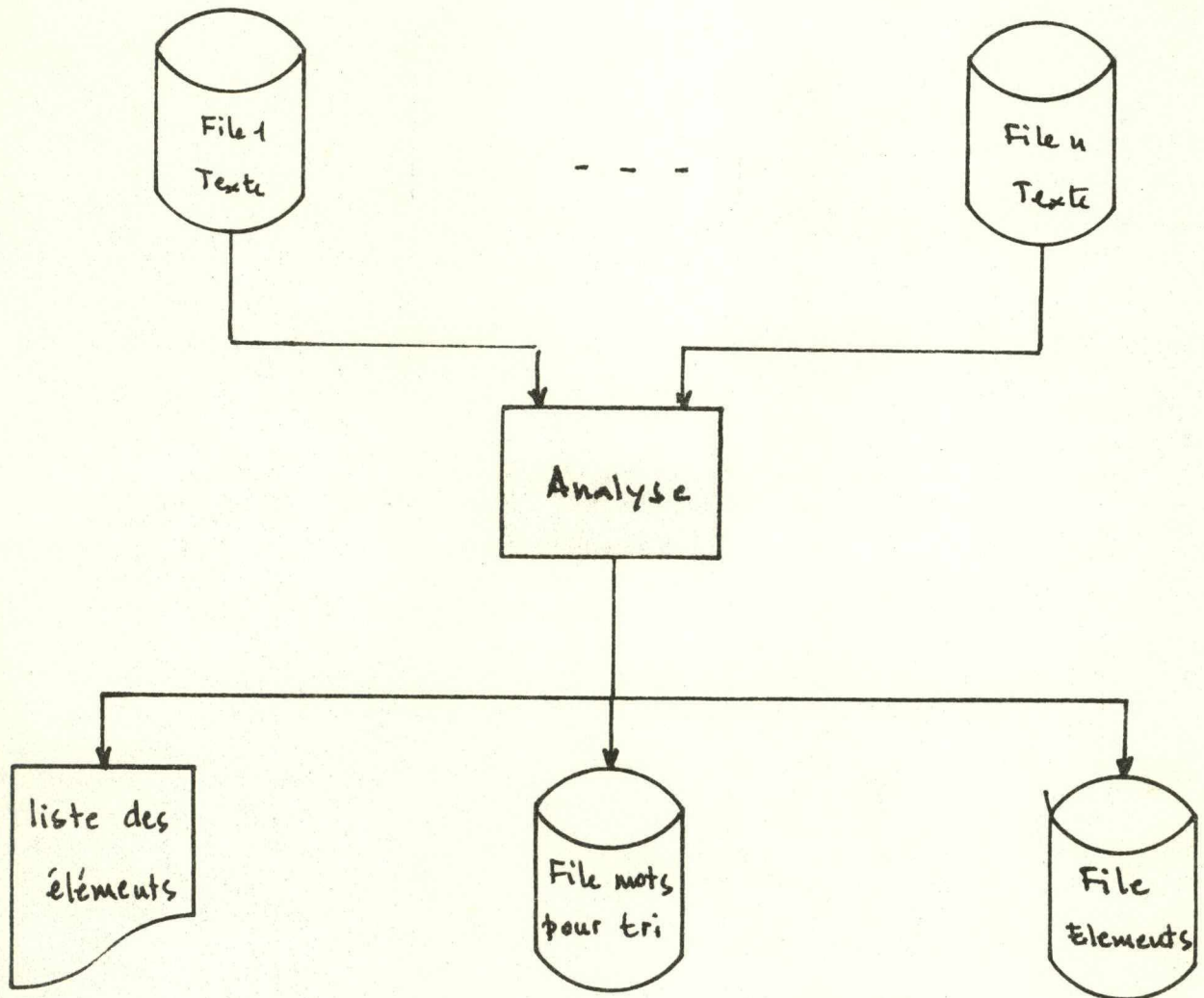
-----0-----

A - A N N E X E -



PROGRAMME D'ANALYSE





Analyse

L: File Elements
 M: File Output
 NR: num. record
 RM: nombre de records mots
 RP: nbre de records point
 RR: nbre de records rel.
 RI: records input
 NS: numero source
 NV: numero verset
 NH: numero Hitz
 MH: marque de Hitz

7

lecture d'un enreg. l2x6

EOF

Fin

RI = RI + 1

lecture d'un char

char = 'w'

traitement source

1

traitement verset

2

6

char = 'l'

char = 'p'

8

let 1 car

EOLN

char = '#'

traitement marque Hitz

3

char = '#'

traitement marque de souped

4

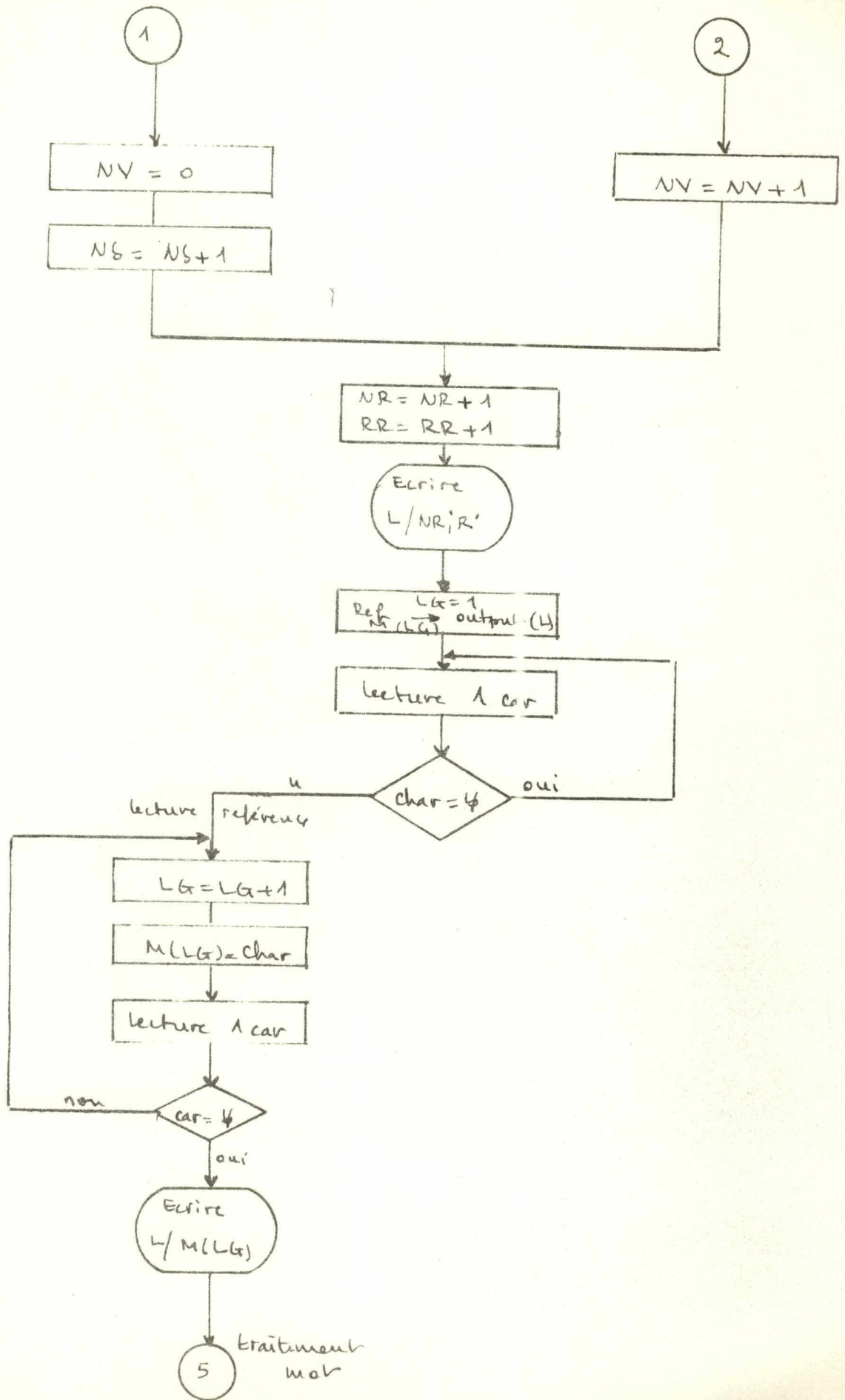
traitement mot

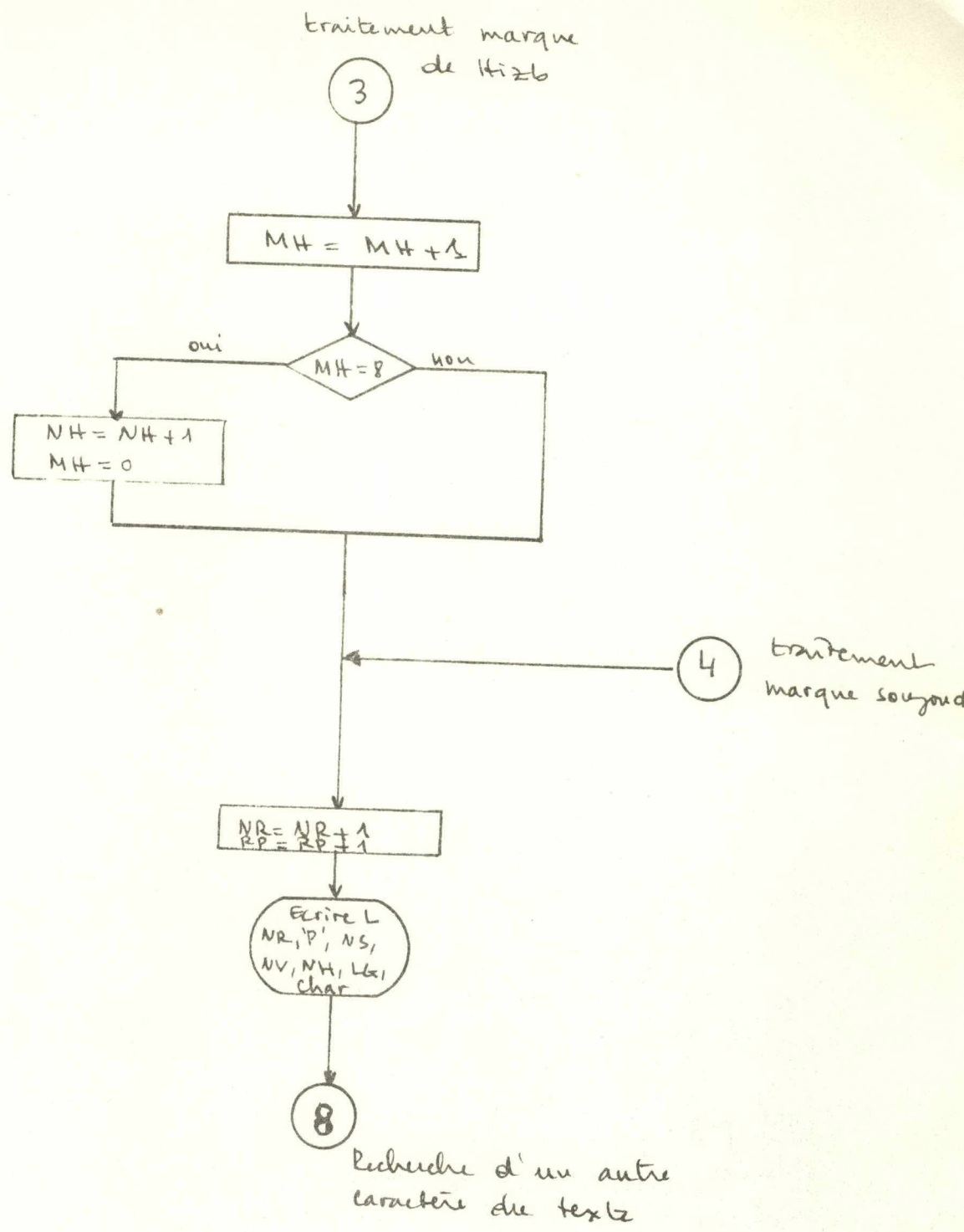
5

92-2

traitement source

Traitement verset





92-4

92-4

5 Traitement Mot

NR = NR + 1
RM = RM + 1

lire / L
NR, M, NS, NV, NH

lire / M
NS, NV, NH

L ← 0 lg mot
K1 ← 0 Index
J ← 0 conson/voyelle
S ← 0 double case

L ← L + 1
caract → M(L)

J = 0

lecture consonne

lecture voyelle

char = consonne

char = voy

S = 1

J = 1

char = 'j'

char = 'w'

Erreur Char

1: NR, K2
2: NT(K2)
3: NR, NV
S = 0

K1 = K1 + 1
NT(K1) = char

K1 = K1 + 1
NT(K1) = 'j'
K1 = K1 + 1
NT(K1) = 'j'
K1 = K1 + 1

S = 1

K2 = K2 + 1
NT(K2) = char

lect 1 car triple

EOLW

car = 'q'

write L, T
M(L),
NT(L)

write L, T
M(L),
NT(L)

