

DÉPARTEMENT DE L'INTÉRIEUR ET DE LA SANTÉ PUBLIQUE

Institut universitaire  
de médecine sociale et préventive  
Lausanne

**PROGRAMMES INFORMATIQUES  
PERMETTANT L'ADAPTATION DES DRG  
AUX STATISTIQUES MEDICALES VESKA**

R. GRIMM, Y. EGGLI, V. KOEHN

Travaux effectués dans le cadre de l'étude "Case Mix", menée par l'Institut universitaire de médecine sociale et préventive de Lausanne et le Service de la santé publique et de la planification sanitaire du canton de Vaud, en collaboration avec les cantons de Berne, Fribourg, Genève, Jura, Neuchâtel, Soleure, Tessin et Valais.

Lausanne, juin 1988

CAHIERS DE RECHERCHES  
ET DE DOCUMENTATION

Adresse pour commande : Institut universitaire de médecine sociale et préventive  
Bibliothèque  
17, rue du Bugnon - CH 1005 Lausanne

Citation suggérée : Grimm R., Eggli Y., Koehn V. - Programmes  
informatiques permettant l'adaptation des DRG  
aux statistiques médicales VESKA. - Lausanne,  
Institut universitaire de médecine sociale et  
préventive, 1988, 19 p. - (Cah Rech Doc IUMSP,  
no 25)

Doc. 5  
5925/92

Institut Universitaire de  
Médecine Sociale et Préventive  
Lausanne

860

10849 B

## INTRODUCTION

L'attribution d'un DRG à chaque séjour hospitalier de soins aigus se fait en trois étapes: - pré-traitement  
- groupement  
- post-traitement

La logique de ces étapes a déjà été développée ailleurs (1,2,3,4). Le présent cahier documente les différents programmes informatiques nécessaires à l'adaptation suisse des DRG.

- La phase de pré-traitement est écrite en langage PASCAL et a été implantée sur CDC CYBER-855 et sur VAX-750.
- Le "Grouper" utilisé était écrit en ASSEMBLEUR et a été installé sur un ordinateur IBM.
- La troisième phase a été réalisée au moyen du progiciel statistique SPSS-X (sous le système d'exploitation VAX-VMS).

L'existence d'un certain nombre de fichiers intermédiaires est donc inévitable. Le schéma de la page suivante décrit l'enchaînement de ces différents programmes. Les pages 3 à 6 de ce cahier reprennent un à un programmes et fichiers nécessaires.

### a) Pré-traitement

Après lecture de la bande VESKA (fichier "ENTREE"), le programme JUSTIF modifie le format des données de la Statistique Médicale Veska. Le programme de transcodage TRANSCO transforme les codes diagnostiques et opératoires de la VESKA (version 1979) en codes ICD9-CM compatibles avec le GROUPER (groupeur de Yale). Les codes VESKA sans équivalent sont transformés en codes spéciaux, ignorés par le GROUPER, et seront repris dans la phase de post-traitement. Le fichier PROBLEM comprend tous les codes ne faisant pas partie des tables de transcodage et permet de détecter les codes VESKA invalides ou des oublis éventuels dans la table de transcodage; ce fichier doit impérativement être vide avant de passer au groupement des séjours en DRG.

### b) Groupement

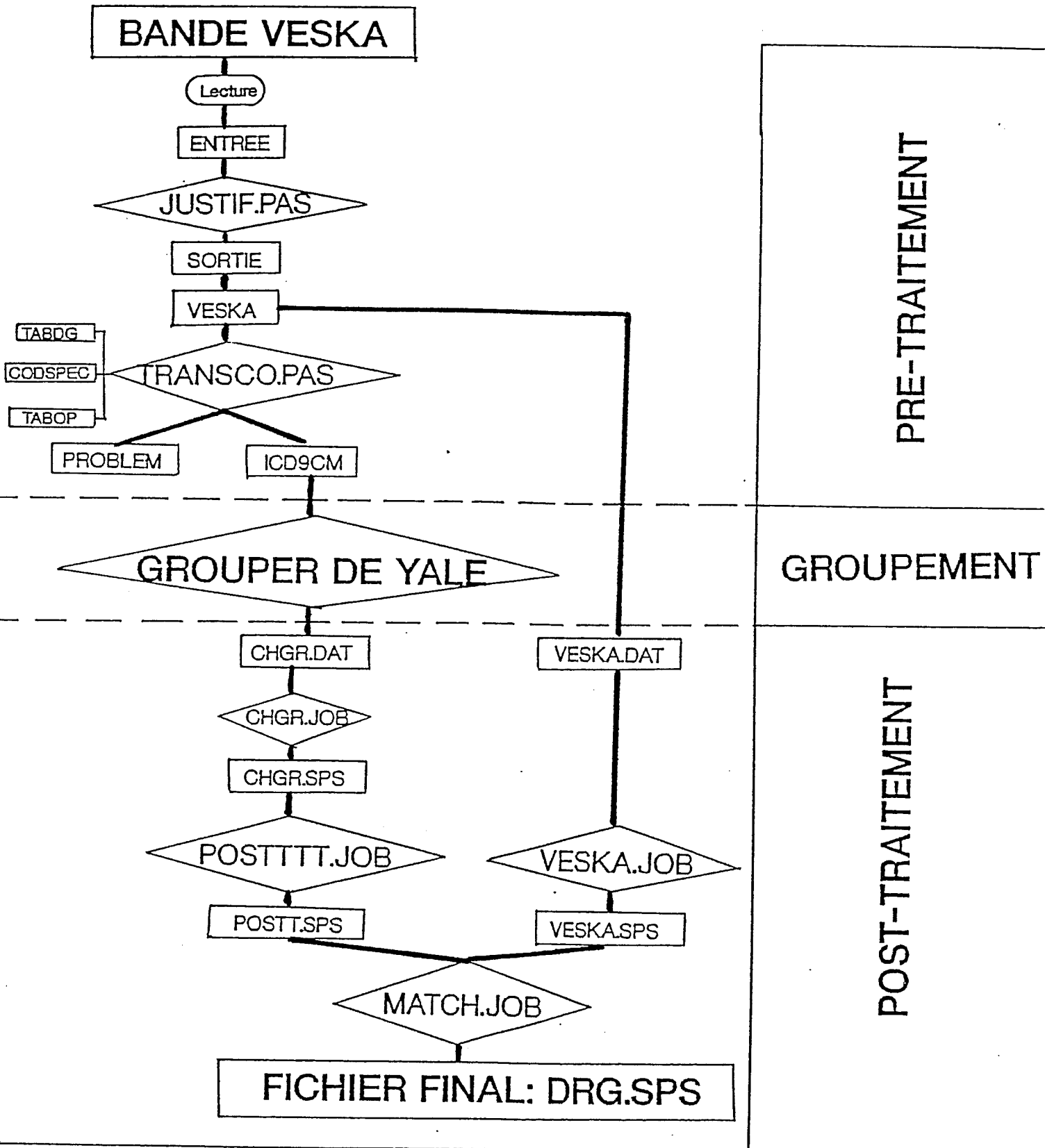
Le fichier ICD9CM est traité par le GROUPER. Ce programme est documenté ailleurs par ses auteurs (5).

### c) Post-traitement

Cette étape attribue les DRG adéquats aux séjours pour lesquels certains codes VESKA n'ont pas de correspondant ICD-9-CM.

Le fichier POSTT.DAT comprenant tous les séjours groupés en DRG est alors fusionné avec le fichier initial VESKA

# DE LA BANDE VESKA AUX DRG



## LEGENDE:

xxx Fichier de nom logique xxx

◇ Programme

Changeement de nom logique

Fusion de deux fichiers

## 1. LE PROGRAMME JUSTIF.PAS

Ce programme (ANNEXE 1) modifie le format dans lequel se trouve initialement la statistique médicale VESKA :

Hôpital	1-6
Service	7-9
Nom (champ en principe occulté)	10-29
Sexe	30
Date de naissance (année, mois, jour)	31-36
Numéro du dossier	37-47 (A)
Numéro postal du domicile	48-51
État civil	52
Confession	53
Nationalité	54-56 (A)
Date d'entrée (année, mois, jour)	57-62
Date de sortie (année, mois, jour)	63-68
Age à la sortie	69-71
Durée de séjour	72-76
Genre d'admission	77
Code d'admission	78
Code de sortie	79
Naissance multiple	80
Réserve	81-100 (A)
Première admission pour diagnostic	101
Date de l'opération principale aammjj	102-107
Nombre de diagnostics	108-109
Nombre d'opérations	110-111

puis la liste des codes diagnostiques, diagnostic principal en tête, et celle des codes opératoires, chaque code en format A7.

