

**Setup de l'API**

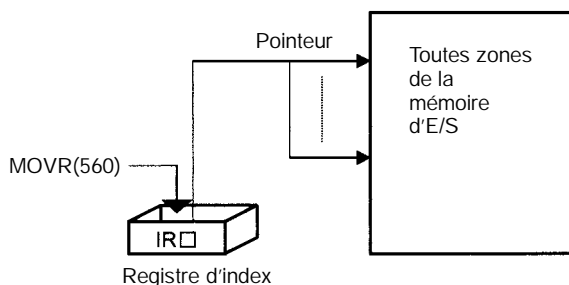
Les bits d'inhibition du rafraîchissement périodique des Unités d'E/S spéciales 0 à 95 correspondent directement aux 96 bits des adresses 226 à 231.

Adresse	Nom	Paramétrage	Par défaut
226 bit 0	Bit d'inhibition du rafraîchissement périodique pour l'Unité d'E/S spéciales 0	0 : Autorisation 1 : Inhibition	0 (Autorisation)
:	:	:	:
231 bit 15	Bit d'inhibition du rafraîchissement périodique pour l'Unité d'E/S spéciales 95	0 : Autorisation 1 : Inhibition	0 (Autorisation)

**13-2 Registres d'index**

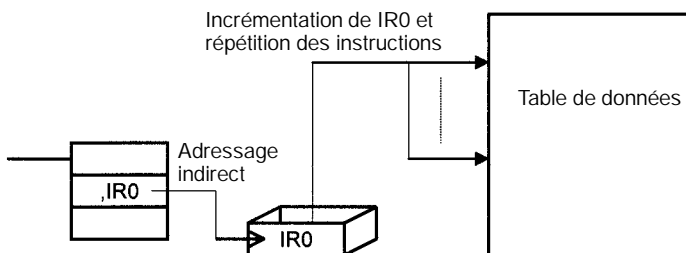
Les registres d'index fonctionnent comme des pointeurs spécifiant les adresses absolues de la mémoire d'E/S. Après rangement, par MOVR(560) ou MOVRW(561), d'une adresse mémoire de l'API dans le registre d'index, utiliser le registre d'index comme un opérande dans d'autres instructions pour adresser indirectement l'adresse mémoire de l'API.

L'avantage des registres d'index est qu'ils peuvent spécifier tout bits ou mots dans la mémoire d'E/S, y compris les temporisateurs et les valeurs courantes de compteurs.



**Utilisation des registres d'index**

Les registres d'index peuvent être un outil puissant lorsqu'ils sont combinés à des boucles de type FOR-NEXT. Le contenu des registres d'index peut être incrémenté, décrémenté et décalé très facilement. Les instructions d'une boucle peuvent ainsi traiter avec efficacité des tables ou des données consécutives.



**Fonctionnement élémentaire**

Les registres d'index sont utilisés selon les étapes suivantes :

- 1, 2, 3... 1. Utiliser MOVR(560) pour ranger dans un registre d'index l'adresse mémoire de l'API, du bit ou du mot souhaité.
2. Utiliser le registre d'index comme un operande dans presque toutes les instructions pour adresser indirectement le bit ou le mot souhaité.
3. Décaler ou incrémenter l'adresse mémoire d'origine de l'API (voir ci-après) pour rediriger le pointeur vers une autre adresse.

- Répéter les étapes 2 et 3 pour exécuter l'instruction sur d'autres numéros d'adresse.

**Décalage, incrémentation et décrémentation d'adresses**

Le tableau suivant montre les variantes possibles pour l'adressage indirect.