7

6

92-5

```

LINE  P_LC LVL   PASCAL   2.0
  1      0  1)
  2      0  1)
  3      0  1)
  4     12  1) PROGRAM ANALYSE (F,L,T,T2,INPUT,OUTPUT);
  5     12  1) LABEL 2,10,20,25,30,40,50,55,56,60,100;
  6    172  1) VAR   BUF:   ARRAY[1..80] OF CHAR;
  7    224  1)       M:     ARRAY [1..26] OF CHAR;
  8    234  1)       MT:    ARRAY [1..30] OF CHAR;
  9    332  1)       FRJ:   ARRAY [1..49] OF INTEGER;
 10    334  1)       FRL:   INTEGER;
 11    384  1)       T,T2,L,F: TEXT;
 12    386  1)       CH:    CHAR;
 13    416  1)       K,II,I,K1,K2,J,S,ERR,LG,RR,RP,NS,NV,NH,MM: INTEGER;
 14    432  1)       X1,X2,NR1,NR2,RM1,RM2,RI1,RI2:   INTEGER;
 15    432  1)
 16      0  2) PROCEDURE ECR_MOT;
 17      0  2)
 18      2  2) VAR K:INTEGER;
 19      0  2) BEGIN
 20      6  2)   WRITE (L,LG:5);
 21     14  2)   FOR II:=1 TO 26 DO
 22     14  2)     BEGIN
 23     31  2)       IF M[II]<>CHR(123) THEN IF M[II]=CHR(161) THEN FRL:=FRL+1
 24     35  2)         ELSE
 25     35  2)           BEGIN
 26     43  2)             K:=ORD(M[II])-172;
 27     55  2)             FRQ[K]:=FRQ[K]+1;
 28     55  2)           END;
 29     64  2)       WRITE (L,M[II]);
 30     70  2)       M[II]:=CHR(123);
 31     78  2)     END;
 32     81  2)   WRITELN (L);
 33      2  2) END;
 34      2  2)
 35      0  2) PROCEDURE ECR_MTR;
 36      0  2)
 37      0  2) BEGIN
 38      6  2)   WRITE (T,LG:3);
 39     31  2)   FOR II:=1 TO 30 DO WRITE (T,MT[II]);
 40     34  2)   WRITELN (T);
 41      0  2) END;
 42      0  2) PROCEDURE INC_NR;
 43      0  2) BEGIN
 44      7  2)   IF NR2=999 THEN BEGIN NR2:=0;
 45     11  2)     NR1:=NR1+1;
 46     11  2)   END
 47     16  2)   ELSE NR2:=NR2+1;
 48      0  2) END;
 49      0  2) PROCEDURE ECR_NB;
 50      0  2) BEGIN
 51      6  2)   WRITE (L,CHR(123));
 52     15  2)   IF X1=0 THEN WRITE(L,X2:6)
 53     16  2)     ELSE BEGIN
 54     21  2)       WRITE (L,X1:3);
 55     36  2)       IF X2<10 THEN WRITE (L,CHR(48),CHR(48),X2:1
 56     41  2)         ELSE IF X2<100 THEN
 57     49  2)           WRITE (L,CHR(48),X2:2)
 58     50  2)           ELSE
 59     55  2)             WRITE (L,X2:3);

```

```

60      55  2)      END;
61      0   2)      END;
62      0   2)
63      0   2)      PROCEDURE ECR_T2;
64      0   2)      BEGIN
65      10  2)          IF X1=0 THEN WRITE(T2,X2:6)
66      11  2)              ELSE BEGIN
67      16  2)                  WRITE(T2,X1:3);
68      29  2)                  IF X2<10 THEN WRITE(T2,'00',X2:1)
69      34  2)                      ELSE IF X2<100 THEN
70      42  2)                          WRITE(T2,'0',X2:2)
71      48  2)                              ELSE WRITE(T2,X2:3);
72      48  2)      END;
73      0   2)      END;
74      0   1)      BEGIN
75      10  1)          RESET (F);
76      13  1)          REWRITE(T2);
77      16  1)          REWRITE (L);
78      19  1)          REWRITE (T);
79      22  1)          WRITELN (L);
80      28  1)          WRITE ('NUMERO DU RECORD = ');
81      34  1)          READ (NR1,NR2);
82      40  1)          WRITE ('NUMERO DE SOURATE = ');
83      44  1)          READ (NS);
84      48  1)          NS:=NS-1;
85      54  1)          WRITE ('NUMERO DU HIZB = ');
86      58  1)          READ (NH);
87      64  1)          WRITE ('MARQUE DE HIZB = ');
88      68  1)          READ (MH);
89      74  1)          WRITE ('RECORDS MOTS = ');
90      80  1)          READ (RM1,RM2);
91      86  1)          WRITE ('RECORDS REFERENCES = ');
92      90  1)          READ (RR);
93      96  1)          WRITE ('RECORDS PONCTUATIONS = ');
94      100 1)          READ (RP);
95      106 1)          WRITE ('RECORDS INPUT = ');
96      112 1)          READ (RI1,RI2);
97      124 1)          WRITE (L,CHR(32):39,'ELEMENT',CHR(32):11);
98      130 1)          WRITE (L,'LG*HH*VVV*SSS*T*N-REC');
99      133 1)          WRITELN (L);
100     136 1)          WRITELN (L);
101     158 1)          FOR II:=1 TO 26 DO M[II]:=CHR(128);
102     160 1)          ERR:=0;
103     162 1)          FRL:=0;
104     184 1)          FOR I:=1 TO 49 DO FRQ[I]:=0;
105     189 1)          WHILE NOT EOF (F) DO
106     189 1)              BEGIN
107     211 1)                  FOR II:=1 TO 80 DO BUF[II]:=CHR(128);
108     217 1)                  IF RI2=999 THEN BEGIN RI2:=0;
109     221 1)                      RI1:=RI1+1;
110     221 1)                      END
111     226 1)                      ELSE RI2:=RI2+1;
112     228 1)                  II:=0;
113     233 1)                  WHILE NOT EOLN (F) DO
114     233 1)                      BEGIN
115     237 1)                          II:=II+1;
116     241 1)                          IF II>80 THEN BEGIN
117     248 1)                              WRITELN (L,'ERREUR LG. ENRG > 80 ');
118     249 1)                              GO TO T00;
119     249 1)                              END;
120     256 1)                          REAG (F,BUF[II]);
121     257 1)                      END;

```

```

122 259 1) I:=1;
123 271 1) IF BUF[I]=CHR(14) THEN I:=I+1; (* ARABIC *)
124 280 1) IF BUF[I]=CHR(203) THEN GOTO 10; (* SOURATE *)
125 289 1) IF BUF[I]=CHR(191) THEN GOTO 20; (* VERSET *)
126 294 1) 2: IF I>90 THEN GOTO 60;
127 302 1) IF BUF[I]=CHR(128) THEN BEGIN
128 306 1) I:=I+1;
129 307 1) GOTO 2;
130 307 1) END;
131 316 1) IF BUF[I]=CHR(131) THEN GOTO 30; (* HIZB *)
132 325 1) IF BUF[I]=CHR(138) THEN GOTO 40; (* SOUJOD *)
133 326 1) GOTO 50; (* MOT *)
134 326 1)
135 326 1) (* TRAITEMENT SOURATE *)
136 326 1)
137 328 1) 10: NV:=0;
138 332 1) NS:=NS+1;
139 333 1) GOTO 25;
140 333 1)
141 333 1) (* TRAITEMENT VERSET *)
142 333 1)
143 337 1) 20: NV:=NV+1;
144 339 1) 25: INC_NR;
145 343 1) RR:=RR+1;
146 345 1) X1:=NR1;
147 347 1) X2:=NR2;
148 349 1) ECR_N3;
149 361 1) WRITE (L,CHR(123),R,CHR(123));
150 363 1) LG:=1;
151 373 1) M[LG]:=BUF[I];
152 377 1) LG:=LG+1;
153 381 1) I:=I+1;
154 394 1) WHILE BUF[I] = CHR(123) DO I:=I+1;
155 402 1) WHILE BUF[I] <> CHR(128) DO BEGIN
156 406 1) LG:=LG+1;
157 416 1) M[LG]:=BUF[I];
158 420 1) I:=I+1;
159 421 1) END;
160 423 1) ECR_MOT;
161 424 1) GOTO 2;
162 424 1)
163 424 1) (* TRAITEMENT MARQUE DE HIZB *)
164 424 1)
165 428 1) 30: MH:=MH+1;
166 436 1) IF MH=8 THEN BEGIN MH:=MH+1;
167 438 1) MH:=0;
168 438 1) END;
169 438 1)
170 438 1) (* TRAITEMENT MARQUE DE SOUJOD *)
171 438 1)
172 440 1) 40: INC_NR;
173 444 1) RP:=RP+1;
174 446 1) X1:=NR1;
175 448 1) X2:=NR2;
176 450 1) ECR_N3;
177 467 1) WRITE (L,CHR(123),CHR(80),CHR(128),NS:3,CHR(128));
178 487 1) WRITE (L,NV:3,CHR(123),MH:2,CHR(128),CHR(49):2,CHR(128));
179 496 1) WRITE (L,BUF[I]);
180 500 1) I:=I+1;
181 503 1) WRITELN (L);
182 504 1) GOTO 2;
183 504 1)

```


184 504 1)
 185 504 1)
 186 506 1)
 187 512 1)
 188 516 1)
 189 516 1)
 190 521 1)
 191 523 1)
 192 525 1)
 193 527 1)
 194 544 1)
 195 558 1)
 196 569 1)
 197 571 1)
 198 573 1)
 199 575 1)
 200 577 1)
 201 580 1)
 202 602 1)
 203 604 1)
 204 612 1)
 205 619 1)
 206 620 1)
 207 620 1)
 208 624 1)
 209 634 1)
 210 638 1)
 211 638 1)
 212 653 1)
 213 662 1)
 214 662 1)
 215 664 1)
 216 672 1)
 217 672 1)
 218 676 1)
 219 682 1)
 220 686 1)
 221 692 1)
 222 696 1)
 223 696 1)
 224 697 1)
 225 697 1)
 226 701 1)
 227 711 1)
 228 711 1)
 229 711 1)
 230 712 1)
 231 712 1)
 232 716 1)
 233 730 1)
 234 744 1)
 235 758 1)
 236 750 1)
 237 760 1)
 238 763 1)
 239 764 1)
 240 764 1)
 241 764 1)
 242 764 1)
 243 765 1)
 244 765 1)
 245 781 1)

```

(* TRAITEMENT MOT *)
50: INC_NR;
   IF RM2=999 THEN BEGIN RM2:=0;
                        RM1:=RM1+1;
                        END
                        ELSE RM2:=RM2+1;
   X1:=NR1;
   X2:=NR2;
   ECR_N3;
   WRITE (L,CHR(128),CHR(77),CHR(128),NS:3,CHR(128));
   WRITE (L,NV:3,CHR(128),NM:2,CHR(128));
   WRITE (T,NS:3,NV:4,NM:3);
   K1:=0;
   K2:=15;
   J:=0;
   S:=0;
   TRAP(255,CH);
   FOR II:=1 TO 30 DO MT[II]:=CHR(128);
   LG:=0;
55: IF (LG > 26) OR (K2 > 30) THEN BEGIN
                                WRITELN (L,"NOT TROP
                                GOTO 100;
                                END;
   LG:=LG+1;
   M[LG]:=BUF[I];
   IF J=0 THEN (* LECTURE D'UNE CONSONNE *)
     BEGIN
       IF ((ORD(BUF[I])>183) AND (ORD(BUF[I])<222))
         OR (ORD(BUF[I])=161) THEN
         BEGIN
           56: J:=1;
           IF ORD(BUF[I])=161 THEN
             BEGIN
               K1:=K1+1;
               MT[K1]:=CHR(214);
               K1:=K1+1;
               MT[K1]:=CHR(191);
               K2:=K2+1;
             END
           ELSE
             BEGIN
               K1:=K1+1;
               MT[K1]:=BUF[I];
             END
           END
         END
       ELSE
         BEGIN
           IF S=1 THEN BEGIN
             IF BUF[I]=CHR(181) THEN MT[K2]:=C
             IF BUF[I]=CHR(182) THEN MT[K2]:=C
             IF BUF[I]=CHR(183) THEN MT[K2]:=C
             S:=0;
           END
           ELSE
             BEGIN ERR:=1;
             GOTO 100;
           END
         END
       END
     END
   ELSE
     (* LECTURE D'UNE VOYELLE *)
     BEGIN
       IF (ORD(BUF[I])>172) AND (ORD(BUF[I])<18
  
```

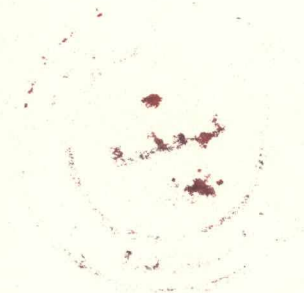
246 781 1)
 247 791 1)
 248 795 1)
 249 805 1)
 250 807 1)
 251 807 1)
 252 808 1)
 253 808 1)
 254 812 1)
 255 813 1)
 256 813 1)
 257 813 1)
 258 817 1)
 259 821 1)
 260 823 1)
 261 825 1)
 262 826 1)
 263 826 1)
 264 834 1)
 265 836 1)
 266 838 1)
 267 842 1)
 268 843 1)
 269 843 1)
 270 844 1)
 271 844 1)
 272 844 1)
 273 844 1)
 274 847 1)
 275 848 1)
 276 854 1)
 277 856 1)
 278 858 1)
 279 860 1)
 280 863 1)
 281 873 1)
 282 893 1)
 283 889 1)
 284 891 1)
 285 893 1)
 286 895 1)
 287 898 1)
 288 904 1)
 289 906 1)
 290 908 1)
 291 910 1)
 292 913 1)
 293 917 1)
 294 928 1)
 295 938 1)
 296 948 1)
 297 952 1)
 298 963 1)
 299 987 1)
 300 978 1)
 301 978 1)
 302 997 1)
 303 999 1)
 304 1007 1)
 305 1007 1)
 306 1026 1)
 307 1034 1)

```

BEGIN
  IF ORD(BUFC[I])=180 THEN S:=1;
  K2:=K2+1;
  MT[K2]:=BUFC[I];
  J:=0;
END
ELSE
BEGIN
  K2:=K2+1;
  GOTO 56;
END
END;
I:=I+1;
IF I > 80 THEN BEGIN
  ECR_MOT;
  ECR_MTR;
  GOTO 60;
END;
IF BUFC[I] = CHR(128) THEN BEGIN
  ECR_MOT;
  ECR_MTR;
  I:=I+1;
  GOTO 2;
END;
GOTO 55;
(* TRAITEMENT FIN DE LIGNE / TEXTE *)
60: READLN (F);
END;
WRITE (L, 'RECORDS MOTS           = ');
X1:=RM1;
X2:=RM2;
ECR_NB;
WRITELN (L);
WRITELN (L, 'RECORDS PONCTUATIONS = ', RP:6);
WRITELN (L, 'RECORDS REFERENCES   = ', RR:6);
WRITE (L, 'RECORDS INPUT           = ');
X1:=RI1;
X2:=RI2;
ECR_NB;
WRITELN (L);
WRITE (L, 'RECORDS OUTPUT           = ');
X1:=NR1;
X2:=NR2;
ECR_NB;
WRITELN (L);
X1:=RM1; X2:=RM2;
WRITE(T2, 'RECORDS MOTS           = '); ECR_T2; WRITELN(T2);
WRITELN(T2, 'RECORDS PONCTUATIONS = ', RP:6);
WRITELN(T2, 'RECORDS REFERENCES   = ', RR:6);
X1:=RI1; X2:=RI2;
WRITE(T2, 'RECORDS INPUT           = '); ECR_T2; WRITELN(T2);
X1:=NR1; X2:=NR2;
WRITE(T2, 'RECORDS OUTPUT           = '); ECR_T2; WRITELN(T2);
FOR I:=1 TO 3 DO WRITELN(T2);
K:=0;
FOR I:=1 TO 47 DO
BEGIN
  WRITELN(T2, CHR(I+172), 'FRQ[I]:6, ' = ');
END;
  