Le programme JUSTIF.PAS supprime les champs "Nom" et "Réserve", justifie tous les champs alphanumériques à gauche et les champs numériques à droite et répartit l'enregistrement initial sur trois lignes :

1ère ligne :

Hôpital	1-6
Service	7-9
Sexe	10
Date de naissance (année, mois, jour)	11-16
Numéro du dossier	17-27
Numéro postal du domicile	28-31
Etat civil	32
Confession	33
Nationalité	34-36
Date d'entrée (année, mois, jour)	37-42
Date de sortie (année, mois, jour)	43-48
Age à la sortie	49-51
Durée de séjour	52-56
Genre d'admission	57
Code d'admission	58
Code de sortie	59
Naissance multiple	60
Première admission pour diagnostic	61
Date de l'opération principale aammjj	62-67
Nombre de diagnostics	68-69
Nombre d'opérations	70-71

2ème ligne :

Diagnostic principal (1er)	1-7
Diagnostics secondaires (2ème)	8-14
. . .	. . .
(10ème)	64-70

3ème ligne :

Opérations (1ère)	1-7
. . .	. . .
(10ème)	64-70

diagnostics, diagnostic

## 2. PROGRAMME DE TRANSCODAGE : TRANSCO.PAS

Ce programme (ANNEXE 2) transforme les enregistrements de la statistique médicale VESKA en données lisibles par le programme de groupement (GROUPER). Cette transformation consiste à :

- transcoder les codes de sortie (4);
- traiter les codes S ("Status nach, séquelles") et "000" (1); les séjours comprenant de tels codes portent alors des marques "S" (pour les codes S) et "Z" (pour les codes 000);
- transcoder les codes diagnostiques et opératoires (2,3).

Le programme TRANSCO utilise :

- Le fichier CODESPEC qui contient la liste des codes opératoires dits "spéciaux", ceux dont le transcodage dépend des diagnostics associés, ainsi que ces mêmes diagnostics associés.
- Des co-fichiers comprenant les deux tables de transcodage suivantes :

TABOP : Table de transcodage des codes opératoires (hormis ceux qui figurent dans CODESPEC)

TABDG : Table de transcodage des codes diagnostiques

Les fichiers de sortie sont :

ICD-9-CM : C'est ce fichier qui sera traité par le programme de groupement.

Chaque enregistrement contient (sur une ligne) :

Sexe (A1)  
 Age (I3)  
 Mode de sortie transcodé (A2)  
 Marque S  
 Marque Z  
 Diagnostic principal transcodé (A5)  
 Diagnostics 2-10 (A5)  
 Opérations 1-10 (A5)

Les codes (diagnostiques ou opératoires) inexistantes sont remplacés par 5 blancs

PROBLEM : Ce fichier contient les codes VESKA non prévus dans les tables de transcodage.

Tous ces fichiers figurent dans l'en-tête du programme TRANSCO dans l'ordre suivant :

VESKA, ICD-9-CM, CODSPEC, TABDG, TABOP, PROBLEM (et OUTPUT).

### 3. DESCRIPTION DU FICHIER-OUTPUT DU GROUPEUR DE YALE

Format : longueur fixe de 129 caractères découpés comme suit :

SEXE	1		
SORTTR	5-6		
CODES	7	(A)	
CODE000	8	(A)	
DGTR1	9-13	(A)	Diagnostic principal (transcodé)
DGTR2	14-18	(A)	Diagnostic secondaire (transcodé)
DGTR3	19-23	(A)	Troisième diagnostic (transcodé)
...	...		...
DGTR10	54-58	(A)	Dixième diagnostic (transcodé)
OPTR1	59-62	(A)	Opération principale (transcodée)
OPTR2	63-66	(A)	Opération secondaire (transcodée)
OPTR3	67-70	(A)	
...	...		...
OPTR10	95-98	(A)	Dixième opération (transcodée)
DRG	99-101		DRG affecté par le "grouper"
MDC	102-103		
RTC	104		Voir ci-dessous
MPR	105-108		"
ADX	109-113		"
SDX	114-118		"
HSI	119-129	(A)	"

RTC : Code de retour du grouper

- 0 : Enregistrement groupé
- 1 : Diagnostic principal invalide
- 2 : L'enregistrement ne concorde avec aucun des DRG du MDC correspondant au diagnostic principal.
- 3 : Age invalide (pas entre 0 et 124)
- 4 : Sexe invalide (ni 1 ni 2)
- 5 : Mode de sortie invalide

MPR : Code opératoire ICD-9-CM utilisé par le grouper

ADX : Code diagnostic ICD-9-CM utilisé par le grouper pour satisfaire le critère de groupement "diagnostic quelconque"

SDX : Code diagnostic ICD-9-CM utilisé par le grouper pour satisfaire le critère de groupement "diagnostic secondaire"

HSI : Numéro d'identification de la version du grouper

(A) : Champ alphanumérique

(Les codes diagnostiques et opératoires sont justifiés à gauche)

Outre le fichier "groupé" le programme fournit un listing-compte-rendu de l'exécution dans lequel figurent les fréquences observées des DRG dans la base de données, celles des MDC et celles des codes-retour. Ces informations permettent un premier contrôle de la base de données utilisée.

#### 4. POST-TRAITEMENT

Le post-traitement comprend plusieurs phases :

VESKA.JOB : Transformation du fichier VESKA.DAT en fichier  
(ANNEXE 3) SPSSx: VESKA.SPS

POSTTTT.JOB : Attribution des DRG corrects aux séjours  
(ANNEXE 4) comprenant des codes n'ayant pas de  
correspondants ICD-9-CM.

CHGR.JOB : Transformation du fichier de sortie du grouper  
(ANNEXE 5) (CHGR.DAT) en fichier SPSSx: CHGR.SPS

MATCH.JOB : Fusion des fichiers VESKA.SPS et CHGR.SPS  
(ANNEXE 6)



Références bibliographiques

- 1 Eggli Y., Grimm R., Paccaud F. - Transcodage des codes opératoires et diagnostiques VESKA (version 1979) en codes ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 53 p. -(Cah Rech Doc IUMSP, no 14)
- 2 Eggli Y., Grimm R., Paccaud F. - Table de transcodage des diagnostics : VESKA (version 1979) - ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 104 p. -(Cah Rech Doc IUMSP, no 20)
- 3 Eggli Y., Grimm R., Paccaud F. - Table de transcodage des opérations : VESKA (version 1979) - ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 66 p. -(Cah Rech Doc IUMSP, no 21)
- 4 Eggli Y., Grimm R., Paccaud F. - Adaptation du "Grouper" aux statistiques médicales VESKA (version 1980-1986). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 9 p. -(Cah Rech Doc IUMSP, no 22)
- 5 The Revised ICD-9-CM Diagnosis Related Groups (DRGs) Software Installation Guide (version 3.0) Fall 1985 - H.S.I.

```

PROGRAM JUSTIF(ENTREE, SORTIE, OUTPUT);

(* CE PROGRAMME COPIE UN ENREGISTREMENT DE LA STATISTIQUE MEDICALE VESKA,
EN LE DIVISANT EN 3 PARTIES DE LONGUEUR 71, 70 ET 70 RESPECTIVEMENT.
IL JUSTIFIE A GAUCHE LES CHAINES DE CARACTERES, A PART DEUX CHAINES
LE NOM ET LA VARIABLE APPELEE "RESERVE", QU'IL SUPPRIME.
IL ECRIT LES CODES DIAGNOSTIQUES ET OPERATOIRES SUR DEUX LIGNES
DIFFERENTES, CONTRAIREMENT A L'ENREGISTREMENT INITIAL OU ILS SE TROUVENT
A LA SUITE LES UNS DES AUTRES. *)

CONST BLANC=' '; LMAX=2;

VAR ENTREE, SORTIE      : TEXT;
    I, NBREDIAG, NBREOPER : INTEGER;
    MANQDIAG, MANQOPER   : BOOLEAN;
    PUISS                : ARRAY [0..LMAX] OF INTEGER;
    CONV                 : ARRAY [CHAR] OF INTEGER;

PROCEDURE INCR(VAR I: INTEGER);
BEGIN
    I:=I+1;
END; (* INCR *)

PROCEDURE LIREVAR(LONG : INTEGER; VAR NOMBRE : INTEGER;
                 VAR MANQ : BOOLEAN);

TYPE CHIFFRE = '0'..'9';

VAR EXPOS : INTEGER;
    CAR   : CHAR;
    CHIF  : CHIFFRE;
    SIGNE : -1..1;

BEGIN
    NOMBRE:=0; SIGNE:=-1;
    MANQ:=TRUE;
    EXPOS:=LONG-1;

    WHILE (EXPOS >= 0) DO
        BEGIN
            IF NOT(EOLN(ENTREE)) THEN READ(ENTREE, CAR) ELSE CAR:=BLANC;
            IF (CAR = '-') THEN SIGNE:=-1
            ELSE IF (CAR <> BLANC) THEN
                BEGIN
                    CHIF:=CAR;
                    NOMBRE:=NOMBRE + CONV[CHIF]*PUISS[EXPOS];
                    MANQ:=FALSE;
                END;
            EXPOS:=EXPOS-1;
        END;
    NOMBRE:=SIGNE*NOMBRE;

END; (* LIREVAR *)

PROCEDURE JUSTGAUCHE(LONG : INTEGER);

(* AVANT APPEL DE CETTE PROCEDURE, LE DERNIER CARACTERE LU DOIT
IMMEDIATMENT PRECEDER LA CHAINE A JUSTIFIER *)

VAR NBLANC, POS, I : INTEGER;
    (* NBLANC: NOMBRE DE BLANCS A GAUCHE DE LA CHAINE A JUSTIFIER
    POS   : POSITION EN LECTURE *)
    CAR : CHAR;

BEGIN
    NBLANC:=0;