Variante	Syntaxe
Adressage indirect	,IRj
Adressage indirect à décalage de constante	Constante,IRj (y compris un + ou un - dans la constante).
Adressage indirect à décalage DR	DRj , IRj
Adressage indirect à incrémentation automatique	Incrémentation de 1 : ,IRj + Incrémentation de 2 : ,IRj ++
Adressage indirect à décrémentation automatique	Décrémentation de 1 : ,- IRj Décrémentation de 2 : ,- IRj

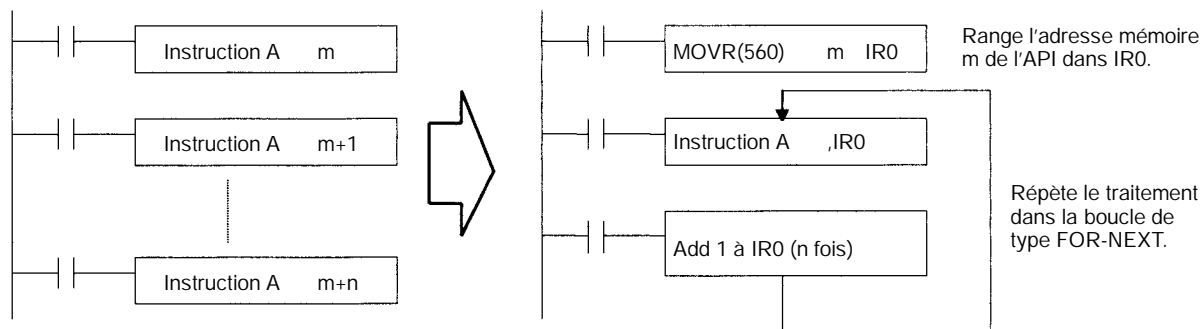
**Instructions adressant directement les registres d'index**

Les registres d'index peuvent être adressés directement par les instructions suivantes.

DOUBLE SIGNED BINARY ADD WITHOUT CARRY : +L(401), DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY : -L(411), DOUBLE INCREMENT BINARY : ++L(591), et DOUBLE DECREMENT BINARY : --L(593)

**Exemple 1**

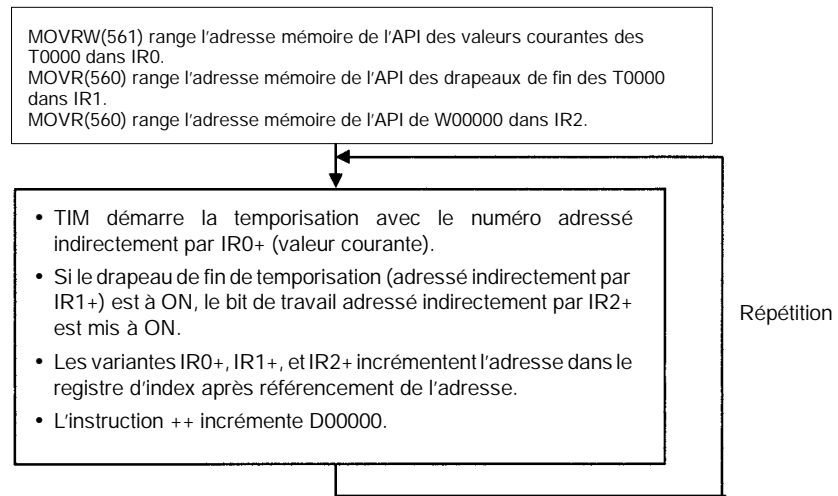
L'exemple suivant montre comment utiliser un registre d'index dans une boucle de programme, pour remplacer une longue série d'instructions. Dans ce cas, l'instruction A est répétée n+1 fois pour réaliser des opérations de lecture et de comparaisons d'une table de valeurs.