```

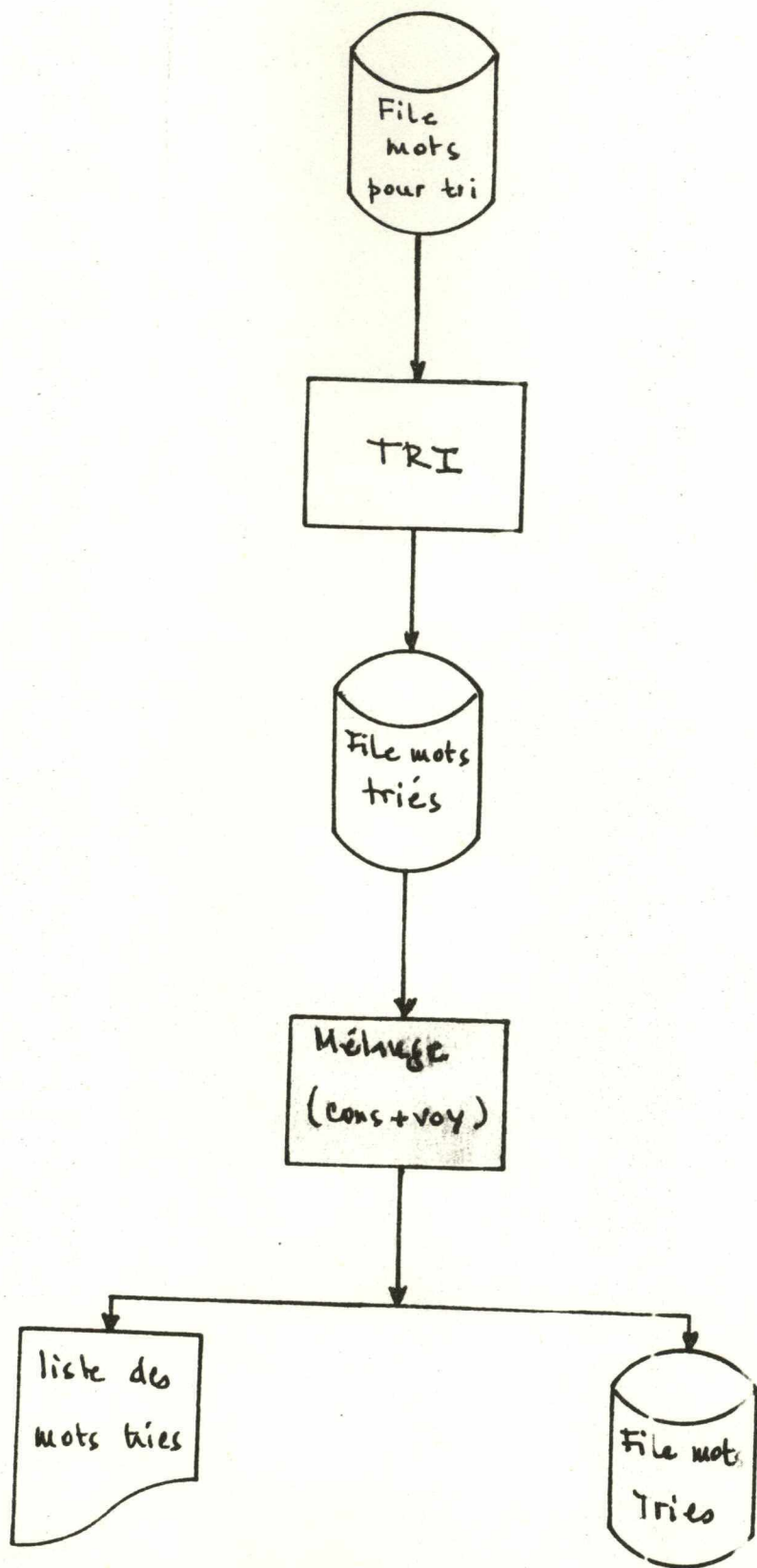
```
308 1051 1)          WRITELN(T2,CHR(221),FPC[49]:6," = ");
309 1054 1)          WRITELN(T2,CHR(161),FRL:6," = ");
310 1031 1)          10): IF ERR=1 THEN  WRITE (L,"ERREUR CAR. INCONNU ",BUF(I));
311  432 1) END.
```

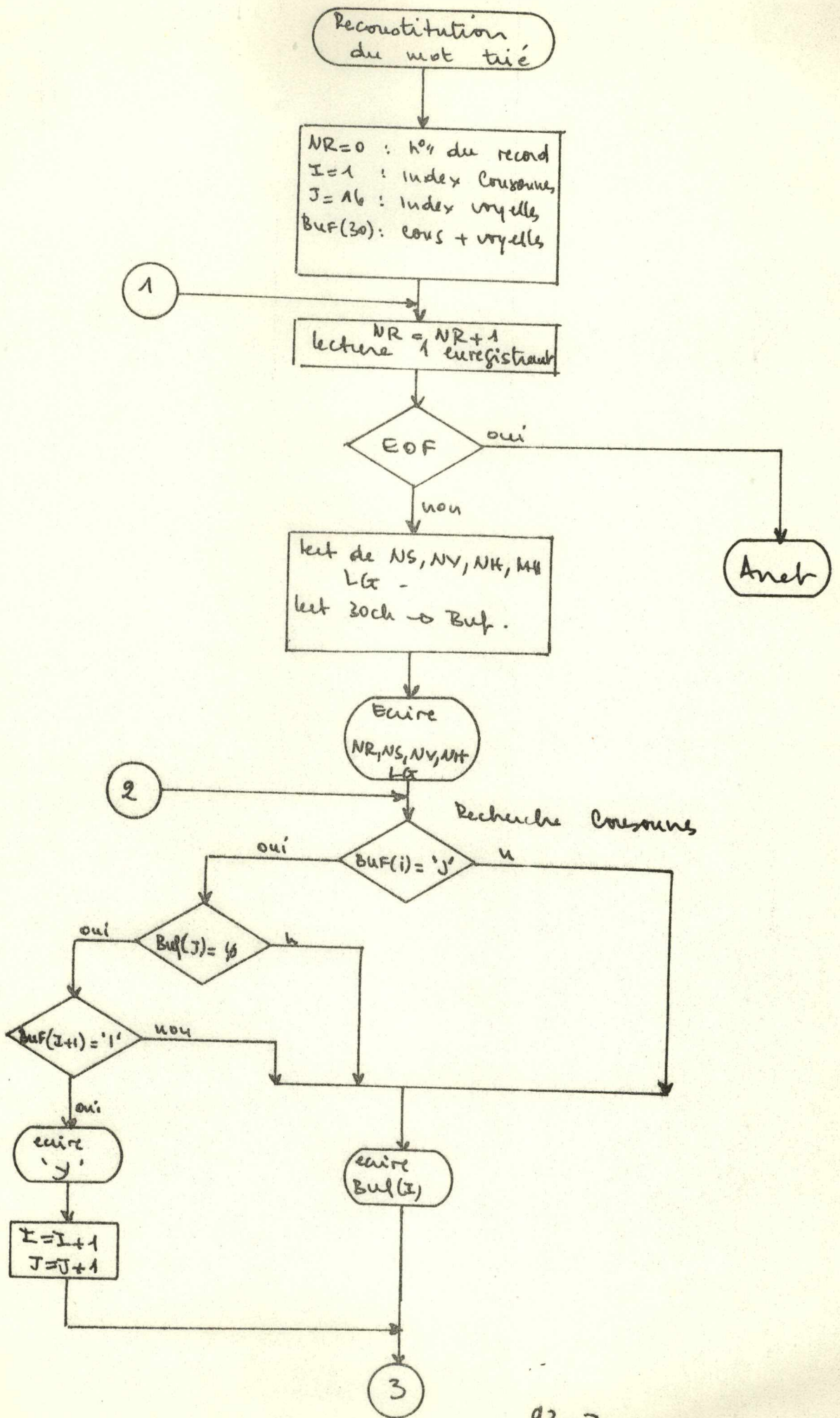
```
**** NO SYNTAX ERROR(S) DETECTED.
**** 311 LINE(S) READ, 5 PROCEDURE(S) COMPILED,
**** 1321 P-INSTRUCTIONS GENERATED.
```