```

```

IF NOT(EOLN(ENTREE)) THEN READ(ENTREE,CAR) ELSE CAR:=BLANC;
POS:=1;

(* DETERMINATION DU NOMBRE DE BLANCS QUI PRECEDENT LA CHAINE *)
WHILE ((POS < LONG) AND (CAR = BLANC)) DO
  BEGIN
    INCR(NBLANC);
    IF NOT(EOLN(ENTREE))
    THEN BEGIN READ(ENTREE,CAR); INCR(POS) END
    ELSE BEGIN POS:=LONG; NBLANC:=LONG-1; CAR:=BLANC END;
  END;
  (* POS = LONG = NBLANC-1 OU CAR <> BLANC *)

IF ((POS=LONG) AND (CAR=BLANC)) THEN INCR(NBLANC);

(* COPIE DE LA CHAINE A PARTIR DU PREMIER CARACTERE NON-BLANC *)
IF (CAR <> BLANC)
THEN BEGIN
  WRITE(SORTIE,CAR);
  WHILE (POS < LONG) DO
    BEGIN
      IF NOT(EOLN(ENTREE)) THEN READ(ENTREE,CAR) ELSE CAR:=BLANC;
      INCR(POS);
      WRITE(SORTIE,CAR);
      END; (* POS = LONG *)
    END; (* IF *)

  (* ADJONCTION A LA FIN DE LA CHAINE DES BLANCS QUI LA PRECEDAIENT *)
  IF (NBLANC > 0) THEN FOR I:=1 TO NBLANC DO WRITE(SORTIE,BLANC);

END; (* JUSTGAUCHE *)

PROCEDURE COPIER(LONG : INTEGER);

VAR POS : INTEGER;
    CAR : CHAR;

BEGIN
  POS:=0;

  WHILE (POS < LONG) DO
    BEGIN
      IF NOT(EOLN(ENTREE)) THEN READ(ENTREE,CAR) ELSE CAR:=BLANC;
      INCR(POS);
      WRITE(SORTIE,CAR);
      END; (* POS = LONG *)
    END; (* COPIER *)

PROCEDURE SAUTER(LONG : INTEGER);

VAR CAR : CHAR;
    I : INTEGER;

BEGIN
  FOR I:=1 TO LONG DO READ(ENTREE,CAR);
  END;

(* DEBUT DU PROGRAMME PRINCIPAL *)

BEGIN
  RESET(ENTREE); REWRITE(SORTIE);

  (* CALCUL PUISSANCES DE 10 *)
  PUISS[0]:=1;
  FOR I:=1 TO LMAX DO PUISS[I]:=PUISS[I-1]*10;

  (* CONVERSION *)
  CONV['0']:=0; CONV['1']:=1; CONV['2']:=2; CONV['3']:=3; CONV['4']:=4;
  CONV['5']:=5; CONV['6']:=6; CONV['7']:=7; CONV['8']:=8; CONV['9']:=9;

  WHILE NOT(EOF(ENTREE)) DO
    BEGIN
      (* 1ERE LIGNE *)
      COPIER(9); SAUTER(20); COPIER(7); JUSTGAUCHE(11); COPIER(6);
      JUSTGAUCHE(3); COPIER(24); SAUTER(20); COPIER(7);
      LIREVAR(2,NBREDIAG,MANQDIAG);
      IF MANQDIAG THEN NBREDIAG:=0;
      WRITE(SORTIE,NBREDIAG:2);
      LIREVAR(2,NBREOPER,MANQOPER);
      IF MANQOPER THEN NBREOPER:=0;
      WRITELN(SORTIE,NBREOPER:2);

      (* 2EME LIGNE *)
      FOR I:=1 TO NBREDIAG DO JUSTGAUCHE(7);
      FOR I:=NBREDIAG+1 TO 10 DO WRITE(SORTIE,' '); WRITELN(SORTIE);

      (* 3EME LIGNE *)
      FOR I:=1 TO NBREOPER DO JUSTGAUCHE(7);
      FOR I:=NBREOPER+1 TO 10 DO WRITE(SORTIE,' '); WRITELN(SORTIE);

      READLN(ENTREE);

      END; (* FIN DU FICHIER ENTREE *)
    END.

```

```

PROGRAM TRANSCO(VESKA,ICD9CM,CODSPEC,TABDG,TABOP,PROBLEM,OUTPUT);

(* FICHIERS D'ENTREE:
VESKA: STATISTIQUE MEDICALE VESKA APRES PASSAGE PAR "JUSTIF"
CODSPEC: CODES OPERATOIRES DONT LE TRANSCODAGE DEPEND DES CODES
DIAGNOSTIQUES ASSOCIES, ET CES CODES DIAGNOSTIQUES
TABDG: TABLE DE TRANSCODAGE DES CODES DIAGNOSTIQUES
TABOP: TABLE DE TRANSCODAGE DES CODES OPERATOIRES

FICHIERS DE SORTIE:
ICD9CM: RESULTAT DU TRANSCODAGE
PROBLEM: CODES N'AYANT PU ETRE TRANSCRIS EN CODES ICD9CM *)

CONST LONG=7; MOYEN=5; COURT=4; NBRECODE=10; LAGE=3;
MAXOP=3200; MAXDG=7600;
NOPAMYG=5; NLYMAMYG=4; NOP04=3; NOP635=2; NDG812=2;
NOP660=3; NDG813=2; NOP735=3; NDG821=2; NMAX=5;
CODERREUR=0;

TYPE LLONG=1..LONG;
LMOYEN=1..MOYEN;
LCOURT=1..COURT;
CODELONG=ARRAY [LLONG] OF CHAR;
CODEMOYEN=ARRAY [LMOYEN] OF CHAR;
CODECOURT=ARRAY [LCOURT] OF CHAR;
NCODE=1..NBRECODE;
NCODE0=0..NBRECODE;
LISTCODE=ARRAY [NCODE] OF CODECOURT;
NONNEG=0..MAXINT;

VAR VESKA,ICD9CM,CODSPEC,TABDG,TABOP,PROBLEM : TEXT;
VESDG : ARRAY [1..MAXDG] OF CODECOURT;
ICD9CMDG : ARRAY [1..MAXDG] OF CODEMOYEN;
VESOP : ARRAY [1..MAXOP] OF CODECOURT;
ICD9CMOP : ARRAY [1..MAXOP] OF CODECOURT;
DIAG,OPER: LISTCODE;
LYMAMYG: ARRAY [1..NLYMAMYG] OF CODECOURT;
OPAMYG: ARRAY [1..NOPAMYG] OF CODECOURT;
(* OPERATIONS DES AMYGDALES, TRAITEMENT SPECIAL *)
OP04: ARRAY [1..NOP04] OF CODECOURT;
DG3540: CODECOURT;
OP635: ARRAY [1..NOP635] OF CODECOURT;
DG812: ARRAY [1..NDG812] OF CODECOURT;
DG813: ARRAY [1..NDG813] OF CODECOURT;
DG821: ARRAY [1..NDG821] OF CODECOURT;
OP660: ARRAY [1..NOP660] OF CODECOURT;
OP735: ARRAY [1..NOP735] OF CODECOURT;
DIAGFICTIF: CODECOURT;
NDIAG: 1..NBRECODE; (* NOMBRE DE CODES DIAG. DANS UN ENREGISTREMENT
DE LA STATISTIQUE MED. APRES EPURATION DES
CODES "S" ET "0" *)
NOPER: 0..NBRECODE; (* IDEM POUR LES CODES OPER. *)
NDG : 0..MAXDG; (* NOMBRE DE CODES DIAG. DANS LA TABLE DE TRANSCODAGE *)
NOP : 0..MAXOP; (* NOMBRE DE CODES OPER. DANS LA TABLE DE TRANSCODAGE *)
NOENREG : NONNEG; (* NOMBRE D'ENREGISTREMENTS TRAITES *)

(*****
PROCEDURE LIRELONG(VAR FICH : TEXT; VAR CODELU : CODELONG);

VAR L: LLONG;

BEGIN
FOR L:=1 TO LONG DO IF NOT(EOLN(FICH)) THEN READ(FICH,CODELU[L])
ELSE CODELU[L]:=' ';
END; (* LIRELONG *)

(*****
PROCEDURE LIREMOYEN(VAR FICH : TEXT; VAR CODELU : CODEMOYEN);

VAR L: LMOYEN;

BEGIN
FOR L:=1 TO MOYEN DO IF NOT(EOLN(FICH)) THEN READ(FICH,CODELU[L])
ELSE CODELU[L]:=' ';
END; (* LIREMOYEN *)

(*****
PROCEDURE LIRECOURT(VAR FICH : TEXT; VAR CODELU : CODECOURT);

VAR L: LCOURT;

BEGIN
FOR L:=1 TO COURT DO IF NOT(EOLN(FICH)) THEN READ(FICH,CODELU[L])
ELSE CODELU[L]:=' ';
END; (* LIRECOURT *)

(*****
PROCEDURE STOCKERLONG(VAR CODE1 : CODELONG; CODE2 : CODELONG);

VAR L : LLONG;

BEGIN
FOR L:=1 TO LONG DO CODE1[L]:=CODE2[L];
END; (* STOCKERLONG *)