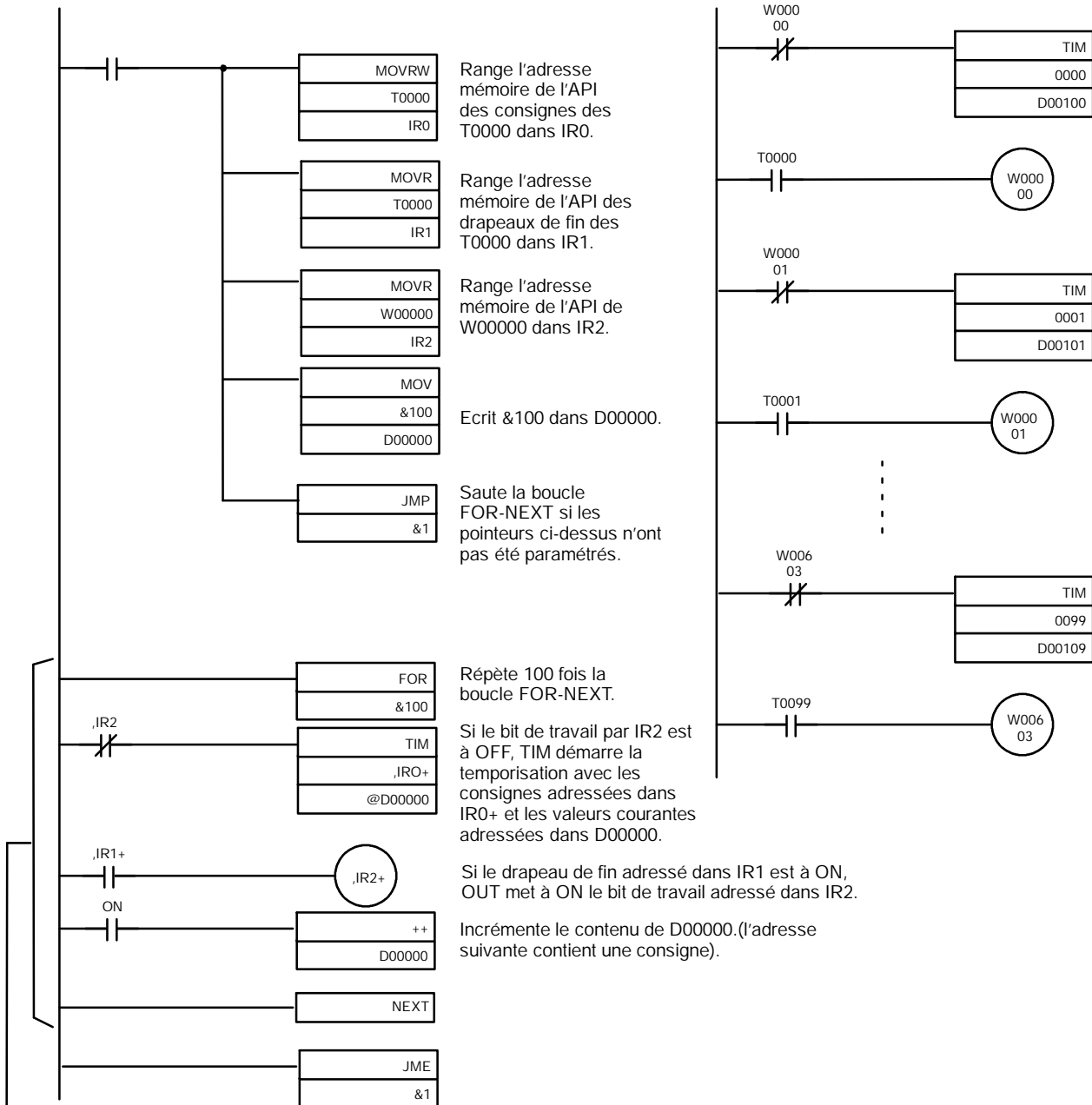
**Exemple 2**

L'exemple suivant montre comment utiliser un registre d'index dans une boucle FOR-NEXT pour définir et démarrer 100 temporisations (T0000 à T099) avec les consignes rangées de D00010 à D00109. Chaque numéro de temporisation

et drapeau de fin sont définis dans les registres d'index et la boucle est répétée à chaque incrémentation des registres d'index.



Le sous-programme à 11 instructions de la partie gauche est équivalent au sous-programme à 200 instructions de la partie droite.



La boucle FOR-NEXT démarre les temporisations de T0000 à T0099 en répétant 100 fois la boucle et en incrémentant le contenu de IR0 (numéro de temporisation/adresse de consigne), de IR1 (adresse de bit de fin), de IR2 (adresse du bit de travail) et de D00000 (adresse de consigne).

**Adressage direct des registres d'index**

Les registres d'index peuvent être adressés directement uniquement pour les instructions décrites dans le tableau suivant.