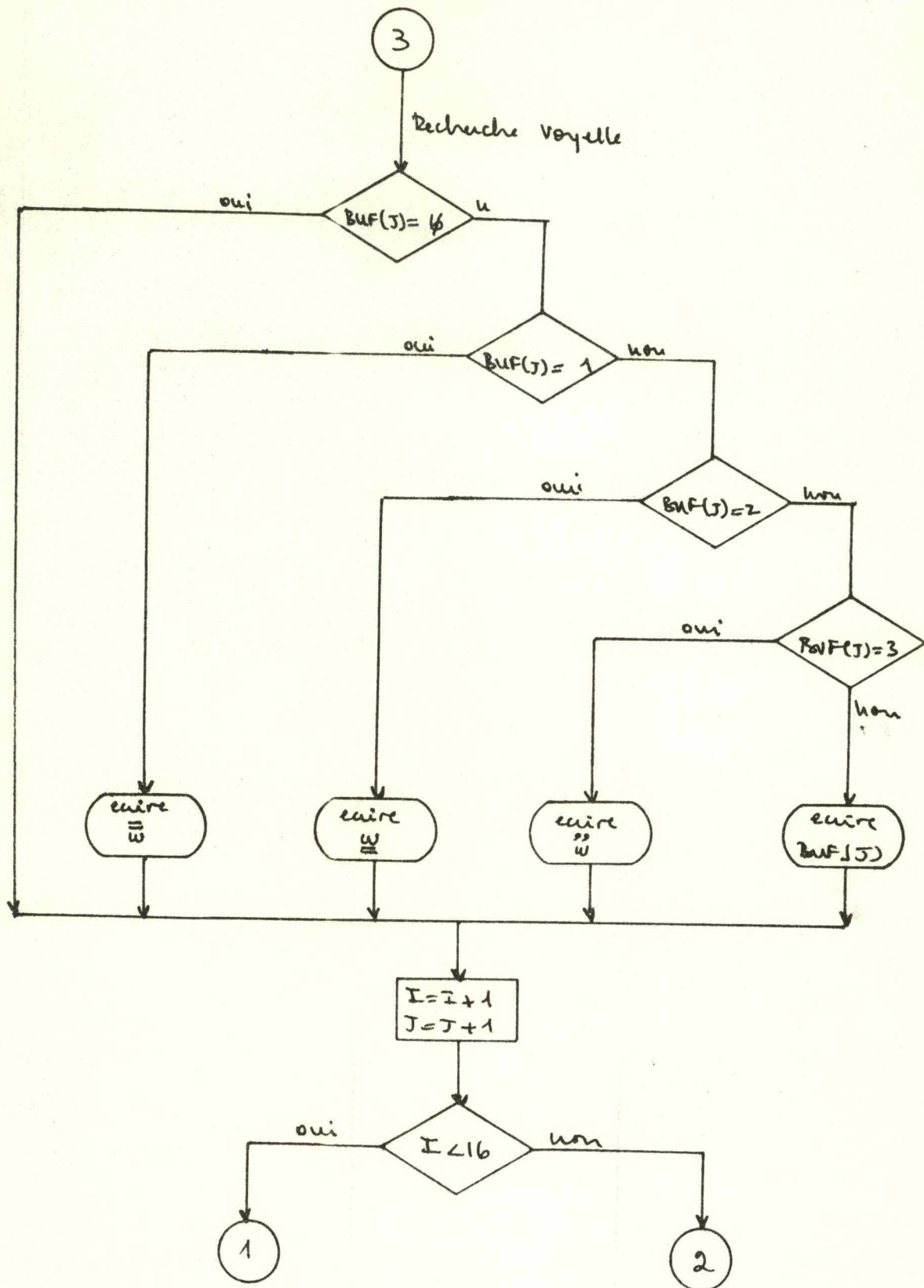


PROGRAMME MELANGE









PASCAL 2.0

```
PROGRAM MELANGE (F,T);
LABEL 10,20,30;
VAR   F,T:      TEXT;
      NR1,NR2,I,J:  INTEGER;
      BUF:      ARRAY [1..30] OF CHAR;
      CH:       CHAR;
      NS,NV,NH,LG: INTEGER;
BEGIN
  RESET (F);
  REWRITE (T);
  WRITE (T,CHR(32):41,"ELEMENT",CHR(32):11);
  WRITE (T,"LG*HH*VVV*SSS*N-REC");
  WRITELN (T);
  WRITELN (T);
  NR1:=0;
  NR2:=0;
  WHILE NOT EOF (F) DO
  BEGIN
    IF NR2=999 THEN BEGIN NR2:=0;
                      NR1:=NR1+1;
                    END
                ELSE NR2:=NR2+1;
    FOR I:=1 TO 30 DO BUF[I]:= CHR(128);
    TRAP (255,CH);
    WHILE NOT EOLN (F) DO
    BEGIN
      READ (F,NS,NV,NH,LG);
      FOR I:=1 TO 30 DO READ (F,BUF[I]);
    END;
    WRITE (T,CHR(128));
    IF NR2=0 THEN WRITE (T,NR2:6)
    ELSE BEGIN
      WRITE (T,NR1:3);
      IF NR2<10 THEN WRITE (T,CHR(48),CHR(49),NR2:1)
      ELSE IF NR2<100 THEN
        WRITE (T,CHR(48),NR2:2)
      ELSE
        WRITE (T,NR2:3);
    END;
    WRITE (T,CHR(128),NS:3,CHR(128),NV:3,CHR(128),NH:2,CHR(128),LG:5);
    I:=1;
    J:=16;
    WHILE I < 16 DO
    BEGIN
      IF BUF[I]=CHR(214) THEN BEGIN
        IF BUF[J]<>CHR(128) THEN GOTO 10;
        IF BUF[I+1]<>CHR(191) THEN GOTO 10;
        WRITE (T,CHR(161));
        I:=I+1;
        J:=J+1;
        GOTO 20;
      END;
      10: WRITE (T,BUF[I]);
      20: IF BUF[J]=CHR(128) THEN GOTO 30;
          IF BUF[J]=CHR(145) THEN BEGIN WRITE (T,CHR(180),CHR(181));
                                         GOTO 30;
          END;
          IF BUF[J]=CHR(146) THEN BEGIN WRITE (T,CHR(180),CHR(182));
                                         GOTO 30;
    END;
  END;

```

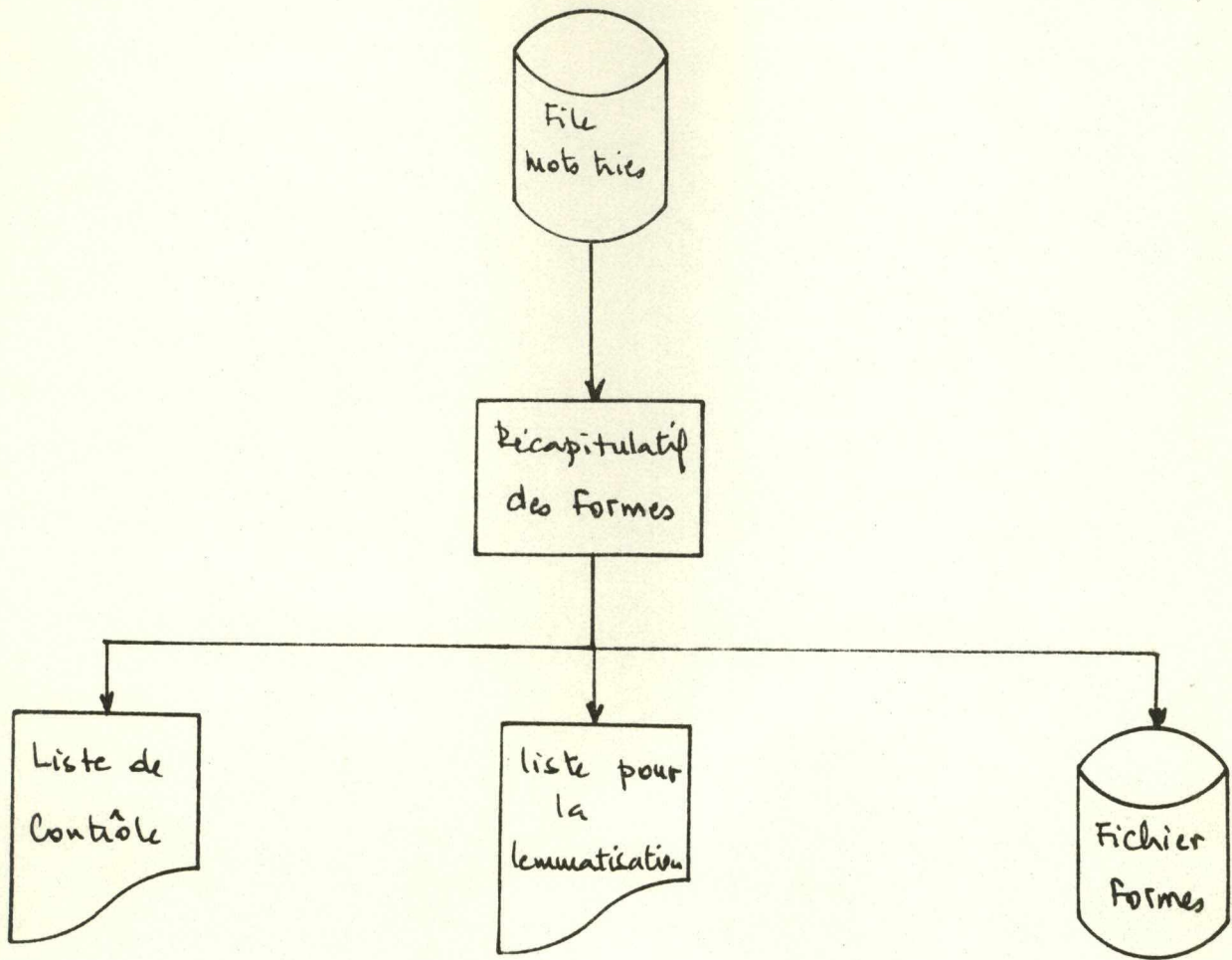


```
                                END;  
IF BUFCJJ=CHR(147) THEN BEGIN WRITE (T,CHR(180),CHR(183));  
                                GOTO 30;  
                                END;  
WRITE (T,BUFCJJ);  
30: I:=I+1;  
   J:=J+1;  
END;  
WRITELN (T);  
READLN (F);  
END;  
END.
```

TAX ERROR(S) DETECTED.
E(S) READ, 0 PROCEDURE(S) COMPILED,
NSTRUCTIONS GENERATED.



PROGRAMME DE PREPARATION
DES LISTES POUR
LA LEMMATISATION



INE P_LC LVL PASCAL 2.0

```
1      0 1)          (* RECAPITULATIF LEMMES *)
2      0 1)
3      6 1) PROGRAM REC_L (F,T2,OUTPUT);
4      6 1) LABEL 30,50,100;
5     222 1) VAR GRD,GRUF:          ARRAY [1..54] OF CHAR;
6     322 1)          REF:          ARRAY [1..50] OF CHAR;
7     324 1)          CH:           CHAR;
8     324 1)          F,T2:        TEXT;
9     340 1)          S,NR1,NR2,I,IR,FR,J,K:  INTEGER;
10    344 1)          FRS:         REAL;
11    344 1)
12    344 1)          (* PROCEDURE ECRITURE FICHER LEMME *)
13    344 1)
14     0 2) PROCEDURE ECR_LEM;
15     0 2) LABEL 10;
16     0 2) BEGIN
17     7 2)     IF NR2=999 THEN BEGIN NR2:=0;
18     11 2)         NR1:=NR1+1;
19     11 2)         END
20     16 2)         ELSE NR2:=NR2+1;
21     21 2)     WRITE (T2,CHR(128));
22     30 2)     IF NR1=0 THEN WRITE (T2,NR2:6)
23     31 2)         ELSE BEGIN
24     36 2)             WRITE (T2,NR1:3);
25     51 2)             IF NR2<10 THEN WRITE (T2,CHR(49),CHR(48),NR2:1)
26     56 2)                 ELSE IF NR2<100 THEN
27     64 2)                     WRITE (T2,CHR(48),NR2:2)
28     70 2)                         ELSE WRITE (T2,NR2:3);
29     70 2)         END;
30     95 2)     WRITE (T2,CHR(128),GRD[23],GRD[24],CHR(128),FR:4); (* LG ET FREQ *)
31    120 2)     FOR I:=25 TO 50 DO WRITE (T2,GRD[I]); (* FORME *)
32    120 2)     (* REFERENCES *)
33    145 2)     FOR I:=9 TO 18 DO WRITE (T2,GRD[I]); (* REF 1 *)
34    150 2)     WRITE (T2,CHR(131));
35    152 2)     I:=1;
36    160 2)     FOR J:=1 TO 2 DO
37    160 2)     BEGIN
38    169 2)         IF REF[I]=CHR(128) THEN GOTO 10;
39    177 2)         FOR K:=1 TO 10 DO BEGIN
40    186 2)             WRITE (T2,REF[I]);
41    190 2)             I:=I+1;
42    198 2)         END;
43    203 2)         WRITE (T2,CHR(131));
44    211 2)     END;
45    214 2)     10:WRITELN (T2);
46     0 2) END;
47     0 2)
48     0 1) BEGIN
49     7 1)     RESET (F);
50    10 1)     READLN (F);
51    13 1)     READLN (F);
52    16 1)     REWRITE (T2);
53    18 1)     NR1:=0;
54    20 1)     NR2:=0;
55    23 1)     FRS:=0;
56    25 1)     S:=0;
57    47 1)     FOR I:=1 TO 50 DO REF[I]:=CHR(128);
58    52 1)     WHILE NOT EOF (F) DO
59    52 1)     BEGIN
```

```

60      74  1)      FOR I:=1 TO 54 DO BUF[I]:=CHR(128);
61      76  1)      I:=0;
62      81  1)      WHILE NOT EOLN (F) DO
63      81  1)      BEGIN
64      85  1)          I:=I+1;
65      89  1)          IF I>54 THEN BEGIN
66      96  1)              WRITELN (T2,"** ERREUR ENRG INPUT **");
67      97  1)              GOTO 100;
68      97  1)          END;
69     104  1)          READ (F,BUF[I]);
70     105  1)      END;
71     108  1)      TRAP (255,CH);
72     112  1)      IF S=0 THEN BEGIN
73     138  1)          FOR I:=1 TO 54 DO GRD[I]:=BUF[I];
74     140  1)          S:=1;
75     142  1)          FR:=1;
76     144  1)          IR:=1;
77     145  1)          GOTO 50;
78     145  1)      END;
79     145  1)
80     145  1)      (* COMPARAISON ELEMENT *)
81     145  1)
82     174  1)      FOR I:=25 TO 54 DO IF BUF[I] <> GRD[I] THEN GOTO 30;
83     174  1)
84     174  1)      (* MEME ELEMENT *)
85     174  1)
86     178  1)      FR:=FR+1;
87     182  1)      IF FR < 7 THEN BEGIN
88     190  1)          FOR I:=9 TO 18 DO BEGIN
89     200  1)              REF[IR]:=BUF[I];
90     204  1)              IR:=IR+1;
91     212  1)          END;
92     212  1)
93     213  1)      GOTO 50;
94     213  1)
95     213  1)      (* MOTS DIFFERENTS *);
96     218  1)      30:FRS:=FRS+FR;
97     220  1)      ECR_LEM;
98     246  1)      FOR I:=1 TO 54 DO GRD[I]:=BUF[I];
99     268  1)      FOR I:=1 TO 50 DO REF[I]:=CHR(128);
100    270  1)      IR:=1;
101    272  1)      FR:=1;
102    275  1)      50:READLN (F);
103    276  1)      END;
104    278  1)      ECR_LEM;
105    283  1)      FRS:=FRS+FR;
106    286  1)      WRITELN (T2);
107    289  1)      WRITELN (T2);
108    299  1)      WRITELN (T2,"NOMBRE DE MOTS = ",FRS);
109    309  1)      WRITELN ("NOMBRE DE MOTS = ",FRS);
110    344  1) 100:END.

```

```

***      NO SYNTAX ERROR(S) DETECTED.
***      110 LINE(S) READ,      1 PROCEDURE(S) COMPILED,
***      525 P-INSTRUCTIONS GENERATED.

```

PROGRAMME D'INTRODUCTION
DES LEMMES EN TEMPS REEL

```

50 208 1) WHILE CH=CHR(128) DO READ(F,CH);
51 210 1) C:=CH;
52 210 1)
53 214 1) READ(F,CH);
54 223 1) WHILE CH=CHR(128) DO READ(F,CH);
55 223 1)
56 223 1) (* LECTURE REF *)
57 223 1) (* ***** *)
58 223 1)
59 245 1) FOR I:=1 TO 21 DO REF[I]:=CHR(128);
70 251 1) REF[1]:=CH;
71 253 1) I:=2;
72 258 1) WHILE NOT EOLN(F) DO
73 258 1) BEGIN
74 265 1)   READ(F,REF[I]);
75 270 1)   IF I=21 THEN GOTO 20;
76 274 1)   I:=I+1;
77 275 1) END;
78 275 1)
79 275 1) (* OUTPUT *)
80 275 1) (* ***** *)
81 275 1)
82 308 1) 20:WRITELN(T, LG, REF, LP, LS1, LS2, C, FRM, FRQ:5);
83 308 1)
84 312 1)   N:=N+1;
85 315 1)   TRAP(255,CH);
86 318 1)   READLN(F);
87 319 1) END;
88 329 1) WRITELN("NOMBRE DE RECORDS = ",N:5);
89 282 1) END.

```

```

**** NO SYNTAX ERROR(S) DETECTED.
**** 89 LINE(S) READ, 0 PROCEDURE(S) COMPILED,
**** 330 P-INSTRUCTIONS GENERATED.