```

```
(*****)  
PROCEDURE STOCKERCOURT(VAR CODE1 : CODECOURT; CODE2 : CODECOURT);
```

10

```
VAR L : LCOURT;
```

```
BEGIN  
FOR L:=1 TO COURT DO CODE1[L]:=CODE2[L];  
END; (* STOCKERCOURT *)
```

```
(*****)  
PROCEDURE TRONQUER(VAR CODE1 : CODECOURT; CODE2 : CODELONG);
```

```
VAR L : LCOURT;
```

```
BEGIN  
FOR L:=1 TO COURT DO CODE1[L]:=CODE2[L];  
END; (* TRONQUER *)
```

```
(*****)  
PROCEDURE INITIALISATIONS;
```

```
VAR I: 1..NMAX;  
L: LCOURT;
```

```
BEGIN (* INITIALISATIONS *)  
RESET(VESKA); RESET(CODSPEC); RESET(TABDG); RESET(TABOP);  
REWRITE(ICD9CM); REWRITE(PROBLEM);
```

```
(* LIRE LA TABLE DE TRANSCODAGE DES CODES DIAGNOSTIQUES *)
```

```
NDG:=0;  
WHILE NOT(EOF(TABDG)) DO  
BEGIN  
NDG:=NDG+1;  
LIRECOURT(TABDG,VESDG[NDG]); GET(TABDG);  
LIREMOYEN(TABDG,ICD9CMDG[NDG]);  
READLN(TABDG);  
END;
```

```
(* OPERATIONS SPECIALES DES AMYGDALES *)
```

```
FOR I:=1 TO NOPAMYG DO  
BEGIN LIRECOURT(CODSPEC,OPAMYG[I]); READLN(CODSPEC) END;  
(* CODES DIAGNOSTIQUES ASSOCIES *)  
FOR I:=1 TO NLYMAMYG DO  
BEGIN LIRECOURT(CODSPEC,LYMAMYG[I]); READLN(CODSPEC) END;
```

```
(* AUTRES OPERATIONS SPECIALES *)
```

```
FOR I:=1 TO NOP04 DO  
BEGIN LIRECOURT(CODSPEC,OP04[I]); READLN(CODSPEC) END;  
LIRECOURT(CODSPEC,DG3540); READLN(CODSPEC);  
FOR I:=1 TO NOP635 DO  
BEGIN LIRECOURT(CODSPEC,OP635[I]); READLN(CODSPEC) END;  
FOR I:=1 TO NDG812 DO  
BEGIN LIRECOURT(CODSPEC,DG812[I]); READLN(CODSPEC) END;  
FOR I:=1 TO NOP660 DO  
BEGIN LIRECOURT(CODSPEC,OP660[I]); READLN(CODSPEC) END;  
FOR I:=1 TO NDG813 DO  
BEGIN LIRECOURT(CODSPEC,DG813[I]); READLN(CODSPEC) END;  
FOR I:=1 TO NOP735 DO  
BEGIN LIRECOURT(CODSPEC,OP735[I]); READLN(CODSPEC) END;  
FOR I:=1 TO NDG821 DO  
BEGIN LIRECOURT(CODSPEC,DG821[I]); READLN(CODSPEC) END;
```

```
(* LIRE LA TABLE DE TRANSCODAGE DES CODES OPERATOIRES *)
```

```
NOP:=0;  
WHILE NOT(EOF(TABOP)) DO  
BEGIN  
NOP:=NOP+1;  
LIRECOURT(TABOP,VESOP[NOP]); GET(TABOP);  
LIRECOURT(TABOP,ICD9CMOP[NOP]);  
READLN(TABOP);  
END;
```

```
FOR L:=1 TO COURT DO DIAGFICTIF[L]:='*';  
NOENREG:=0;
```

```
WRITELN(PROBLEM,' CODES DIAGNOSTIQUES ET OPERATOIRES SANS CORRESPONDANTS');  
WRITELN(PROBLEM);  
WRITELN(PROBLEM,' CODE DIAG. CODE OPER. NUMERO DU CAS');  
WRITELN(PROBLEM);
```

```
END; (* INITIALISATIONS *)
```

```
(*****)  
PROCEDURE TRAITERENR;
```

```
VAR DIAGNOSTIC,SPECIALS,ESSE,SPECIAL0,ZERO,OPSPEC,PREMIERESSAI : BOOLEAN;  
CODEVES : CODECOURT;  
N : NCODE;  
C,C1,C2 : CHAR;  
I: 1..LAGE;
```

```
(*****)  
PROCEDURE SAUTER(N: NONNEG);
```

```
VAR I: NONNEG;
```

```
BEGIN  
FOR I:=1 TO N DO IF NOT(EOLN(VESKA)) THEN GET(VESKA);  
END;
```

```

BEGIN
  CASE C OF
    ' ': CONV:=0;
    '0': CONV:=0;
    '1': CONV:=1;
    '2': CONV:=2;
    '3': CONV:=3;
    '4': CONV:=4;
    '5': CONV:=5;
    '6': CONV:=6;
    '7': CONV:=7;
    '8': CONV:=8;
    '9': CONV:=9;
  END;
END;

(*****
FUNCTION EGAL(CODE1, CODE2 : CODECOURT): BOOLEAN;
VAR L: LCOURT;
    EG: BOOLEAN;
BEGIN
  EG:=TRUE;
  FOR L:=1 TO COURT DO EG:=EG AND (CODE1[L]=CODE2[L]);
  EGAL:=EG;
END; (* EGAL *)

(*****
FUNCTION EXISTE(CODEDG : CODECOURT): BOOLEAN;
VAR I: NCODE0;
    TROUVE: BOOLEAN;
BEGIN
  TROUVE:=FALSE; I:=0;
  WHILE (NOT(TROUVE) AND (I<NDIAG)) DO
    BEGIN
      I:=I+1;
      TROUVE:=EGAL(DIAG[I],CODEDG);
    END;
  EXISTE:=TROUVE;
END; (* EXISTE *)

(*****
PROCEDURE EPURER;
VAR DIAGORIG: ARRAY [NCODE] OF CODELONG;
    CODELU: CODELONG;
    SAUTERDG: BOOLEAN;
    NDIAGSEL, NOPERSEL: NCODE0;
    N: NCODE;
(*****
FUNCTION CODES (C:CODELONG) : BOOLEAN;
BEGIN
  CODES:=(C[1]='S');
END; (* CODES *)

(*****
FUNCTION CODE000 (C:CODELONG) : BOOLEAN;
BEGIN
  CODE000:=((C[1]='0') AND (C[2]='0') AND (C[3]='0'));
END; (* CODE000 *)

(*****
PROCEDURE SUPPRS(VAR CODELU: CODELONG);
VAR L: LLONG;
BEGIN
  FOR L:=1 TO (LONG-1) DO CODELU[L]:=CODELU[L+1];
  CODELU[LLONG]:=' ';
END; (* SUPPRS *)

(*****
BEGIN (* EPURER *)
ESSE:=FALSE; ZERO:=FALSE; SPECIALS:=FALSE; SPECIAL0:=FALSE;
(* STOCKER LES DIAG. ORIGINELS EN N'ELIMINANT QUE LES CODES "000" *)
NDIAGSEL:=0;
FOR N:=1 TO NDIAG DO
  BEGIN
    LIRELONG(VESKA, CODELU);
    IF CODE000(CODELU) THEN ZERO:=TRUE
    ELSE BEGIN
      NDIAGSEL:=NDIAGSEL+1;
      STOCKERLONG(DIAGORIG[NDIAGSEL], CODELU)
    END
  END;
READLN(VESKA);
SPECIAL0:=(NDIAGSEL=0);

