

9/2.1

Le désassembleur

Qu'est-ce qu'un désassembleur ?

Comme son nom l'indique, c'est un programme qui transforme le code hexadécimal exécutable situé en mémoire en mnémoniques assembleurs. Par opposition, un programme d'assemblage ou assembleur transformera les codes mnémoniques en codes hexadécimaux exécutables.

Analyse du problème :

Comment procéder pour convertir les codes hexadécimaux en mnémoniques ?

Le micro-processeur Z80 possède de nombreux modes d'adressage. Pour clarifier les choses, établissons un tableau qui donne les codes mnémoniques correspondant à chaque code hexadécimal sur 8 bits.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC,NN	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ DEPL	LD DE,NN	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR DEPL	ADD HL,DE	LDA,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,DEPL	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LP H,n	DAA	JR Z, DEPL	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,DEPL	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR C,DEPL	ADD HL,SP	LD A,(NN)	DEC SP	INC A	DEC A	LD A,n	CCF
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
C	RET NZ	POP BC	JP NZ,ADR	JP ADR	CALL NZ,ADR	PUSH BC	ADD A,n	RST 00	RET Z	RET	JP Z,ADR	3	CALL Z,ADR	CALL ADR	ADC A,n	RST 08
D	RET NC	POP DE	JP NC,ADR	OUT (N),A	CALL NC,ADR	PUSH DE	SUB n	RST 10H	RET C	EXX	JP C,ADR	IN A,(N)	CALL C,ADR	1	SBC A,n	RST 18H
E	RET PO	POP HL	JP PO,ADR	EX (SP),HL	CALL PO,ADR	PUSH HL	AND n	RST 20H	RET PE	JP (HL)	JP PE,ADR	EX DE,HL	CALLPE,ADR	4	XOR n	RST 28H
F	RET P	POP AF	JP P,ADR	DI	CALL P,ADR	PUSH AF	OR n	RST 30H	RET M	LD SP,HL	JP M,ADR	EI	CALL M,ADR	2	CP n	RST 38H

- Une *instruction* peut être codée sur 1, 2, 3 ou 4 octets.

Codage sur un octet :

Nous avons, par exemple :

51	LD	D,C
96	SUB	(HL)

Codage sur deux octets :

Nous avons, par exemple :

20 dd	JR	NZ,dd
ED 68	LD	L,(C)

Codage sur trois octets :

Nous avons, par exemple :

DD 86 dd	ADD	A,(IX + dd)
E2 aa aa	JP	PO,aaaa

Codage sur quatre octets :

Nous avons, par exemple :

DD CB dd 7E	BIT	7,(IX + dd)
FD 36 dd nn	LD	(IY + dd),nn

- Les *mnémoniques* « simples » peuvent être codés de trois manières différentes :

1) Codage sur un octet :

Par exemple :

00	NOP
----	-----

Il suffit dans ce cas d'afficher le mnémonique.

2) Codage sur deux octets :

Par exemple :

06 nn	LD B,nn
-------	---------

Dans ce cas, il faut afficher la « **racine** » du mnémonique (le début du code) suivie de l'octet lu en mémoire.

3) Codage sur trois octets :

Par exemple :

CDnnnn	CALL nnnn
--------	-----------

Dans ce cas, il faut afficher la « **racine** » du mnémonique (le début du code) suivie des 2 octets lus en mémoire.

En ce qui concerne les mnémoniques qualifiés de « simples » dans ce qui précède, les références aux contenus sont toujours implicites (sur un registre), comme par exemple LD H,(HL), et ne font jamais appel à la mémoire, aux exceptions près suivantes :

LD	(nnnn),HL	Cas n° 5
LD	HL,(nnnn)	Cas n° 6
LD	(nnnn),A	Cas n° 7
LD	A,(nnnn)	Cas n° 8
OUT	(nnnn),A	Cas n° 9
IN	A,(nnnn)	Cas n° 10

Chacune de ces instructions fera l'objet d'un traitement particulier comme indiqué ci-dessus (cas n° 5 à cas n° 10).

Cette distinction est faite, car, dans le programme désassembleur, l'affichage du mnémonique se fera en trois temps pour ces cas particuliers :

- début du mnémonique (appelé « racine » dans ce qui précède),
- octet ou octets lu(s) en mémoire,
- fin du mnémonique.