```

*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **
*** */RLEM.L **

96-13

01 12

LINE	P	LC	LVL	PASCAL	Z.C
1		8	1)	PROGRAM RLEM(F,T,INPUT,OUTPUT);	
2		8	1)	VAR	
3		8	1)	F,T: TEXT;	
4		58	1)	FRM: ARRAY [1..25] OF CHAR;	
5		100	1)	REF: ARRAY [1..21] OF CHAR;	
6		160	1)	LP,LPS : ARRAY [1..15] OF CHAR;	
7		220	1)	CV : ARRAY [1..30] OF CHAR;	
8		280	1)	CON,VOY: ARRAY [1..15] OF CHAR;	
9		296	1)	LS1,LS2: ARRAY [1..4] OF CHAR;	
10		300	1)	LG : ARRAY [1..2] OF CHAR;	
11		306	1)	C,CS,CH: CHAR;	
12		316	1)	M,N1,N2,I,J: INTEGER;	
13		318	1)	FRQ : INTEGER;	
14		318	1)		
15		318	1)	(* S/P REGROUPEMENT CONS-VOY *)	
16		318	1)	(* ***** *)	
17		318	1)		
18		0	2)	PROCEDURE L_CV;	
19		0	2)	LABEL 10;	
20		6	2)	VAR I,J,K: INTEGER;	
21		0	2)	BEGIN	
22		23	2)	FOR I:=1 TO 30 DO CV[I]:=CHR(128);	
23		29	2)	I:=1; J:=1; K:=1;	
24		37	2)	WHILE CON[J]<>CHR(129) DO	
25		37	2)	BEGIN	
26		61	2)	IF (CON[J]=CHR(214)) AND (CON[J+1]=CHR(191)) AND (VOY[K]=CHR(161))	
27		63	2)	THEN BEGIN	
28		69	2)	CV[I]:=CHR(161);	
29		77	2)	J:=J+1; K:=K+1;	
30		77	2)	END	
31		88	2)	ELSE CV[I]:=CON[J];	
32		92	2)	I:=I+1;	
33		101	2)	IF VOY[K]=CHR(129) THEN GOTO 10;	
34		117	2)	IF (ORD(VOY[K])>=145) AND (ORD(VOY[K])<=147) THEN	
35		117	2)	BEGIN	
36		123	2)	CV[I]:=CHR(180);	
37		127	2)	I:=I+1;	
38		143	2)	CV[I]:=CHR(ORD(VOY[K])+36); I:=I+1;	
39		144	2)	GOTO 10;	
40		144	2)	END;	
41		158	2)	CV[I]:=VOY[K]; I:=I+1;	
42		166	2)	10:J:=J+1; K:=K+1;	
43		167	2)	END;	
44		6	2)	END;	
45		6	2)		
46		6	2)		
47		0	1)	BEGIN	
48		13	1)	RESET(F); REWRITE(T);	
49		19	1)	N:=0; N1:=0; N2:=0;	
50		24	1)	WHILE NOT EOF(F) DO	
51		24	1)	BEGIN	
52		49	1)	FOR I:=1 TO 15 DO CON[I]:=CHR(128); VOY:=CON;	
53		59	1)	READ(F, LG, FRQ, REF);	
54		82	1)	FOR I:=1 TO 10 DO READ(F, CON[I]);	
55		105	1)	FOR I:=1 TO 10 DO READ(F, VOY[I]);	
56		109	1)	READ(F, C);	
57		111	1)	L_CV;	
58		137	1)	FOR I:=1 TO 15 DO LP[I]:=CV[I];	
59		162	1)	FOR I:=1 TO 15 DO CON[I]:=CHR(128); VOY:=CON;	

```

60 176 1) READ(F,CON,VOY,LS1,LS2);
61 178 1) L_CV;
62 204 1) FOR I:=1 TO 25 DO FRM[I]:=CVC[I];
63 212 1) IF (LP<>LPS) OR (C<>CS) THEN
64 212 1) BEGIN
65 217 1) LPS:=LP; CS:=C;
66 223 1) N1:=N1+1; N2:=C;
67 244 1) FOR I:=1 TO 10 DO WRITE(T,CHR(128));
68 247 1) WRITELN(T);
69 271 1) FOR I:=1 TO 10 DO WRITE(T,CHR(128)); WRITELN(T);
70 292 1) FOR I:=1 TO 20 DO WRITE(T,CHR(128));
71 305 1) WRITELN(T,N1:5,LP,C);
72 326 1) FOR I:=1 TO 25 DO WRITE(T,CHR(128));
73 328 1) I:=1;
74 340 1) WHILE (LP[I]<>CHR(128)) AND (I<=15) DO
75 350 1) BEGIN WRITE(T,"-"); I:=I+1; END;
76 353 1) WRITELN(T);
77 377 1) FOR I:=1 TO 10 DO WRITE(CHR(128)); WRITELN(T);
78 377 1) END;
79 381 1) N2:=N2+1;
80 419 1) WRITELN(T,CHR(128),N2:5,CHR(128),LG,CHR(128),FRQ:5,FRM,LS1,LS2
81 423 1) N:=N+1;
82 426 1) READLN(F);
83 429 1) TRAP(255,CH);
84 430 1) END;
85 440 1) WRITELN("NOMBRE DE RECORDS = ",N:5);
86 318 1) END.

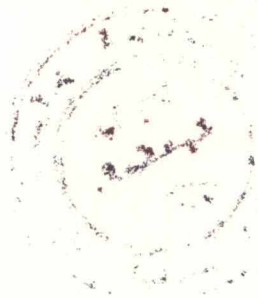
```

```

**** NO SYNTAX ERRGR(S) DETECTED.
**** 86 LINE(S) READ, 1 PROCEDURE(S) COMPILED,
**** 609 P-INSTRUCTIONS GENERATED.

```

*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **
*** */ORDN.L **

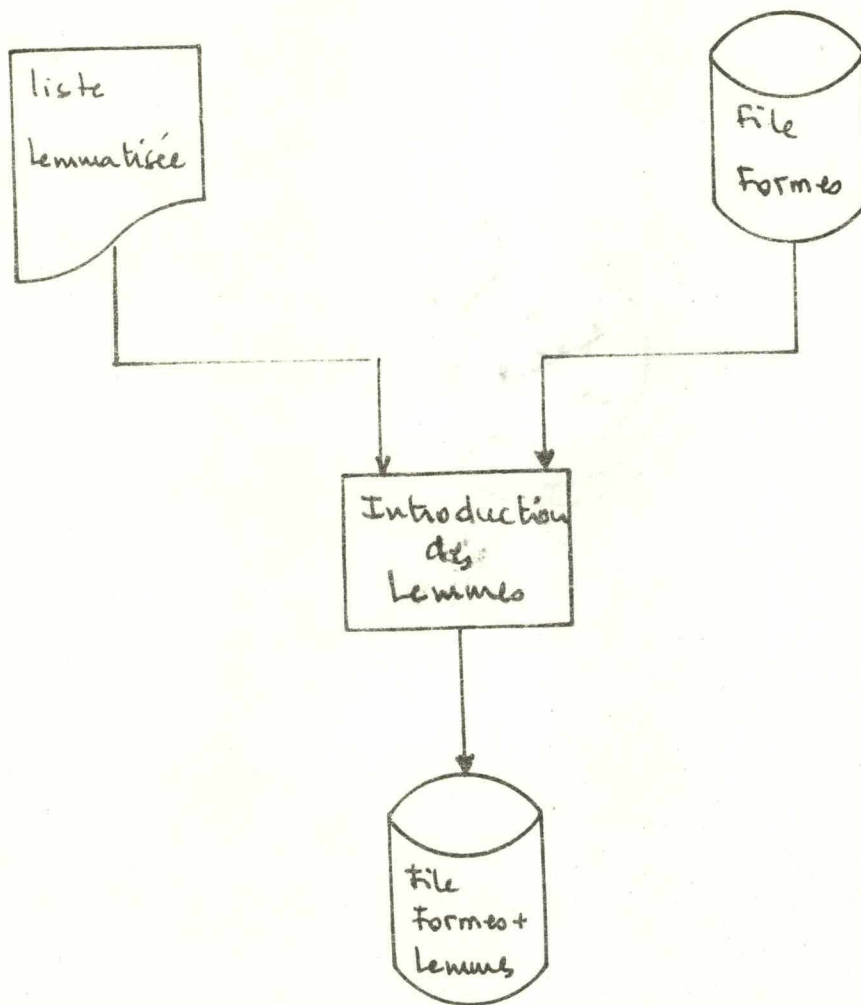


LINE P_LC LVL PASCAL 2.0

```

1      8  1) PROGRAM ORD(F,T,INPUT,OUTPUT);
2      8  1) LABEL 10,20;
3      8  1) VAR
4      8  1)   F,T: TEXT;
5     58  1)   FRM: ARRAY [1..25] OF CHAR;
6    100  1)   REF: ARRAY [1..21] OF CHAR;
7    130  1)   LP : ARRAY [1..15] OF CHAR;
8    190  1)   CV : ARRAY [1..30] OF CHAR;
9    250  1)   CON,VOY: ARRAY [1..15] OF CHAR;
10   266  1)   LS1,LS2: ARRAY [1..4] OF CHAR;
11   270  1)   LG : ARRAY [1..2] OF CHAR;
12   274  1)   C,CH: CHAR;
13   280  1)   N,I,J: INTEGER;
14   292  1)   FRQ : INTEGER;
15   292  1)
16   282  1)
17     0  1) BEGIN
18    13  1)   RESET(F); REWRITE(T);
19    15  1)   N:=0;
20    20  1)   WHILE NOT EOF(F) DO
21    20  1)     BEGIN
22    42  1)       FOR I:=1 TO 25 DO FRM[I]:=CHR(128);
23    64  1)       FOR I:=1 TO 15 DO LP[I]:=CHR(128);
24    86  1)       FOR I:=1 TO 4 DO LS1[I]:=CHR(128);
25    89  1)       LS2:=LS1;
26   111  1)       READ(F,CH,I,CH,LG[1],LG[2],CH,FRQ);
27   113  1)       I:=0;
28   117  1)       READ(F,CH);
29   117  1)
30   117  1)       (* LECTURE FORME *)
31   117  1)       (* ***** *)
32   117  1)
33   121  1)       WHILE CH<>CHR(128) DO
34   121  1)         BEGIN
35   125  1)           I:=I+1;
36   131  1)           FRM[I]:=CH;
37   135  1)           READ(F,CH);
38   136  1)         END;
39   136  1)
40   136  1)       (* LECTURE LENNE *)
41   136  1)       (* ***** *)
42   136  1)
43   145  1)       WHILE CH=CHR(128) DO READ(F,CH);
44   149  1)       J:=1; I:=0;
45   153  1)       WHILE CH<>CHR(128) DO
46   153  1)         BEGIN
47   164  1)           IF CH=CHR(141) THEN BEGIN I:=0; J:=J+1; GOTO 10; END;
48   168  1)           I:=I+1;
49   169  1)           CASE J OF
50   177  1)             1:LP[I]:=CH;
51   184  1)             2:LS1[I]:=CH;
52   191  1)             3:LS2[I]:=CH;
53   194  1)           END;
54   198  1)           10:READ(F,CH);
55   199  1)         END;
56   199  1)
57   199  1)       (* LECTURE CODE *)
58   199  1)       (* ***** *)
59   199  1)

```



NE P_LC LVL PASCAL 2.0

```
1      8 1) PROGRAM LEMME (F,T,INPUT,OUTPUT);
2      8 1) LABEL 5,10,15,20,30;
3      8 1) VAR F,T : TEXT;
4     328 1)      B : ARRAY [1..160] OF CHAR;
5     358 1)      LM : ARRAY [1..15] OF CHAR;
6     398 1)      M : ARRAY [1..20] OF CHAR;
7     402 1)      C : ARRAY [1..2] OF CHAR;
8     404 1)      CH : CHAR;
9     410 1)      J,I,N : INTEGER;
10     0 1) BEGIN
11     13 1) RESET(F); REWRITE(T);
12     19 1) WRITE("NUMERO DU RECORD = ");
13     24 1) READLN(N);
14     30 1) READ(F,CH,I);
15     38 1) IF I=N THEN BEGIN RESET(F); GOTO 15; END;
16     44 1) WHILE I<>N-1 DO BEGIN
17     47 1)      READLN(F);
18     50 1)      TRAP(255,CH);
19     56 1)      READ(F,CH,I);
20     57 1)      END;
21     60 1) READLN(F);
22     65 1) 15:WHILE NOT EOF(F) DO
23     65 1) BEGIN
24     87 1)      FOR I:=1 TO 160 DO B[I]:=CHR(128);
25     109 1)      FOR I:=1 TO 20 DO M[I]:=CHR(128);
26     111 1)      I:=0;
27     116 1)      WHILE NOT EOLN(F) DO
28     116 1)      BEGIN
29     120 1)          I:=I+1;
30     127 1)          READ(F,B[I]);
31     128 1)      END;
32     153 1) 10:FOR I:=1 TO 40 DO WRITE(B[I]);
33     155 1)      J:=0;
34     157 1)      I:=0;
35     161 1)      READ(CH);
36     165 1)      WHILE CH <> " " DO
37     165 1)      BEGIN
38     170 1)          IF CH=CHR(131) THEN GOTO 30;
39     174 1)          IF CH=CHR(143) THEN BEGIN
40     180 1)              M[I]:=CHR(128);
41     184 1)              I:=I-1;
42     185 1)              GOTO 5;
43     185 1)          END;
44     197 1)          IF (J=0) AND (CH=CHR(128)) THEN BEGIN WRITELN; GOTO 20; END;
45     201 1)          I:=I+1;
46     203 1)          J:=1;
47     209 1)          M[I]:=CH;
48     213 1)          5:READ(CH);
49     214 1)      END;
50     217 1)      WRITELN;
51     226 1)      IF M[I]=CHR(129) THEN GOTO 20;
52     248 1)      FOR I:=1 TO 15 DO LM[I]:=CHR(128);
53     250 1)      I:=1;
54     258 1)      WHILE M[I]<>CHR(128) DO BEGIN
55     268 1)          LM[I]:=M[I];
56     272 1)          I:=I+1;
57     273 1)      END;
58     281 1)      WHILE M[I]=CHR(128) DO BEGIN
59     286 1)          IF I=30 THEN GOTO 10;
```

```

60 290 1) I:=I+1;
61 291 1) END;
62 313 1) C[1]:=MCI; C[2]:=MCI+1;
63 318 1) 20:WRITE(T,CHR(128));
64 343 1) FOR I:=4 TO 40 DO WRITE(T,B[I]);
65 368 1) FOR I:=1 TO 15 DO WRITE(T,LM[I]);
66 387 1) WRITE(T,C[1],C[2],CHR(128));
67 414 1) FOR I:=1 TO 21 DO WRITE(T,B[I+41]);
68 417 1) WRITELN(T);
69 420 1) READLN(F);
70 423 1) TRAP(255,CH);
71 424 1) END;
72 410 1) 30:END.