Groupe d'instruction	Nom de l'instruction	Mnémonique	Fonction primaire
Instructions de déplacement de données	MOVE TO REGISTER	MOVR(560)	Range l'adresse mémoire de l'API d'un bit ou d'un mot dans le registre d'index.
	MOVE TIMER/COUNTER PV TO REGISTER	MOVRW(561)	
Instructions de traitement des tables de données	SET RECORD LOCATION	SETR(635)	Transmet l'adresse mémoire de l'API rangée dans le registre d'index.
	GET RECORD NUMBER	GETR(636)	
Instructions de déplacement de données	DOUBLE MOVE	MOVL(498)	Transfère entre les registres d'index. Utilisé pour les échanges et les comparaisons.
	DOUBLE DATA EXCHANGE	XCGL(562)	
Instructions de comparaison	DOUBLE EQUAL	=L(301)	
	DOUBLE NOT EQUAL	L(306)	
	DOUBLE LESS THAN	L(311)	
	DOUBLE LESS THAN OR EQUAL	=L(316)	
	DOUBLE GREATER THAN	L(321)	
DOUBLE GREATER THAN OR EQUAL	=L(326)		
DOUBLE COMPARE	CMPL(060)		
Instructions d'incrémentaion/décrémentaion	DOUBLE INCREMENT BINARY	++L(591)	Modifie l'adresse mémoire de l'API dans le registre d'index par incrémentaion, décrémentaion ou décalage de son contenu.
	DOUBLE DECREMENT BINARY	--L(593)	
Instructions mathématiques symboliques	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L(401)	
	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L(411)	

**Rem.** Les instructions pour opérande de longueur double (c-à-d celles avec un "L" à la fin) sont utilisées pour les registres d'index IR0 à IR15 du fait que chaque registre contient deux mots.

**Traitements avec les registres d'index**

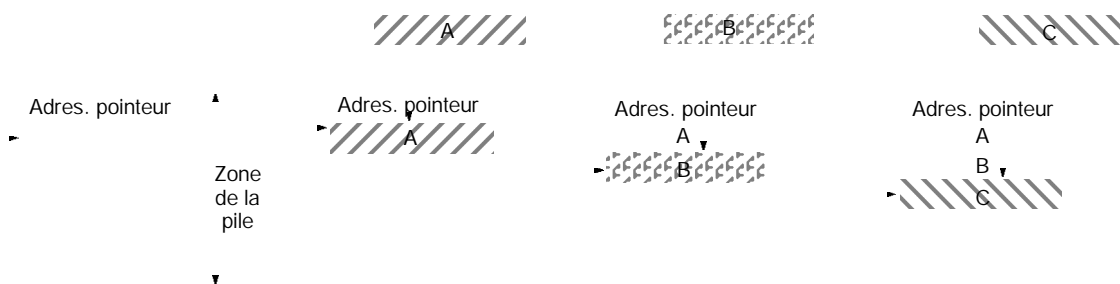
Les instructions de traitement des tables de données du CS1 complètent les fonctions des registres d'index. Ces instructions peuvent être divisées entre les instructions de traitement de pile et les instructions de traitement de tables.

Traitement			Usage	Instructions
Traitement de pile			Tables de données fonctionnant en FIFO (1er entré, 1er sorti) ou en LIFO (dernier entré, 1er sorti).	SSET(630), PUSH(632), FIFO(633), et LIFO(634)
Traitement de table	Tables à enregistrements d'un mot (Instructions de plage)	Traitement standard	Recherche de valeur comme le checksum, une valeur particulière, la valeur maximum ou minimum d'une plage.	FCS(180), SRCH(181), MAX(182), MIN(183), et SUM(184)
		Traitement spécial	Réalise d'autres traitements de tables comme des comparaisons ou des transmissions.	Association de plusieurs registres d'index par des instructions comme SRCH(181), MAX(182), MIN(183), et des instructions de comparaison.
	Tables à enregistrements de mots multiples (Instructions d'enregistrement de table)		Traitement de données dans des enregistrements ayant des longueurs de mot multiple.	Association de plusieurs registres d'index par des instructions comme DIM(631), SETR(635), GETR(636), et des instructions de comparaison.