```

```

(* STOCKER LES CODES OPER. EN ELIMINANT LES CODES "S" ET "000" *)
NOPERSEL:=0;
FOR N:=1 TO NOPER DO
  BEGIN
    LIRELONG(VESKA, CODELU);
    IF CODES(CODELU) THEN ESSE:=TRUE
    ELSE IF CODE000(CODELU) THEN ZERO:=TRUE
    ELSE BEGIN
      NOPERSEL:=NOPERSEL+1;
      TRONQUER(OPER(NOPERSEL), CODELU)
    END
  END;
READLN(VESKA);
NOPER:=NOPERSEL;

```

```

IF SPECIAL0 THEN NDIAG:=1
ELSE BEGIN
  NDIAG:=NDIAGSEL;
  NDIAGSEL:=0;
  FOR N:=1 TO NDIAG DO
    BEGIN
      SAUTERDG:=FALSE;
      IF CODES(DIAGORIG[N])
      THEN BEGIN
        ESSE:=TRUE;
        IF (NOPERSEL>0) THEN SUPPRS(DIAGORIG[N]);
        SAUTERDG:=(NOPERSEL=0) OR (CODE000(DIAGORIG[N]));
      END;
      IF NOT(SAUTERDG)
      THEN BEGIN
        NDIAGSEL:=NDIAGSEL+1;
        TRONQUER(DIAG(NDIAGSEL), DIAGORIG[N])
      END;
    END; (* FOR *)
  SPECIALS:=(NDIAGSEL-0);
  IF SPECIALS THEN NDIAG:=1 ELSE NDIAG:=NDIAGSEL;
END;

```

```

IF SPECIALS OR SPECIAL0 THEN STOCKERCOURT(DIAG[1], DIAGFICTIF);
(* VALEUR FICTIVE PAS UTILISEE *)
END; (* EPURER *)

```

```

(*****
PROCEDURE TRANSCODE(DIAGNOSTIC : BOOLEAN; VAR CODEVES : CODECOURT);

```

```

VAR TROUVE : BOOLEAN;
    NOICD9CM : NONNEG;

```

```

(*****
PROCEDURE CHERCHERCORR(DIAGNOSTIC : BOOLEAN;
    VAR TROUVE : BOOLEAN; VAR NOICD9CM : NONNEG);

```

```

VAR NODEB, NOFIN, NOMIL : NONNEG;
    ECHEC : BOOLEAN;
    CODEDEB, CODEMIL, CODEFIN : CODECOURT;

```

```

(*****
FUNCTION CHIFFRE(C : CHAR) : BOOLEAN;

```

```

BEGIN
  CHIFFRE:=((C>='0') AND (C<='9'));
END;

```

```

(*****
FUNCTION LETTRE(C : CHAR) : BOOLEAN;

```

```

BEGIN
  LETTRE:=((C>='A') AND (C<='Z'));
END;

```

```

(*****
FUNCTION AVANT(CODE1, CODE2 : CODECOURT) : BOOLEAN;

```

```

(* ATTENTION: L'ORDRE DES CAR. EN PASCAL N'EST PAS L'ORDRE STANDARD
  ASCII *)

```

```

VAR L : LCOURT;
    C1, C2 : CHAR;

```

```

BEGIN
  L:=1;
  WHILE ((CODE1[L] = CODE2[L]) AND (L < COURT)) DO L:=L+1;
  C1:=CODE1[L]; C2:=CODE2[L];
  AVANT:=((CHIFFRE(C1) AND (LETTRE(C2) OR (CHIFFRE(C2) AND (C1<C2))))
    OR (LETTRE(C1) AND LETTRE(C2) AND (C1<C2)))
    OR ((C1=' ') AND (C2<>' ')));
END; (* AVANT *)

```

```

(*****),
BEGIN (* CHERCHERCORR *)
(* LE CODE LU EST-IL COMPRIS ENTRE LE PREMIER ET LE DERNIER CODE ? *)
NODEB:=1;
IF DIAGNOSTIC
THEN BEGIN
    STOCKERCOURT(CODEDEB,VESDG[1]);
    STOCKERCOURT(CODEFIN,VESDG[NDG]);
    NOFIN:=NDG;
    END
ELSE BEGIN
    STOCKERCOURT(CODEDEB,VESOP[1]);
    STOCKERCOURT(CODEFIN,VESOP[NOP]);
    NOFIN:=NOP;
    END;
ECHEC:=AVANT(CODEVES, CODEDEB) OR AVANT(CODEFIN, CODEVES);
TROUVE:=FALSE;

WHILE NOT(TROUVE OR ECHEC) DO
    BEGIN
        NOMIL:=TRUNC((NODEB+NOFIN)/2.0);
        IF DIAGNOSTIC THEN STOCKERCOURT(CODEMIL,VESDG[NOMIL])
        ELSE STOCKERCOURT(CODEMIL,VESOP[NOMIL]);
        IF AVANT(CODEVES, CODEMIL)
        THEN NOFIN:=NOMIL-1
        ELSE IF AVANT(CODEMIL, CODEVES) THEN NODEB:=NOMIL+1
            ELSE BEGIN TROUVE:=TRUE; NOICD9CM:=NOMIL END;
        ECHEC:=(NOFIN<NODEB);
        END;

    END; (* CHERCHERCORR *)

(*****)
PROCEDURE STOCKERCORR(NOICD9CM : NONNEG);

VAR L,LL : LMOYEN;

BEGIN
    IF (NOICD9CM=CODERREUR)
    THEN BEGIN
        WRITE(ICD9CM,'');
        IF DIAGNOSTIC THEN LL:=COURT ELSE LL:=COURT-1;
        FOR L:=1 TO LL DO WRITE(ICD9CM, CODEVES[L])
        END
    ELSE IF DIAGNOSTIC
        THEN FOR L:=1 TO MOYEN DO WRITE(ICD9CM, ICD9CMDG[NOICD9CM, L])
            ELSE FOR L:=1 TO COURT DO WRITE(ICD9CM, ICD9CMOP[NOICD9CM, L]);
    END; (* STOCKERCORR *)

(*****)
PROCEDURE MESSAGE;

VAR L : LCOURT;

BEGIN (* MESSAGE *)
    IF DIAGNOSTIC
    THEN BEGIN
        WRITE(PROBLEM, ' ');
        FOR L:=1 TO COURT DO WRITE(PROBLEM,DIAG[N,L]);
        WRITE(PROBLEM, ' ');
        END
    ELSE BEGIN
        WRITE(PROBLEM, ' ');
        FOR L:=1 TO COURT DO WRITE(PROBLEM, OPER[N,L]);
        WRITE(PROBLEM, ' ');
        END;
    WRITELN(PROBLEM, NOENREG:6);
    END; (* MESSAGE *)

(*****)
PROCEDURE TRAITOPSPEC;

VAR NOCAS: 0..9;
    AMYGDAL: BOOLEAN;
    I: 1..NMAX;

BEGIN
    NOCAS:=0; OPSPEC:=TRUE;
    (* OPERATIONS DES AMYGDALES *)
    FOR I:=1 TO NOPAMYG DO IF EGAL(CODEVES, OPAMYG[I]) THEN NOCAS:=I;
    IF (NOCAS<>0)
    THEN BEGIN