```

```

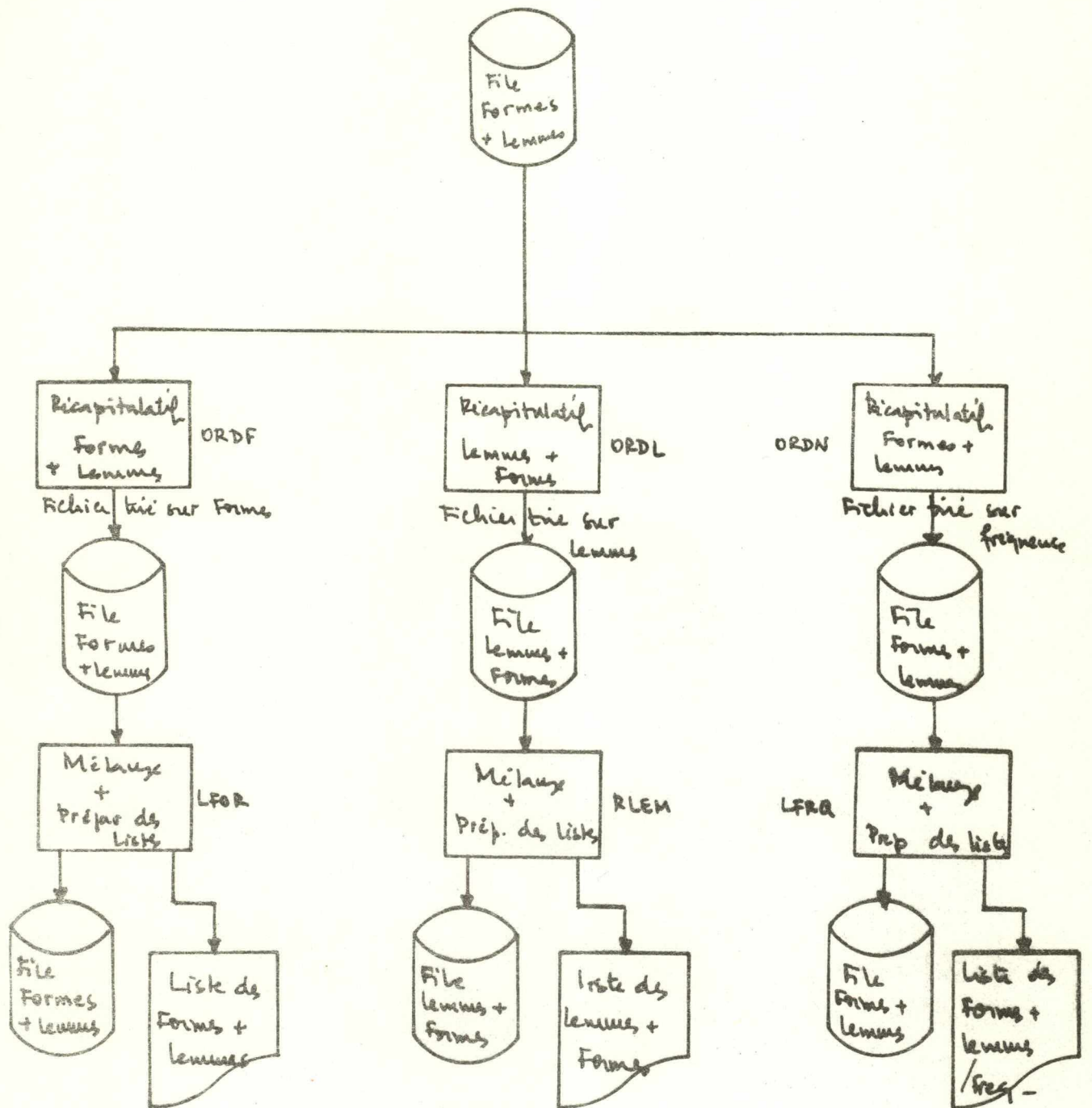
*** NO SYNTAX ERROR(S) DETECTED.
*** 72 LINE(S) READ, 0 PROCEDURE(S) COMPILED,
*** 425 P-INSTRUCTIONS GENERATED.

```

PROGRAMME DE GENERATION
DES LISTES :

- FORMES
- LEMMES
- LEMMES PAR ORDE DE
FREQUENCE
- INDEX CONTRAIRE

-----o-----



*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **
*** */ORDF.L **



96-2

96-2

INE P_LC LVL PASCAL 2.0

```
1 4 1) PRO.PAM ORD(F,T,INPUT,OUTPUT);
2 5 1) LABEL 10,20;
3 8 1) VAR
4 8 1) F,T: TEXT;
5 58 1) FRM: ARRAY [1..25] OF CHAR;
6 100 1) REF: ARRAY [1..21] OF CHAR;
7 130 1) LP : ARRAY [1..15] OF CHAR;
8 190 1) CV : ARRAY [1..30] OF CHAR;
9 250 1) CON,VOY: ARRAY [1..15] OF CHAR;
10 266 1) LS1,LS2: ARRAY [1..4] OF CHAR;
11 270 1) LG : ARRAY [1..2] OF CHAR;
12 274 1) C,CH: CHAR;
13 290 1) N,I,J: INTEGER;
14 282 1) FRQ : INTEGER;
15 282 1)
16 282 1) (* S/P DECOUPAGE CONS-VOY *)
17 282 1) (* ***** *)
18 282 1)
19 0 2) PROCEDURE D_CV;
20 0 2) LABEL 10,20;
21 10 2) VAR I,J,K1,K2,S: INTEGER;
22 0 2) BEGIN
23 23 2) FOR I:=1 TO 15 DO CON[I]:=CHR(128);
24 26 2) VOY:=CON;
25 36 2) I:=1; J:=0; K1:=0; K2:=0; S:=0;
26 44 2) WHILE CV[I]<>CHR(129) DO
27 44 2) BEGIN
28 48 2) IF J=0 THEN (* LECTURE CONSONNE *)
29 48 2) BEGIN
30 72 2) IF ((ORD(CV[I])>133) AND (ORD(CV[I])<222)) OR (ORD(CV[I])=161)
31 72 2) BEGIN
32 74 2) J:=1;
33 82 2) IF ORD(CV[I])=161 THEN
34 82 2) BEGIN
35 86 2) K1:=K1+1;
36 92 2) CON[K1]:=CHR(214);
37 96 2) K1:=K1+1;
38 102 2) CON[K1]:=CHR(191);
39 106 2) K2:=K2+1;
40 106 2) END
41 107 2) ELSE
42 107 2) BEGIN
43 111 2) K1:=K1+1;
44 121 2) CON[K1]:=CV[I];
45 121 2) END
46 121 2) END
47 122 2) BEGIN
48 122 2) BEGIN
49 126 2) IF S=1 THEN BEGIN
50 140 2) IF CV[I]=CHR(181) THEN VOY[K2]:=CHR(145);
51 154 2) IF CV[I]=CHR(182) THEN VOY[K2]:=CHR(146);
52 168 2) IF CV[I]=CHR(183) THEN VOY[K2]:=CHR(147);
53 170 2) S:=0;
54 170 2) END
55 182 2) ELSE BEGIN WRITELN('ERREUR ',CV);
56 183 2) GOTO 20;
57 183 2) END
58 183 2) END
59 183 2) END
```

96-3

```

60 184 2) ELSE (* LECTURE VOYELLE *)
61 184 2) BEGIN
62 200 2) IF (ORD(CV[I])>172) AND (ORD(CV[I])<184) THEN
63 200 2) BEGIN
64 210 2) IF ORD(CV[I])=180 THEN S:=1;
65 214 2) K2:=K2+1;
66 224 2) VOY[K2]:=CV[I];
67 226 2) J:=0;
68 226 2) END
69 227 2) ELSE
70 227 2) BEGIN
71 231 2) K2:=K2+1;
72 232 2) GOTO 10;
73 232 2) END
74 232 2) END;
75 236 2) I:=I+1;
76 237 2) END;
77 10 2) 2J:END;
78 10 2)
79 0 1) BEGIN
80 13 1) RESET(F); REWRITE(T);
81 15 1) N:=0;
82 20 1) WHILE NOT EOF(F) DO
83 20 1) BEGIN
84 42 1) FOR I:=1 TO 25 DO FRM[I]:=CHR(123);
85 64 1) FOR I:=1 TO 15 DO LP[I]:=CHR(128);
86 86 1) FOR I:=1 TO 4 DO LS1[I]:=CHR(128);
87 89 1) LS2:=LS1;
88 111 1) READ(F,CH,I,CH,LG[1],LG[2],CH,FRQ);
89 113 1) I:=0;
90 117 1) READ(F,CH);
91 117 1)
92 117 1) (* LECTURE FORME *)
93 117 1) (* ***** *)
94 117 1)
95 121 1) WHILE CH<>CHR(128) DO
96 121 1) BEGIN
97 125 1) I:=I+1;
98 131 1) FRM[I]:=CH;
99 135 1) READ(F,CH);
100 136 1) END;
101 136 1)
102 136 1) (* LECTURE LEMME *)
103 136 1) (* ***** *)
104 136 1)
105 145 1) WHILE CH=CHR(128) DO READ(F,CH);
106 149 1) J:=1; I:=0;
107 153 1) WHILE CH<>CHR(128) DO
108 153 1) BEGIN
109 154 1) IF CH=CHR(141) THEN BEGIN I:=0; J:=J+1; GOTO 10; END;
110 168 1) I:=I+1;
111 169 1) CASE J OF
112 177 1) 1:LP[I]:=CH;
113 184 1) 2:LS1[I]:=CH;
114 191 1) 3:LS2[I]:=CH;
115 194 1) END;
116 198 1) 10:READ(F,CH);
117 199 1) END;
118 199 1)
119 199 1) (* LECTURE CODE *)
120 199 1) (* ***** *)
121 199 1)

```

```

122 208 1)   WHILE CH=CHR(123) DO READ(F,CH);
123 210 1)   C:=CH;
124 210 1)
125 214 1)   READ(F,CH);
126 223 1)   WHILE CH=CHR(128) DO READ(F,CH);
127 223 1)
128 223 1)   (* LECTURE REF *)
129 223 1)   (* ***** *)
130 223 1)
131 245 1)   FOR I:=1 TO 21 DO REF[I]:=CHR(128);
132 251 1)   REF[1]:=CH;
133 253 1)   I:=2;
134 258 1)   WHILE NOT EOLN(F) DO
135 258 1)   BEGIN
136 265 1)     READ(F,REF[I]);
137 270 1)     IF I=21 THEN GOTO 20;
138 274 1)     I:=I+1;
139 275 1)   END;
140 275 1)
141 275 1)   (* OUTPUT *)
142 275 1)   (* ***** *)
143 275 1)
144 303 1)   20:WRITE(T,LG,FRQ:5,REF,LP,LS1,LS2,C);
145 303 1)
146 325 1)   FOR I:=1 TO 30 DO CV[I]:=CHR(128);
147 351 1)   FOR I:=1 TO 25 DO CV[I]:=FRM[I];
148 353 1)   D_CV;
149 364 1)   WRITELN(T,CON,VOY);
150 368 1)   N:=N+1;
151 371 1)   TRAP(255,CH);
152 374 1)   READLN(F);
153 375 1)   END;
154 385 1)   WRITELN("NOMBRE DE RECORDS = ",N:5);
155 282 1) END.

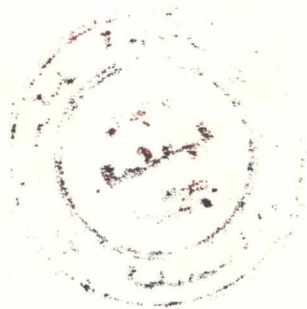
```

```

*** NO SYNTAX ERROR(S) DETECTED.
*** 155 LINE(S) READ, 1 PROCEDURE(S) COMPILED,
*** 624 P-INSTRUCTIONS GENERATED.