**Traitement de pile**

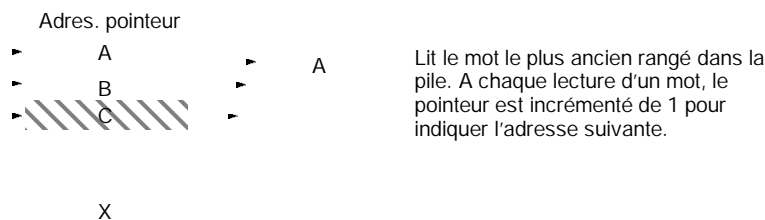
Les instructions de pile représentent des tables de données définies spécialement et appelées piles. Les données peuvent être représentées à partir d'une pile de type FIFO (premier entré, premier sorti) ou LIFO (dernier entré, premier sorti).

La pile doit être définie dans une zone particulière de la mémoire d'E/S. Les premiers mots de la pile indiquent la longueur de la pile et contiennent le pointeur de pile. Le pointeur de pile est incrémenté à chaque écriture de données dans la pile pour indiquer l'adresse de rangement de la donnée.

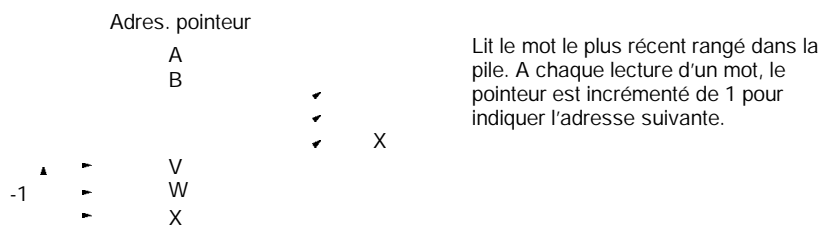


**Rem.** Actuellement, les deux premiers mots de la pile contiennent l'adresse mémoire de l'API du dernier mot de la pile et les deux mots suivants contiennent le pointeur de pile.

Le diagramme suivant montre le fonctionnement de la pile de type FIFO.



Le diagramme suivant montre le fonctionnement de la pile de type LIFO.



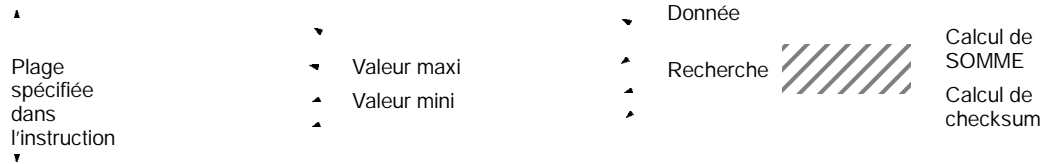
Le tableau suivant énumère les instructions de pile et leurs fonctions. Des applications typiques pour les piles sont le traitement d'informations pour des systèmes d'emmagasinage automatique ou des traitements de résultats de tests.

Instruction	Fonction
SSET(630)	Définition de la zone de pile.
PUSH(632)	Range les données dans le mot disponible suivant de la pile.
FIFO(633)	Lecture des données dans la pile de type FIFO.
LIFO(634)	Lecture des données dans la pile de type LIFO.

**Traitement de tables (Instructions de plages)**

Les instructions de plage définissent sur une plage de mots, lesquels peuvent être considérés comme table d'enregistrements à un mot. Ces instructions réalisent des opérations standard comme la recherche de valeurs minimum et maximum dans une plage, la recherche d'une valeur particulière ou le calcul du checksum.

L'adresse mémoire de l'API du mot résultat (mots contenant les valeurs mini ou maxi, la recherche de données, etc.) est automatiquement rangée dans IRO. Le registre d'index (IRO) peut être utilisé comme un opérande dans des instructions de type MOV(021) pour lire le contenu d'un mot ou réaliser d'autres traitements.



Le tableau suivant énumère les instructions de plage et leurs fonctions.