```



```

AMYGDAL:=FALSE;

FOR I:=1 TO NLYMAMYG DO AMYGDAL:=AMYGDAL OR EXISTE(LYMAMYG[I])
END
ELSE IF (EGAL(CODEVES,OP04[1]) OR EGAL(CODEVES,OP04[2]) OR
EGAL(CODEVES,OP04[3]))
THEN NOCAS:=6
ELSE IF (EGAL(CODEVES,OP635[1]) OR EGAL(CODEVES,OP635[2]))
THEN NOCAS:=7
ELSE IF (EGAL(CODEVES,OP660[1]) OR EGAL(CODEVES,OP660[2]) OR
EGAL(CODEVES,OP660[3])) THEN NOCAS:=8
ELSE IF (EGAL(CODEVES,OP735[1]) OR
EGAL(CODEVES,OP735[2]) OR EGAL(CODEVES,OP735[3]))
THEN NOCAS:=9;

CASE NOCAS OF
0: OPSPEC:=FALSE;
1,2: IF AMYGDAL THEN WRITE(ICD9CM,'2811')
ELSE WRITE(ICD9CM,'2912');
3: IF AMYGDAL THEN WRITE(ICD9CM,'2890')
ELSE WRITE(ICD9CM,'290 ');
4,5: IF AMYGDAL THEN WRITE(ICD9CM,'2899')
ELSE WRITE(ICD9CM,'2959');
6: IF EXISTE(DG3540)
THEN WRITE(ICD9CM,'0443')
ELSE WRITE(ICD9CM,'0449');
7: IF (EXISTE(DG812[1]) OR EXISTE(DG812[2]))
THEN WRITE(ICD9CM,'7941')
ELSE WRITE(ICD9CM,'7901');
8: IF (EXISTE(DG813[1]) OR EXISTE(DG813[2]))
THEN WRITE(ICD9CM,'7942')
ELSE WRITE(ICD9CM,'7902');
9: IF (EXISTE(DG821[1]) OR EXISTE(DG821[2]))
THEN WRITE(ICD9CM,'7945')
ELSE WRITE(ICD9CM,'7905');
END; (* CASE *)
END; (* TRAITOPSPEC *)

(*****
BEGIN (* TRANSCODE *)
IF DIAGNOSTIC
THEN IF SPECIAL0
THEN WRITE(ICD9CM,'00000')
ELSE IF SPECIALS
THEN WRITE(ICD9CM,'V669 ')
ELSE
ELSE TRAITOPSPEC;
TROUVE:=(DIAGNOSTIC AND (SPECIAL0 OR SPECIALS)) OR
(NOT(DIAGNOSTIC) AND OPSPEC);

IF NOT(TROUVE)
THEN BEGIN
CHERCHERCORR(DIAGNOSTIC,TROUVE,NOICD9CM);
IF TROUVE THEN STOCKERCORR(NOICD9CM)
ELSE IF PREMIERESSAI
THEN BEGIN
PREMIERESSAI:=FALSE;
CODEVES[COURT]:=' ';
TRANSCODE(DIAGNOSTIC, CODEVES)
END
ELSE BEGIN STOCKERCORR(CODERREUR); MESSAGE END;
END;
END; (* TRANSCODE *)

(*****
BEGIN (* TRAITERENR *)
NOENREG:=NOENREG+1;

```

```

(* 1ERE LIGNE *)
SAUTER(9); READ(VESKA,C); WRITE(ICD9CM,C); (* SEXE *) SAUTER(38);
FOR I:=1 TO LAGE DO BEGIN READ(VESKA,C); WRITE(ICD9CM,C) END; (* AGE *)
SAUTER(7); READ(VESKA,C); (* MODE DE SORTIE *)
IF ((C>='0') AND (C<='4')) OR (C='8') OR (C='9'))
THEN CASE C OF
  '1': BEGIN C1:='0'; C2:='1' END;
  '2': BEGIN C1:='0'; C2:='5' END;
  '3': BEGIN C1:='0'; C2:='5' END;
  '4': BEGIN C1:='0'; C2:='3' END;
  '8','9': BEGIN C1:='2'; C2:='0' END;
  '0': BEGIN C1:='3'; C2:='0' END;
END
ELSE BEGIN C1:=' '; C2:=' ' END;
WRITE(ICD9CM,C1,C2);
SAUTER(8); READ(VESKA,C1,C2); NDIAG:=10*CONV(C1)+CONV(C2);
READLN(VESKA,C1,C2); NOPER:=10*CONV(C1)+CONV(C2);

(* 2EME ET 3EME LIGNES *)
(* LIRE LES CODES DIAG. ET OPER. EN ELIMINANT LES CODES "S" ET "0" *)
EPURER;
IF ESSE THEN WRITE(ICD9CM,'S') ELSE WRITE(ICD9CM,' ');
IF ZERO THEN WRITE(ICD9CM,'Z') ELSE WRITE(ICD9CM,' ');
FOR N:=1 TO NDIAG DO
  BEGIN
    PREMIERESSAI:=TRUE;
    STOCKERCOURT(CODEVES,DIAG[N]);
    TRANSCODE(TRUE,CODEVES)
  END;
FOR N:=NDIAG+1 TO NBRECODE DO WRITE(ICD9CM,' ');
FOR N:=1 TO NOPER DO
  BEGIN
    PREMIERESSAI:=TRUE;
    STOCKERCOURT(CODEVES,OPER[N]);
    TRANSCODE(FALSE,CODEVES)
  END;
WRITELN(ICD9CM);
END; (* TRAITERENR *)

(*****
(* PROGRAMME PRINCIPAL *)
BEGIN
INITIALISATIONS;
WHILE NOT(EOF(VESKA)) DO TRAITERENR;
END.

```

### ANNEXE 3: PROGRAMME VESKAJOB

TITLE "CREATION D'UN FICHER VESKA86.SPS"  
 FILE HANDLE ENTREE/NAME-'VESKA86.DAT'  
 FILE HANDLE SORTIE/NAME-'VESKA86.SPS'  
 DATA LIST FILE-ENTREE RECORDS=3

```

/1
CANTON      1-2
HOPITAL     1-6
ANNENAISS  11-12
NUMDOSS    17-27(A)
ETATCIV     32
ANNENTR    37-38
ANNESORT   43-44
AGESORT    49-51
CODENTR    58
PREMADM    61
JOURPROP   66-67
SERVICE    7-9
MOINAISS   13-14
CONFESS    33
MOISENTR   39-40
MOISSORT   45-46
NJOUHOSP   52-56
MODESORT   59
ANNEPROP   62-63
NBREDIAG   68-69
SEXE       10
JOUNAISS   15-16
NUMPOST    28-31
NAT        34-36(A)
JOURNTR    41-42
JOURSORT   47-48
MODENTR    57
NAISMULT   60(A)
MOISPROP   64-65
NBREOPER   70-71
/2
DIAG1      1-7(A)
DIAG4      22-28(A)
DIAG7      43-49(A)
DIAG10     64-70(A)
DIAG2      8-14(A)
DIAG5      29-35(A)
DIAG8      50-56(A)
DIAG3      15-21(A)
DIAG6      36-42(A)
DIAG9      57-63(A)
/3
OPER1      1-7(A)
OPER4      22-28(A)
OPER7      43-49(A)
OPER10     64-70(A)
OPER2      8-14(A)
OPER5      29-35(A)
OPER8      50-56(A)
OPER3      15-21(A)
OPER6      36-42(A)
OPER9      57-63(A)
SAVE OUTFILE-SORTIE