```

*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **
*** */LFDR.L **



LINE P_LC LVL PASCAL 2.0

```
1      8 1) PROGRAM LLEN(F,T,INPUT,OUTPUT);
2      8 1) VAR
3      8 1)   F,T: TEXT;
4      58 1)   FRM: ARRAY [1..25] OF CHAR;
5     100 1)   REF: ARRAY [1..21] OF CHAR;
6     130 1)   LP : ARRAY [1..15] OF CHAR;
7     190 1)   CV : ARRAY [1..30] OF CHAR;
8     250 1)   CON,VOY: ARRAY [1..15] OF CHAR;
9     266 1)   LS1,LS2: ARRAY [1..4] OF CHAR;
10    270 1)   LG : ARRAY [1..2] OF CHAR;
11    274 1)   C,CH: CHAR;
12    280 1)   N,I,J: INTEGER;
13    292 1)   FRQ : INTEGER;
14    292 1)
15    282 1) (* S/P REGROUPEMENT CONS-VOY *)
16    282 1) (* ***** *)
17    282 1)
18     0 2) PROCEDURE L_CV;
19     0 2) LABEL 10;
20     6 2) VAR I,J,K: INTEGER;
21     0 2) BEGIN
22     23 2)   FOR I:=1 TO 30 DO CV[I]:=CHR(128);
23     29 2)   I:=1; J:=1; K:=1;
24     37 2)   WHILE CON[J]<>CHR(128) DO
25     37 2)     BEGIN
26     61 2)       IF (CON[J]=CHR(214)) AND (CON[J+1]=CHR(191)) AND (VOY[K]=CHR(
27     63 2)         THEN BEGIN
28     69 2)           CV[I]:=CHR(161);
29     77 2)           J:=J+1; K:=K+1;
30     77 2)         END
31     58 2)       ELSE CV[I]:=CON[J];
32     92 2)       I:=I+1;
33    101 2)       IF VOY[K]=CHR(128) THEN GOTO 10;
34    117 2)       IF (ORD(VOY[K])>=145) AND (ORD(VOY[K])<=147) THEN
35    117 2)         BEGIN
36    123 2)           CV[I]:=CHR(180);
37    127 2)           I:=I+1;
38    139 2)           CV[I]:=CHR(ORD(VOY[K])+36);
39    140 2)           GOTO 10;
40    140 2)         END;
41    154 2)         CV[I]:=VOY[K]; I:=I+1;
42    162 2)         TO:J:=J+1; K:=K+1;
43    163 2)         END;
44     6 2)   END;
45     6 2)
46     6 2)
47     0 1) BEGIN
48    13 1)   RESET(F); REWRITE(T);
49    15 1)   N:=0;
50    20 1)   WHILE NOT EOF(F) DO
51    20 1)     BEGIN
52    41 1)       READ(F,LG,FRQ,REF,LP,LS1,LS2,C);
53    66 1)       FOR I:=1 TO 15 DO CON[I]:=CHR(128); VOY:=CON;
54    74 1)       READ(F,CON,VOY);
55    76 1)       L_CV;
56   102 1)       FOR I:=1 TO 25 DO FRM[I]:=CV[I];
57   106 1)       N:=N+1;
58   139 1)       WRITELN(T,CHR(128),N:5,LG,FRQ:5,FRM:22,LP:15,LS1:3,LS2:3,C,
59   148 1)         CHR(128),RE=);
```

```
60 151 1)          READLN(F);
61 154 1)          TRAP(255,CH);
62 155 1)          END;
63 165 1)          WRITELN("NOMBRE DE RECORDS = ",N:5);
64 282 1) END.
```

```
*** NO SYNTAX ERROR(S) DETECTED.
*** 64 LINE(S) READ, 1 PROCEDURE(S) COMPILED,
*** 330 P-INSTRUCTIONS GENERATED.
```

96-8

*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **
*** */ORDL.L **

LINE P_LC LVL PASCAL 2.0

```

1      8  1) PROGRAM ORD(F,T,INPUT,OUTPUT);
2      8  1) LABEL 10,20;
3      8  1) VAR
4      8  1)   F,T: TEXT;
5     58  1)   FRM: ARRAY [1..25] OF CHAR;
6    100  1)   REF: ARRAY [1..21] OF CHAR;
7    130  1)   LP : ARRAY [1..15] OF CHAR;
8    190  1)   CV : ARRAY [1..30] OF CHAR;
9    250  1)   CON,VOY: ARRAY [1..15] OF CHAR;
10   266  1)   LS1,LS2: ARRAY [1..4] OF CHAR;
11   270  1)   LG : ARRAY [1..2] OF CHAR;
12   274  1)   C,CH: CHAR;
13   280  1)   N,I,J: INTEGER;
14   282  1)   FRQ : INTEGER;
15   282  1)
16   282  1) (* S/P DECOUPAGE CCNS-VOY *)
17   282  1) (* ***** *)
18   282  1)
19      0  2) PROCEDURE D_CV;
20      0  2) LABEL 10,20;
21     10  2) VAR I,J,K1,K2,S: INTEGER;
22      0  2) BEGIN
23     23  2)   FOR I:=1 TO 15 DO CON[I]:=CHR(128);
24     26  2)   VOY:=CON;
25     36  2)   I:=1; J:=0; K1:=0; K2:=0; S:=0;
26     44  2)   WHILE CV[I]<>CHR(128) DO
27     44  2)   BEGIN
28     48  2)     IF J=0 THEN (* LECTURE CONSONNE *)
29     48  2)     BEGIN
30     72  2)       IF ((ORD(CV[I])>183) AND (ORD(CV[I])<222)) OR (ORD(CV[I])=161
31     72  2)       BEGIN
32     74  2)         10: J:=1;
33     82  2)         IF ORD(CV[I])=161 THEN
34     82  2)         BEGIN
35     86  2)           K1:=K1+1;
36     92  2)           CON[K1]:=CHR(214);
37     96  2)           K1:=K1+1;
38    102  2)           CON[K1]:=CHR(191);
39    106  2)           K2:=K2+1;
40    106  2)         END
41    107  2)         ELSE
42    107  2)         BEGIN
43    111  2)           K1:=K1+1;
44    121  2)           CON[K1]:=CV[I];
45    121  2)         END
46    121  2)       END
47    122  2)     BEGIN
48    122  2)     IF S=1 THEN BEGIN
49    126  2)       IF CV[I]=CHR(181) THEN VOY[K2]:=CHR(145);
50    140  2)       IF CV[I]=CHR(182) THEN VOY[K2]:=CHR(146);
51    154  2)       IF CV[I]=CHR(183) THEN VOY[K2]:=CHR(147);
52    168  2)       S:=0;
53    170  2)     END
54    170  2)     ELSE BEGIN WRITELN('ERREUR ',CV);
55    182  2)       GOTO 20;
56    183  2)     END
57    183  2)   END
58    183  2) END
59    183  2)

```

```

60 184 2) ELSE (* LECTURE VOYELLE *)
61 184 2) BEGIN
62 200 2) IF (ORD(CV[I])>172) AND (ORD(CV[I])<184) THEN
63 200 2) BEGIN
64 210 2) IF ORD(CV[I])=180 THEN S:=1;
65 214 2) K2:=K2+1;
66 224 2) VOY[K2]:=CV[I];
67 226 2) J:=0;
68 226 2) END
69 227 2) END ELSE
70 227 2) BEGIN
71 231 2) K2:=K2+1;
72 232 2) GOTO 10;
73 232 2) END
74 232 2) END;
75 236 2) I:=I+1;
76 237 2) END;
77 10 2) 20:END;
78 10 2)
79 0 1) BEGIN
80 13 1) RESET(F); REWRITE(T);
81 15 1) N:=0;
82 20 1) WHILE NOT ECF(F) DO
83 20 1) BEGIN
84 42 1) FOR I:=1 TO 25 DO FRM[I]:=CHR(128);
85 64 1) FOR I:=1 TO 15 DO LP[I]:=CHR(128);
86 86 1) FOR I:=1 TO 4 DO LS1[I]:=CHR(128);
87 89 1) LS2:=LS1;
88 111 1) READ(F,CH,I,CH,LG[1],LG[2],CH,FRQ);
89 113 1) I:=0;
90 117 1) READ(F,CH);
91 117 1)
92 117 1) (* LECTURE FORME *)
93 117 1) (* ***** *)
94 117 1)
95 121 1) WHILE CH<>CHR(123) DO
96 121 1) BEGIN
97 125 1) I:=I+1;
98 131 1) FRM[I]:=CH;
99 135 1) READ(F,CH);
100 136 1) END;
101 136 1)
102 136 1) (* LECTURE LEMME *)
103 136 1) (* ***** *)
104 136 1)
105 145 1) WHILE CH=CHR(128) DO READ(F,CH);
106 149 1) J:=1; I:=0;
107 153 1) WHILE CH<>CHR(128) DO
108 153 1) BEGIN
109 164 1) IF CH=CHR(141) THEN BEGIN I:=0; J:=J+1; GOTO 10; END;
110 168 1) I:=I+1;
111 169 1) CASE J OF
112 177 1) 1:LP[I]:=CH;
113 184 1) 2:LS1[I]:=CH;
114 191 1) 3:LS2[I]:=CH;
115 194 1) END;
116 198 1) 10:READ(F,CH);
117 199 1) END;
118 199 1)
119 199 1) (* LECTURE CODE *)
120 199 1) (* ***** *)
121 199 1)

```

```

122 208 1) WHILE CH=CHR(128) DO READ(F,CH);
123 210 1) C:=CH;
124 210 1)
125 214 1) READ(F,CH);
126 223 1) WHILE CH=CHR(128) DO READ(F,CH);
127 223 1)
128 223 1) (* LECTURE REF *)
129 223 1) (* ***** *)
130 223 1)
131 245 1) FOR I:=1 TO 21 DO REF[I]:=CHR(128);
132 251 1) REF[1]:=CH;
133 253 1) I:=2;
134 258 1) WHILE NOT EOLN(F) DO
135 258 1) BEGIN
136 265 1) READ(F,REF[I]);
137 270 1) IF I=21 THEN GOTO 20;
138 274 1) I:=I+1;
139 275 1) END;
140 275 1)
141 275 1) (* OUTPUT *)
142 275 1) (* ***** *)
143 275 1)
144 288 1) 20:WRITE(T,10,FR:5,REF);
145 310 1) FOR I:=1 TO 30 DO CV[I]:=CHR(128);
146 336 1) FOR I:=1 TO 15 DO CV[I]:=LPC[I];
147 338 1) D_CV;
148 351 1) WRITE(T,CON:TQ,V0Y:10,C);
149 351 1)
150 373 1) FOR I:=1 TO 30 DO CV[I]:=CHR(128);
151 399 1) FOR I:=1 TO 25 DO CV[I]:=FRM[I];
152 401 1) D_CV;
153 420 1) WRITELN(T,CON,V0Y,LS1,LS2);
154 424 1) N:=N+1;
155 427 1) TRAP(255,CH);
156 430 1) READLN(F);
157 431 1) END;
158 441 1) WRITELN('NOMBRE DE RECORDS = ',N:5);
159 282 1) END.

```

```

**** NO SYNTAX ERROR(S) DETECTED.
**** 159 LINE(S) READ, 1 PROCEDURE(S) COMPILED,
**** 690 P-INSTRUCTIONS GENERATED.

```

*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **
*** */LFRQ.L **



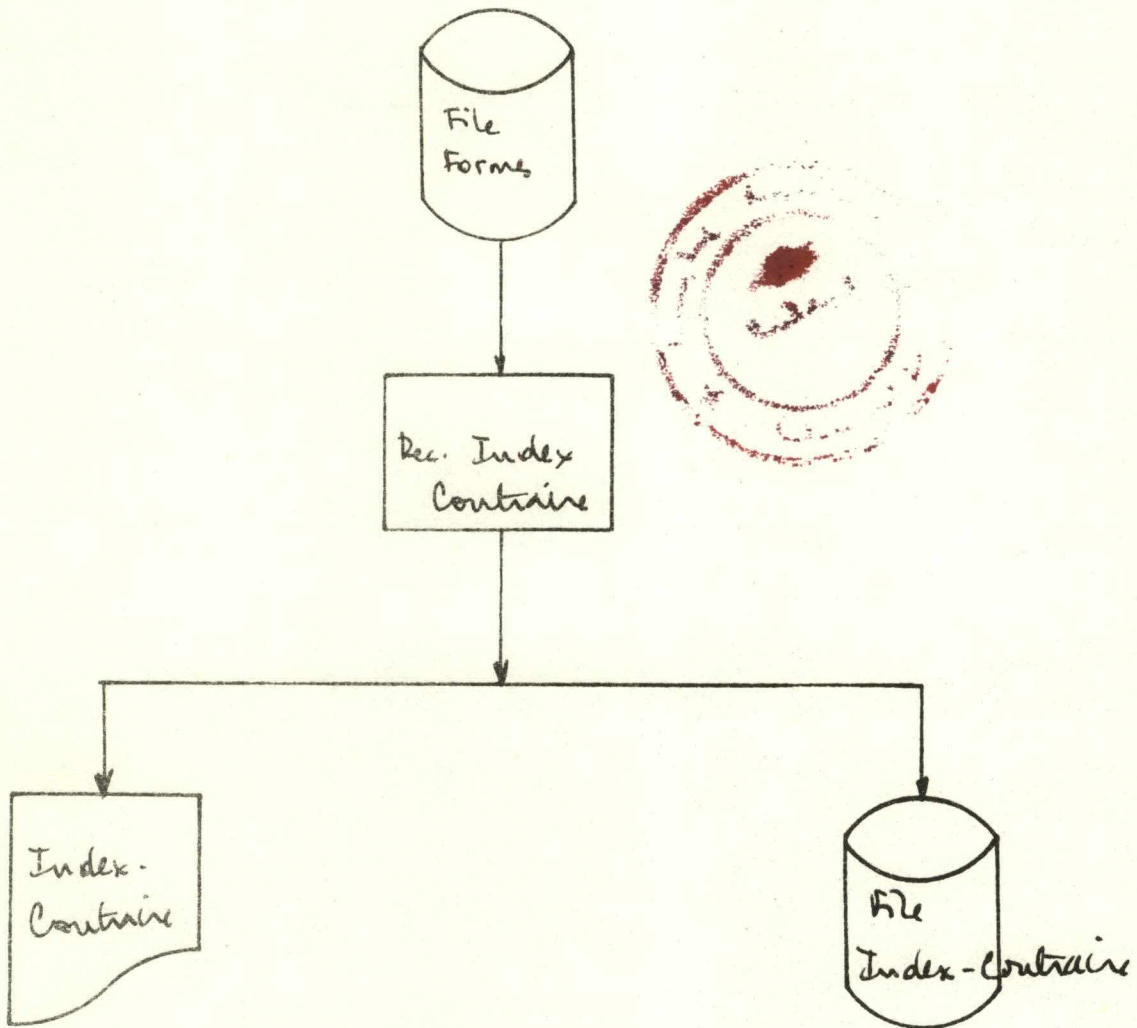
LIVE P_LC LVL PASCAL 2.0

```
1 3 1) PROGRAM LLEM(F,T,INPUT,OUTPUT);
2 8 1) VAR
3 8 1) F,T: TEXT;
4 33 1) FRM: ARRAY [1..25] OF CHAR;
5 100 1) REF: ARRAY [1..21] OF CHAR;
6 130 1) LP : ARRAY [1..15] OF CHAR;
7 146 1) LS1,LS2: ARRAY [1..4] OF CHAR;
8 150 1) LG : ARRAY [1..2] OF CHAR;
9 154 1) C,CH: CHAR;
10 160 1) N,I,J: INTEGER;
11 162 1) FRQ : INTEGER;
12 162 1)
13 162 1)
14 0 1) BEGIN
15 13 1) RESET(F); REWRITE(T);
16 15 1) N:=0;
17 20 1) WHILE NOT EOF(F) DO
18 20 1) BEGIN
19 44 1) READ(F,LC,REF,LP,LS1,LS2,C FRM,FR);
20 48 1) N:=N+1;
21 78 1) WRITELN(T,CHR(128),FRQ:5,L ,FRM,LP:15,LS1:3,LS2:3,C,
22 37 1) CHR(128),REF);
23 90 1) READLN(F);
24 93 1) TRAP(255,CH);
25 94 1) END;
26 104 1) WRITELN('NOMBRE DE RECORDS = ' N:5);
27 162 1) END.
```

```
**** NO SYNTAX ERROR(S) DETECTED.
**** 27 LINE(S) READ, 0 PROCEDURE(S) COMPIL D,
**** 105 P-INSTRUCTIONS GENERATED.
```