Instruction	Fonction	Description
SRCH(181)	Recherche d'une donnée.	Recherche d'une donnée dans une plage et rangement dans IRO de l'adresse mémoire de l'API du mot contenant cette donnée.
MAX(182)	Recherche d'une valeur maxi.	Recherche d'une valeur maximum dans une plage et rangement dans IRO de l'adresse mémoire de l'API du mot contenant cette valeur.
MIN(183)	Recherche d'une valeur mini.	Recherche d'une valeur minimum dans une plage et rangement dans IRO de l'adresse mémoire de l'API du mot contenant cette valeur.
SUM(184)	Calcul de somme.	Calcul de la somme de données d'une plage.
FCS(180)	Calcul de checksum.	Calcul du checksum de données d'une plage.

Les registres d'index peuvent être associés à d'autres instructions (par exemple des instructions de comparaison) dans des boucles FOR-NEXT pour réaliser des opérations complexes sur une plage de mots.

**Traitement de tables (Instructions de tables d'enregistrement)**

Les instructions de tables d'enregistrements définissent des tables de données constituées d'enregistrements d'égales longueurs. Pour un traitement facile, les enregistrements sont accessibles par leur numéro.

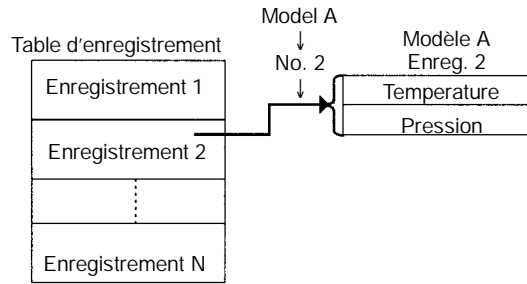
Instruction	Fonction	Description
DIM(631)	Définition d'une table d'enregistrements	Définition de la longueur et du nombre de chaque enregistrement.
SETR(635)	Paramétrage de l'emplacement d'enregistrement.	Écrit l'emplacement d'un enregistrement (l'adresse mémoire de l'API du début de l'enregistrement) dans le registre d'index spécifié.
GETR(636)	Lecture de l'emplacement d'enregistrement.	Lit le numéro d'enregistrement contenant l'adresse mémoire de l'API dans le registre d'index spécifié.

**Rem.** Les numéros d'enregistrement et les adresses de mots sont relatifs aux registres d'index. Dans l'instruction SETR(635), spécifier un numéro d'enregistrement pour ranger le début de l'enregistrement dans un registre d'index. Lorsqu'une lecture de donnée est nécessaire dans l'enregistrement, ajouter un décalage au registre d'index pour accéder à n'importe quel mot de l'enregistrement.

Utiliser les instructions de tables d'enregistrements avec des registres d'index pour les types d'instructions suivantes : lecture/écriture de données d'enregistrements, recherches d'enregistrements, lecture d'enregistrements, comparaison et calculs sur des données d'enregistrements.

Une application typique de tables d'enregistrements est la sauvegarde dans un enregistrement de données constructeur pour différents modèles d'un produit

(par exemple les réglages de température et de pression). L'accès à un modèle particulier se fait par changement du numéro d'enregistrement.



Les tables d'enregistrements sont utilisées selon les étapes suivantes :

- 1, 2, 3...**
1. Définir la structure de la table d'enregistrement par l'instruction DIM(631) et paramétrer l'adresse mémoire de l'API dans le registre d'index par SETR(635).
  2. Décaler ou incrémenter l'adresse mémoire de l'API dans le registre d'index pour lire ou comparer des mots de l'enregistrement.
  3. Décaler ou incrémenter l'adresse mémoire de l'API dans le registre d'index pour accéder à un autre enregistrement.
  4. Répéter, si nécessaire, les étapes 2 et 3.

**Exemple**

L'exemple suivant utilise les registres d'index et les instructions de tables d'enregistrements pour comparer trois valeurs aux mots 1, 3, et 5 de chaque enregistrement. Si une égalité est trouvée, le numéro d'enregistrement est rangé en D00000.