```

## ANNEXE 4: PROGRAMME POSTTTT.JOB

```

TITLE "POST-TRAITEMENT"
FILE HANDLE ENTREE/NAME='CHGR86.SPS'
FILE HANDLE SORTIE/NAME='POSTT86.SPS'
GET FILE-ENTREE
STRING MD (A2)
RECODE MDC (1-'1') (2-'2') (3-'3') (4-'4') (5-'5') (6-'6')
(7-'7') (8-'8') (9-'9') (10-'10') (11-'11')
(12-'12') (13-'13') (14-'14') (15-'15') (16-'16')
(17-'17') (18-'18') (19-'19') (20-'20') (21-'21')
(22-'22') (23-'23') (24-'24') INTO MD

STRING RRRR1 TO RRRR10 (A4)
COMPUTE RRRR1-OPTR1
COMPUTE RRRR2-OPTR2
COMPUTE RRRR3-OPTR3
COMPUTE RRRR4-OPTR4
COMPUTE RRRR5-OPTR5
COMPUTE RRRR6-OPTR6
COMPUTE RRRR7-OPTR7
COMPUTE RRRR8-OPTR8
COMPUTE RRRR9-OPTR9
COMPUTE RRRR10-OPTR10
DO IF (RRRR1 EQ 'I' AND RRRR2 EQ ' ' AND RRRR3 EQ ' '
AND RRRR4 EQ ' ' AND RRRR5 EQ ' ' AND RRRR6 EQ ' '
AND RRRR7 EQ ' ' AND RRRR8 EQ ' ' AND RRRR9 EQ ' '
AND RRRR10 EQ ' ')
RECODE DRG (1 THRU 473-473)
END IF
DO IF (RRRR1 EQ 'I' AND RRRR2 EQ 'I' AND RRRR3 EQ ' '
AND RRRR4 EQ ' ' AND RRRR5 EQ ' ' AND RRRR6 EQ ' '
AND RRRR7 EQ ' ' AND RRRR8 EQ ' ' AND RRRR9 EQ ' '
AND RRRR10 EQ ' ')
RECODE DRG (1 THRU 473-473)
END IF
DO IF (RRRR1 EQ 'I' AND RRRR2 EQ 'I' AND RRRR3 EQ 'I'
AND RRRR4 EQ ' ' AND RRRR5 EQ ' ' AND RRRR6 EQ ' '
AND RRRR7 EQ ' ' AND RRRR8 EQ ' ' AND RRRR9 EQ ' '
AND RRRR10 EQ ' ')
RECODE DRG (1 THRU 473-473)
END IF
RECODE RRRR1 TO RRRR10 ('P' 'P1' 'P2' 'P3' 'P4' 'P5'
'P6' 'P7' 'P8' 'P9' 'P10' 'P11'
'P12' 'P13' 'P14' 'P15' 'P16' 'P17'
'P18' 'P19' 'P20' 'P21' 'P22' 'P23'
'I' ' ' ' ')

STRING TR1 TO TR10 (A1)
STRING OP1 TO OP10 (A2)
COMPUTE TR1-SUBSTR(RRRR1,1,1)
COMPUTE TR2-SUBSTR(RRRR2,1,1)
COMPUTE TR3-SUBSTR(RRRR3,1,1)
COMPUTE TR4-SUBSTR(RRRR4,1,1)
COMPUTE TR5-SUBSTR(RRRR5,1,1)
COMPUTE TR6-SUBSTR(RRRR6,1,1)
COMPUTE TR7-SUBSTR(RRRR7,1,1)
COMPUTE TR8-SUBSTR(RRRR8,1,1)
COMPUTE TR9-SUBSTR(RRRR9,1,1)
COMPUTE TR10-SUBSTR(RRRR10,1,1)
COMPUTE OP1-SUBSTR(RRRR1,2,2)
COMPUTE OP2-SUBSTR(RRRR2,2,2)
COMPUTE OP3-SUBSTR(RRRR3,2,2)
COMPUTE OP4-SUBSTR(RRRR4,2,2)
COMPUTE OP5-SUBSTR(RRRR5,2,2)
COMPUTE OP6-SUBSTR(RRRR6,2,2)
COMPUTE OP7-SUBSTR(RRRR7,2,2)
COMPUTE OP8-SUBSTR(RRRR8,2,2)
COMPUTE OP9-SUBSTR(RRRR9,2,2)
COMPUTE OP10-SUBSTR(RRRR10,2,2)
DO IF (TR1 EQ 'S' AND OP1 NE MD AND OP2 EQ ' ' AND OP3 EQ ' '
AND OP4 EQ ' ' AND OP5 EQ ' ' AND OP6 EQ ' ' AND OP7 EQ ' '
AND OP8 EQ ' ' AND OP9 EQ ' ' AND OP10 EQ ' ')
RECODE DRG (1 THRU 473-468)
END IF
DO IF (TR1 EQ 'S' AND TR2 EQ 'S' AND OP1 NE MD AND OP2 NE MD
AND OP3 EQ ' ' AND OP4 EQ ' ' AND OP5 EQ ' ' AND OP6 EQ ' '
AND OP7 EQ ' ' AND OP8 EQ ' ' AND OP9 EQ ' ' AND OP10 EQ ' ')
RECODE DRG (1 THRU 473-468)
END IF
DO IF (TR1 EQ 'S' AND TR2 EQ 'S' AND TR3 EQ 'S' AND OP1 NE MD AND OP2 NE MD
AND OP3 NE MD AND OP4 EQ ' ' AND OP5 EQ ' ' AND OP6 EQ ' '
AND OP7 EQ ' ' AND OP8 EQ ' ' AND OP9 EQ ' ' AND OP10 EQ ' ')
RECODE DRG (1 THRU 473-468)
END IF

```

Transformation de la variable MDC (numérique) en MD (αnumérique)

Transformation des variables OPTRi (numériques) en RRRRi (αnumérique)

Attribution du DRG 473 aux séjours ne comprenant que des codes opératoires I

Suppression des codes Pi et I

Extraction du premier caractère des chaînes de caractères RRRRi

Extraction des deux caractères suivants

Attribution du DRG 468 aux séjours ne présentant que des codes Sxx incompatibles avec le MDC xx

```

DO IF ((TR1 EQ 'S') AND (OP1 EQ MD))
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR2 EQ 'S' AND OP2 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR3 EQ 'S' AND OP3 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR4 EQ 'S' AND OP4 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR5 EQ 'S' AND OP5 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR6 EQ 'S' AND OP6 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR7 EQ 'S' AND OP7 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR8 EQ 'S' AND OP8 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR9 EQ 'S' AND OP9 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (TR10 EQ 'S' AND OP10 EQ MD)
RECODE DRG (1 THRU 473-472)
END IF
DO IF (RRRR1 EQ 'S ' OR RRRR2 EQ 'S ' OR RRRR3 EQ 'S '
OR RRRR4 EQ 'S ' OR RRRR5 EQ 'S ' OR RRRR6 EQ 'S '
OR RRRR7 EQ 'S ' OR RRRR8 EQ 'S ' OR RRRR9 EQ 'S '
OR RRRR10 EQ 'S ')
RECODE DRG (1 THRU 473-472)
END IF
DO IF (DGTR1 EQ 'D ')
RECODE DRG (1 THRU 473-469)
END IF
DO IF ((TR1 EQ 'S' OR TR2 EQ 'S' OR TR3 EQ 'S' OR TR4 EQ 'S' OR TR5 EQ 'S'
OR TR6 EQ 'S' OR TR7 EQ 'S' OR TR8 EQ 'S' OR TR9 EQ 'S' OR TR10 EQ 'S')
AND MDC EQ 23)
RECODE DRG (462 THRU 467-461)
END IF
DO IF ((TR1 EQ 'S' OR TR2 EQ 'S' OR TR3 EQ 'S' OR TR4 EQ 'S' OR TR5 EQ 'S'
OR TR6 EQ 'S' OR TR7 EQ 'S' OR TR8 EQ 'S' OR TR9 EQ 'S' OR TR10 EQ 'S')
AND MDC EQ 19)
RECODE DRG (425 THRU 432-424)
END IF
DO IF ((TR1 EQ 'S' OR TR2 EQ 'S' OR TR3 EQ 'S' OR TR4 EQ 'S' OR TR5 EQ 'S'
OR TR6 EQ 'S' OR TR7 EQ 'S' OR TR8 EQ 'S' OR TR9 EQ 'S' OR TR10 EQ 'S')
AND MDC EQ 18)
RECODE DRG (416 THRU 423-415)
END IF
DO IF (DRG EQ 470 OR DRG EQ 473)
RECODE MDC (1 THRU 23-24)
END IF
DO IF (DRG EQ 472 AND (RRRR1 EQ 'S ' OR RRRR2 EQ 'S ' OR RRRR3 EQ 'S '
OR RRRR4 EQ 'S ' OR RRRR5 EQ 'S ' OR RRRR6 EQ 'S ' OR RRRR7 EQ 'S '
OR RRRR8 EQ 'S ' OR RRRR9 EQ 'S ' OR RRRR10 EQ 'S '))
RECODE MDC (1 THRU 23-24)
END IF
SAVE OUTFILE-SORTIE/DROP=RRRR1 TO RRRR10 OP1 TO OP10
TR1 TO TR10 MD

```

Attribution du DRG 472  
aux séjours présentant  
des codes Sxx compatibles  
avec le MDC xx

Remarque: Les codes Sxx  
incompatibles avec le  
MDC xx et présent simulta-  
nément avec des codes  
ICD-9-CM ne seront pas  
pris en compte par le  
grouper

Attribution du DRG 472  
aux séjours présentant  
un code S

Attribution du DRG 469  
aux séjours ayant un  
code D pour code  
diagnostique principal

Attribution du DRG 461  
aux patients du MDC 23  
ayant subi une interven-  
tion chirurgicale S (non  
reconnu par le grouper)

Attribution du DRG 424  
aux patients du MDC 19  
ayant subi une interven-  
tion chirurgicale S

Attribution du DRG 415  
aux patients du MDC 18  
ayant subi une interven-  
tion chirurgicale S

Attribution du MDC 24  
aux séjours présentant  
des codes invalides

Seuls les séjours du  
DRG 472 présentant des  
codes S (et non pas Sxx  
pour lesquels le MDC est  
connu) sont attribués au  
MDC 24

ANNEXE 5: PROGRAMME CHGR.JOB

```

TITLE 'CREATION FICHER SPSS DES DONNES GROUPEES DE 1986'
FILE HANDLE CHGR84A/NAME='[DRG]CHGR86.DAT'
FILE HANDLE CHGR84B/NAME='[DRG]CHGR86.SPS'
DATA LIST FILE=CHGR84A RECORDS=1 /
  SEXE 1 AGE 2-4 SORTTR 5-6 CODES 7(A) CODE000 8(A)
  DGTR1 9-13(A) DGTR2 14-18(A) DGTR3 19-23(A) DGTR4 24-28(A) DGTR5 29-33(A)
  DGTR6 34-38(A) DGTR7 39-43(A) DGTR8 44-48(A) DGTR9 49-53(A) DGTR10 54-58(A)
  OPTR1 59-62 (A) OPTR2 63-66(A) OPTR3 67-70(A) OPTR4 71-74(A)
  OPTR5 75-78(A) OPTR6 79-82(A) OPTR7 83-86(A) OPTR8 87-90(A) OPTR9 91-94(A)
  OPTR10 95-98(A)
  DRG 99-101 MDC 102-103 RTC 104 MPR 105-108(A) ADX 109-113(A) SDX 114-118(A)
  HSI 119-129(A)
SAVE OUTFILE=CHGR84B/DROP=HSI