96 - 20

96 20



*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **
*** */RECI.L **


```

LINE  P_LC LVL  PASCAL  2.0
  1    0  1)          (* RECAPITULATIF INDEX-CONTRAIRE *)
  2    0  1)
  3    6  1) PROGRAM  REC_I (F,T1,OUTPUT);
  4    6  1) LABEL    30,50,100;
  5   222 1) VAR      GRD,BUF:          ARRAY [1..54] OF CHAR;
  6   224 1)          CH:              CHAR;
  7   224 1)          F,T1:           TEXT;
  8   238 1)          S,NR1,NR2,I,FR,J,K: INTEGER;
  9   242 1)          FRS:            REAL;
 10   242 1)
 11   242 1)          (* PROCEDURE ECRITURE INDEX-CONTRAIRE *)
 12   242 1)
 13    0  2) PROCEDURE ECR_IDC;
 14    0  2) BEGIN
 15    7  2)   IF NR2=999 THEN BEGIN NR2:=0;
 16   11  2)   NR1:=NR1+1;
 17   11  2)   END
 18   16  2)   ELSE NR2:=NR2+1;
 19   21  2)   WRITE (T1,CHR(128));
 20   30  2)   IF NR1=0 THEN WRITE (T1,NR2:6)
 21   31  2)   ELSE BEGIN
 22   36  2)   WRITE (T1,NR1:3);
 23   51  2)   IF NR2<10 THEN WRITE (T1,CHR(48),CHR(48),NR2:1)
 24   56  2)   ELSE IF NR2<100 THEN
 25   64  2)   WRITE (T1,CHR(48),NR2:2)
 26   70  2)   ELSE WRITE (T1,NR2:3)
 27   70  2)   END;
 28   95  2)   WRITE (T1,CHR(128),GRD[23],GRD[24],CHR(128),FR:4); (* LG ET FREQ *)
 29   97  2)   I:=50;
 30   97  2)   (* INDEX CONTRAIRE *)
 31  105  2)   WHILE GRD[I] = CHR(128) DO BEGIN
 32  110  2)   WRITE (T1,CHR(128));
 33  114  2)   I:=I-1;
 34  115  2)   END;
 35  117  2)   I:=25;
 36  125  2)   WHILE GRD[I] <> CHR(128) DO BEGIN
 37  134  2)   WRITE (T1,GRD[I]);
 38  138  2)   I:=I+1;
 39  139  2)   END;
 40  142  2)   WRITELN (T1);
 41    0  2) END;
 42    0  2)
 43    0  1) BEGIN
 44    7  1)   RESET (F);
 45   10  1)   READLN (F);
 46   13  1)   READLN (F);
 47   16  1)   REWRITE (T1);
 48   28  1)   WRITE (T1,CHR(32):40,"INDEX-CONTRAIRE",CHR(32):10);
 49   34  1)   WRITE (T1,"*FRQ*LG*N-REC*");
 50   37  1)   WRITELN (T1);
 51   40  1)   WRITELN (T1);
 52   42  1)   NR1:=0;
 53   44  1)   NR2:=0;
 54   47  1)   FRS:=0;
 55   49  1)   S:=0;
 56   54  1)   WHILE NOT EOF (F) DO
 57   54  1)   BEGIN
 58   76  1)   FOR I:=1 TO 54 DO BUF[I]:=CHR(128);
 59   78  1)   I:=0;

```

```

60      83  1)      WHILE NOT EOLN (F) DO
61      83  1)      BEGIN
62      87  1)          I:=I+1;
63      91  1)          IF I>54 THEN BEGIN
64      98  1)              WRITELN (T1, '** ERREUR ENRG INPUT **')
65      99  1)              GOTO 100;
66      99  1)          END;
67     106  1)          READ (F, BUF[I]);
68     107  1)      END;
69     110  1)      TRAP (255, CH);
70     114  1)      IF S=0 THEN BEGIN
71     140  1)          FOR I:=1 TO 54 DO GRD[I]:=BUF[I];
72     142  1)          S:=1;
73     144  1)          FR:=1;
74     145  1)          GOTO 50;
75     145  1)      END;
76     145  1)
77     145  1)      (* COMPARAISON ELEMENT *)
78     145  1)
79     174  1)      FOR I:=25 TO 54 DO IF BUF[I] <> GRD[I] THEN GOTO 30;
80     174  1)
81     174  1)      (* MEME ELEMENT *)
82     174  1)
83     178  1)      FR:=FR+1;
84     179  1)      GOTO 50;
85     179  1)
86     179  1)      (* MOTS DIFFERENTS *);
87     181  1)      30: ECR_IDC;
88     186  1)          FRS:=FRS+FR;
89     212  1)          FOR I:=1 TO 54 DO GRD[I]:=BUF[I];
90     214  1)          FR:=1;
91     217  1)          50: READLN (F);
92     218  1)      END;
93     220  1)      ECR_IDC;
94     225  1)      FRS:=FRS+FR;
95     228  1)      WRITELN (T1);
96     231  1)      WRITELN (T1);
97     241  1)      WRITELN (T1, 'NOMBRE DE MOTS =', FRS);
98     251  1)      WRITELN ('NOMBRE DE MOTS =', FRS);
99     242  1)  100: END.

```

```

****      NO SYNTAX ERROR(S) DETECTED.
****      99 LINE(S) READ,      1 PROCEDURE(S) COMPILED,
****      395 P-INSTRUCTIONS GENERATED.

```

b ₇	0	0	0	0	1	1	1	1					
b ₆	0	0	1	1	0	0	1	1					
b ₅	0	1	0	1	0	1	0	1					
		0	1	2	3	4	5	6	7				
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0	NUL	CADDR	SP	•	0	@	'	ب	ظ
0	0	0	1	1	PON	DC1 XON	!	1	1	لا	'	ت	ع
0	0	1	0	2	POFF	SEND LINE	"	2	2		'	ة	ف
0	0	1	1	3		DC3 XOFF	#	3	3		'	ث	ف
0	1	0	0	4		SEND SCREEN	ع	4	4		'	ج	ق
0	1	0	1	5		PRINT	%	5	5		'	ح	ك
0	1	1	0	6	FOL	EOP	&	6	6		'	خ	ل
0	1	1	1	7	BEL	OVER RIDE	'	7	7		'	د	م
1	0	0	0	8	SPACE BACK	→	(8	8		'	ز	ن
1	0	0	1	9	TAB	↑)	9	9		'	ر	ه
1	0	1	0	A	LINE FEED	↓	*	:	:		'	ز	و
1	0	1	1	B	IL	ESC	+	!	!		'	س	ي
1	1	0	0	C	HOME	IC	,	<	<		'	ش	ا
1	1	0	1	D	LINE RETURN	DC	-	=	=	3	4	ص	ى
1	1	1	0	E	SO ARABIC	DL	.	>	>	4	ء	ض	-
1	1	1	1	F	SI LATIN	CLEAR	/	?	?	5	ا	ط	DEL

ASCII ARABIC CODE SET

				b ₇	0	0	0	0	1	1	1	1
				b ₆	0	0	1	1	0	0	1	1
				b ₅	0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7
b ₄	b ₃	b ₂	b ₁									
0	0	0	0	0	NULL	CADDR	SP	0	@	P	'	p
0	0	0	1	1	PON	DC1 XON	!	1	A	Q	a	q
0	0	1	0	2	POFF	SEND LINE	"	2	B	R	b	r
0	0	1	1	3		DC3 XOFF		3	C	S	c	s
0	1	0	0	4		SEND SCREEN	\$	4	D	T	d	t
0	1	0	1	5		PRINT	%	5	E	U	e	u
0	1	1	0	6	EOL	EOP	&	6	F	V	f	v
0	1	1	1	7	BELL	OVER PIDE	'	7	G	W	g	w
1	0	0	0	8	BACK SPACE	→	(8	H	X	h	x
1	0	0	1	9	TAB	↑)	9	I	Y	i	y
1	0	1	0	A	LINE FEED	↓	*	:	J	Z	j	z
1	0	1	1	B	IL	ESC	+	;	K	[k	
1	1	0	0	C	HOME	IC	,	<	L	\	l	!
1	1	0	1	D	LINE RETURN	DC	-	=	M]	m	
1	1	1	0	E	SO ARAB	DL	.	>	N	^	n	
1	1	1	1	F	SI LATIN	CLEAR	/	?	O	-	o	RUB OUT

ASCII CODE SET LATIN

BIBLIOGRAPHIE

-----o-----

- Roberto BUSA - THE USE OF PUNCHED CARDS IN LINGUISTIC ANALYSIS
2nd édition R.S Casey New - York 1958
- UN LEXIQUE LATIN ELECTRONIQUE
édition de l'Académie Tchecoslovaque des sciences Prague 1968
- COMPUTERS AND THE HUMANITIES
Vol 14, no 2, oct 1980
The anals of humanities computing,
the index thomisticus
- AN INVENTORY OF FIFTEEN MILLION WORDS
In IBM literary data processing
Conference proceeding september 9-11, 1964
- Mustapha CHOUEMI - LE VERBE DANS LE CORAN
Editeur KLINCESIECH,
Paris 1968
- A. HADJ - SALAH - LINGUISTIQUE ARABE ET LINGUISTIQUE GENERALE
Essai de Méthodologie et d'épistémologie
du 'Ilm al 'Arabiyya Alger 1979
- Susan HOCKEY - A GUID TO COMPUTER APPLICATIONS IN THE HUMANITIES
Duckworth, London 1980
- Salah Eddine KECHRID - TRADUCTION ET NOTES DU CORAN
1ère édition Habib El Lamssi
Beyrouth 1983

- Régis BLACHERE - LE CORAN
6ème édition 1980
Presse Universitaire de France
(Que-sais-je ?)
- INTRODUCTION AU CORAN
2ème édition 1977
GP Maisonneuve et Larose
- P. BRATLEY, S. LUSIGNAN - INFORMATION PROCESSING IN DICTIONARY
MAKING
Some technical guidelines
Publication no 196, Département
d'information - Université de Montréal
- RG. MILLS - MAN MACHINE COMMUNICATION AND PROBLEM
SOLVING
C.A Cuadra (ed) Annual Review of information
Science and technology 1967
- A.W PRATT, M.G PACAK - AUTOMATIC PROCESSING OF MEDICAL ENGLISH
International conference on computational
linguistics Stockholom, Suède 1969
- TABARI - MOHAMED SCEAU DES PROPHETES
Islam - Sindbad 1980
- A. ZAMPOLLI - TEXT PROCESSING FOR INDICES AND CONCORDANCES
(notes 1984)
- M. Fuad 'ABD-AL-BAQI - DICTIONNAIRE INDEXE DES MOTS DU CORAN
Beyrouth, 1364 H / 1945



MM. ALLARD, ELZIERRE, GARDIN, HOURS
- ANALYSE CONCEPTUELLE DU CORAN SUR CARTES PERFOREES

1963

Bernard QUEMADA - CAHIERS DE LEXICOLOGIE
Revue internationale de lexicologie et lexicographie

Paul IMPS - TRESOR DE LA LANGUE FRANCAISE
Edité par le centre national de la recherche scientifique (Nancy)
Distributeur exclusif KLINCKSIECK

- LA PRATIQUE DES ORDINATEURS DANS LA CRITIQUE DES TEXTES
Colloques internationaux du CNRS
no 579 / 21-31 mars 1978

M. Said CHERIFI - LE TEXTE CORANIQUE
ed SNED Unité Réghia 1984

- DICTIONNAIRE ARABE
El Mungad fi al lugatu wa al 'i'llam
édition 26 Dar al Masraq 1975

