

C463.

Joseph SULTAN

*ordinogrammes
logiques de
programmation*

DUNOD
Paris 1971

Ce fascicule est destiné à tous ceux qui, de près ou de loin, ont ou vont toucher à la programmation.

— Il n'explique aucun « LANGAGE MACHINE ».

Son seul but serait d'arriver à inculquer à tous et à toutes, débutants ou confirmés, le moyen d'élaborer correctement un ORDINOGRAMME par le biais des exercices.

— Il ne traite évidemment que d'une vingtaine de problèmes plus ou moins compliqués, mais malgré tout assez généraux.

Le programmeur, ou le futur programmeur, pourront alors y « puiser » des idées pour leurs problèmes personnels.

Sachez bien le lire.

TABLE DES MATIÈRES

Introduction	1
Les opérations ordinateurs	3
I. <i>La lecture</i>	4
II. <i>L'écriture</i>	6
III. <i>Le calcul</i>	8
IV. <i>La comparaison</i>	9
V. <i>L'ordre de mouvement</i>	11
Le raisonnement logique ordinateur	13
<i>Les problèmes de contrôles</i>	15
Notre premier problème	15
P. N° 2 : Le contrôle de séquence	20
<i>Les problèmes de totalisations</i>	24
P. N° 3 : Totalisation à un étage	24
<i>Les problèmes de fusion</i>	28
P. N° 4 : Fusion de deux fichiers	28
<i>Les problèmes de mise à jour (note)</i>	30
<i>L'aiguillage</i>	32
<i>Les problèmes d'aiguillages</i>	37
P. N° 5 : Utilisation de l'aiguillage	38
A vous de jouer	43
P. N° 6 : Contrôle de groupes	45
P. N° 7 : Totalisation à deux étages	47
P. N° 8 : Fusion de trois fichiers	50
P. N° 9 : Contrôles multiples	52
P. N° 10 : Contrôles multiples	55
P. N° 11 : Totalisation 4 quatre étages	59
P. N° 12 : Fusion de quatre fichiers (et plus...)	63

P. N° 15	170
P. N° 16	175
P. N° 17	180
P. N° 18	185

INTRODUCTION

Avons-nous le droit d'appeler « **cerveau électronique** » une machine qui par définition ne *pense pas* mais se permet malgré tout de battre nos meilleurs joueurs d'échecs sur leur propre terrain ?

Vous savez, tout comme moi, que pour *jouer* et notamment aux échecs, il faut **raisonner** et que pour *raisonner* il est nécessaire de pouvoir **penser** !

ALORS ?

Aussi paradoxal que cela puisse paraître, l'*ordinateur* peut « **raisonner sans penser** » et, de par ce fait, il applique un *raisonnement* sans *raisonner* ou, mieux encore; sans même savoir qu'il s'agit d'un raisonnement.

Des ordres sont placés dans sa mémoire et il les exécute *un à un* sans se soucier de leur *vraisemblance logique*.

— « Demandez-lui », par exemple, d'additionner « + 10 » à un nombre quelconque puis, aussitôt après, « demandez-lui » de soustraire cette même valeur à ce même nombre : il exécutera ces deux opérations sans « sourciller ».

Fidèle à l'homme, on pourrait le comparer à l'un de ces employés modèles *rigoureusement obéissants* et ne prenant **jamais de décision** sans leur chef.

Le « *chef* » pour l'ordinateur, c'est le « *programme* » ou, si vous préférez, c'est la *liste* de tous les ordres qui lui sont nécessaires pour l'accomplissement d'une tâche bien spécifique.

Une étude sévère doit être faite avant l'élaboration d'une telle liste. « Le *programmeur* » qui en a la charge doit comprendre que la plus petite erreur risque de prendre des proportions immenses aux yeux de son employeur.

Imaginez ce qu'il pourrait advenir de toute une armée si, en pleine bataille, le général la commandant se mettait à improviser et à disposer ses hommes un peu partout au hasard de ses « inspirations » ?

Un programme, c'est aussi une bataille : pour la gagner il faut l'avoir étudiée avec minutie dans ses moindres détails et surtout avoir préparé un plan où même l'imprévu aura été prévu.