```

ANNEXE 6: PROGRAMME MATCH.JOB

```

TITLE "FUSION DES FICHERS VESKA86.SPS ET POSTT86.SPS"
FILE HANDLE VESKA86/NAME='[drg]VESKA86.SPS'
FILE HANDLE POSTT86/NAME='[drg]POSTT86.SPS'
FILE HANDLE DRG86/NAME='[drg]DRG86.SPS'
MATCH FILES FILE=VESKA86/FILE=POSTT86/MAP/
KEEP=CANTON HOPITAL SERVICE SEXE AGE ANNENAISS MOINAISS JOUNAISS NUMDOSS
  NUMPOST ETATCIV CONFESS NAT ANNENTR MOISENTR JOURENTR ANNESORT MOISSORT
  JOURSORT NJOUHOSP MODENTR CODENTR MODESORT NAISMULT PREMADM
  ANNEPROP MOISPROP JOURPROP NBREDIAG NBREOPER
  DIAG1 DIAG2 DIAG3 DIAG4 DIAG5 DIAG6 DIAG7 DIAG8 DIAG9 DIAG10
  OPER1 OPER2 OPER3 OPER4 OPER5 OPER6 OPER7 OPER8 OPER9 OPER10
  DGTR1 DGTR2 DGTR3 DGTR4 DGTR5 DGTR6 DGTR7 DGTR8 DGTR9 DGTR10
  OPTR1 OPTR2 OPTR3 OPTR4 OPTR5 OPTR6 OPTR7 OPTR8 OPTR9 OPTR10
  SORTTR CODES CODE000 DRG MDC RTC MPR ADX SDX
SAVE OUTFILE=DRG86

```

CAHIERS DE RECHERCHES ET DE DOCUMENTATION

- 1 s.1 Paccaud F., Grimm R., Gutzwiller F. - Analyse de la dotation en lits par groupes diagnostiques : exemple du service d'obstétrique dans les hôpitaux de zone. - Lausanne, Institut universitaire de médecine sociale et préventive, 1985, 15 p.
- 1 s.2 Paccaud F., Grimm R., Gutzwiller F. - Projections de la dotation en lits par groupes diagnostiques et par classes d'âges : hôpitaux de zone, 1990-2010 (version provisoire). - Lausanne, Institut universitaire de médecine sociale et préventive, 1985, 41 p.  
(Remplacé par Cah Rech Doc IUMSP no 1 s.5)
- 1 s.3 Paccaud F., Eggimann B. - Groupes diagnostiques utilisés sur SIMULIT 13. - Lausanne, Institut universitaire de médecine sociale et préventive, 1985, 13 p.
- 1 s.4 Grimm R., Paccaud F. - SIMULIT. Un modèle de simulation pour l'analyse et la planification de l'activité hospitalière. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 15 p.
- 1 s.5 Paccaud F., Grimm R., Gutzwiller F. - Projections de l'utilisation des lits dans le canton de Vaud : hôpitaux de zone, 1990-2010. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 53 p.  
(Remplace Cah Rech Doc IUMSP no 1 s.2)
- 1 s.6 Paccaud F., Eggimann B. - Groupes diagnostiques utilisés sur SIMULIT 14 (adaptation CHUV, 1ère révision). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 12 p.
- 1 s.7 Grimm R., Koehn V., Paccaud F. - Projections de l'utilisation des lits dans le canton de Vaud : CHUV, 1990 - 2010. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 34 p.
2. Eggimann B., Gutzwiller F. - Listériose : étude cas témoins en Suisse romande hiver 1984-85. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 15 p.
3. ROB 1. 2. 3. etc. (ROBETH) : Rapports techniques et documentation relatifs au progiciel d'analyse statistique robuste ROBETH-ROBSYS. - Lausanne, Institut universitaire de médecine sociale et préventive.
4. Levi F. - Survie en cas de cancer dans le canton de Vaud. Rapport statistique descriptif. Cas incidents 1974-1980. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 67 p.
5. Paccaud F., Schenker L., Patel M., Grimm R. - Etude Case Mix : une étude intercantonale des clientèles hospitalières (protocole de l'étude). - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 22 p.
6. Eggimann B., Paccaud F., Gutzwiller F. - Utilisation de la coronarographie dans la population résidente en Suisse. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 29 p.
7. Hausser D., Lehmann Ph., Gutzwiller F., Burnand B., Rickenbach M. - Evaluation de l'impact de la brochure tous ménages d'information sur le SIDA distribuée par l'OFSP. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 82 p.
8. Rickenbach M., Wietlisbach V., Berode M., Guillemin M. - La Plombémie en Suisse en 1985 : résultats de l'enquête MONICA pour les cantons de Vaud et Fribourg. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 26 p. + annexes.

9. Patel M., Burnand B., Rickenbach M., Hausser D., Gutzwiller F. - Modification du style de vie, une alternative au traitement pharmacologique lors d'hypertension modérée. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 28 p.
10. Chrzanowski R.S. - Microcomputer Model of Diffusion of New Medical Technologies. Project presented to the Faculty of the University of Texas Health Science Center at Houston, School of Public Health, in partial fulfillment of the requirements for the Degree of Master of Public Health. - Lausanne, Institut universitaire de médecine sociale et préventive, 1986, 89 p. + annexes.
11. Scheder P.-A., Junod B. - Cancer des voies aéro-digestives supérieures (VADS) et types de boissons alcooliques. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 38 p.
12. Huguenin M., Paccaud F., Gutzwiller F. - Recensement des patients dans les hôpitaux, cliniques, établissements spécialisés et homes valaisans : résultats d'une enquête en 1985. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 83 p. + annexes.
13. Van der Loos M.C. - Prévention de l'ostéoporose post-ménopausique par l'hormonothérapie substitutive : éléments d'analyse coût-bénéfice. (Th. Méd. Lausanne. 1986). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 127 p.
14. Eggli Y., Grimm R., Paccaud F. - Transcodage des codes opératoires et diagnostiques VESKA (Version 1979) en codes ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 53 p.
15. Gutzwiller F., Glasser J.H., Chrzanowski R., Paccaud F., Patel M. (Eds.) - Evaluation des technologies médicales. Assessment of medical technologies. Travaux présentés pendant le Congrès TEKMED 87. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 165 p.
- 16s.1 Lehmann Ph., Hausser D., Dubois-Arber F., Gutzwiller F. - Protocole d'évaluation de la campagne de lutte contre le SIDA de l'Office fédéral de la santé publique (OFSP). 1987-1988. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 20 p. + annexes.
- 16s.2 Lehmann Ph., Hausser D., Dubois-Arber F., Gutzwiller F. - Evaluation de la campagne de lutte contre le SIDA de l'Office fédéral de la santé publique (OFSP). 1987-1988. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 102 p.
17. Dubois-Arber F., Paccaud F., Gutzwiller F. - Epidémiologie de la stérilité. Démographie de la fécondité en Suisse. Revue des enquêtes de prévalence publiées. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 42 p.
18. Scheder P.-A. - Des usagers de médecines alternatives racontent (itinéraires thérapeutiques et conception de la santé). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 45 p.
19. Hausser D., Lehmann Ph., Dubois F., Gutzwiller F. - Evaluation des campagnes de prévention contre le SIDA en Suisse. (Rapport intermédiaire, juillet 1987). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 39 p.
20. Eggli Y., Grimm R., Paccaud F. - Table de transcodage des diagnostics : VESKA (version 1979) - ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 104 p.
21. Eggli Y., Grimm R., Paccaud F. - Table de transcodage des opérations : VESKA (version 1979) - ICD-9-CM. - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 66 p.
22. Eggli Y., Grimm R., Paccaud F. - Adaptation du "Grouper" aux statistiques médicales VESKA (1980-1986). - Lausanne, Institut universitaire de médecine sociale et préventive, 1987, 9 p.
23. Hausser D., Lehmann Ph., Dubois-Arber F., Gutzwiller F. - Evaluation des campagnes de prévention contre le SIDA en Suisse. Décembre 1987. - Lausanne, Institut universitaire de médecine sociale et préventive, 1988, 96 p.



24. Rohrer M.H. - La prévalence des douleurs du dos et de ses facteurs de risque chez les citoyens suisses convoqués à leur recrutement en Suisse romande en 1985. (Th. Méd. Lausanne, 1988). - Lausanne, Institut universitaire de médecine sociale et préventive, 1988, 51 p.
25. Grimm R., Egli Y., Koehn V. - Programmes informatiques permettant l'adaptation des DRG aux statistiques médicales VESKA. - Lausanne, Institut universitaire de médecine sociale et préventive, 1988, 19 p.