Dans un mnémonique « simple », l'affichage se fait de la manière suivante :

- mnémonique,
- éventuellement adresse ou déplacement lu en mémoire.

Les codes CB, DD, ED et FD sont particulièrement complexes. Ils représentent chacun plusieurs dizaines de mnémoniques en fonction des codes hexadécimaux qui les suivent. Nous leur consacrerons un traitement spécial :

1 pour DD, 2 pour FD, 3 pour CB et 4 pour ED.

La liste des mnémoniques possibles pour ces 4 codes spéciaux est la suivante :

1^{er} op - code = DD

8E dd	ADC	A,(IX + dd)
86 dd	ADD	A,(IX + dd)
09	ADD	IX, BC
19	ADD	IX, DE
29	ADD	IX, IX
39	ADD	IX, SP
A6 dd	AND	(IX + dd)
CB dd 46	BIT	0, (IX + dd)
CB dd 4E	BIT	1, (IX + dd)
CB dd 56	BIT	2, (IX + dd)
CB dd 5E	BIT	3, (IX + dd)
CB dd 66	BIT	4, (IX + dd)
CB dd 6E	BIT	5, (IX + dd)
CB dd 76	BIT	6, (IX + dd)
CB dd 7E	BIT	7, (IX + dd)
BE dd	CP	(IX + dd)
35 dd	DEC	(IX + dd)
2B	DEC	IX
E3	EX	(SP),IX
34 dd	INC	(IX + dd)
23	INC	IX
E9	JP	(IX)
77 dd	LD	(IX + dd),A

70 dd	LD	(IX + dd),B
71 dd	LD	(IX + dd), C
72 dd	LD	(IX + dd), D
73 dd	LD	(IX + dd), E
74 dd	LD	(IX + dd), H
75 dd	LD	(IX + dd), L
36 dd nn	LD	(IX + dd), n
22 nn nn	LD	(nn), IX
7E dd	LD	A, (IX + dd)
46 dd	LD	B, (IX + dd)
4E dd	LD	C, (IX + dd)
56 dd	LD	D, (IX + dd)
5E dd	LD	E, (IX + dd)
66 dd	LD	H, (IX + dd)
2A nn nn	LD	IX, (nn)
21 nn nn	LD	IX, nn
6E dd	LD	L,(IX + dd)
F9	LD	SP,IX
B6 dd	OR	(IX + dd)
E1	POP	IX
E5	PUSH	IX
CB dd 86	RES	0, (IX + dd)
CB dd 8E	RES	1, (IX + dd)
CB dd 96	RES	2, (IX + dd)
CB dd 9E	RES	3, (IX + dd)
CB dd A6	RES	4, (IX + dd)
CB dd AE	RES	5, (IX + dd)
CB dd B6	RES	6, (IX + dd)
CB dd BE	RES	7, (IX + dd)
CB dd 16	RL	(IX + dd)
CB dd 06	RLC	(IX + dd)
CB dd 1E	RR	(IX + dd)
CB dd 0E	RRC	(IX + dd)
9E dd	SBC	A,(IX + dd)
CB dd C6	SET	0, (IX + dd)
CB dd CE	SET	1, (IX + dd)
CB dd D6	SET	2, (IX + dd)
CB dd DE	SET	3, (IX + dd)
CB dd E6	SET	4, (IX + dd)
CB dd EE	SET	5, (IX + dd)
CB dd F6	SET	6, (IX + dd)
CB dd FE	SET	7, (IX + dd)
CB dd 26	SLA	(IX + dd)
CB dd 2E	SRA	(IX + dd)
CB dd 3E	SRL	(IX + dd)
96 dd	SUB	(IX + dd)
AE dd	XOR	(IX + dd)

1^{er} op - code = FD

8E dd	ADC	A, (IY + dd)
86 dd	ADD	A, (IY + dd)
09	ADD	IY, BC
19	ADD	IY, DE
29	ADD	IY, IX
39	ADD	IY, SP
A6 dd	AND	(IY + dd)
CB dd 46	BIT	0, (IY + dd)
CB dd 4E	BIT	1, (IY + dd)
CB dd 56	BIT	2, (IY + dd)
CB dd 5E	BIT	3, (IY + dd)
CB dd 66	BIT	4, (IY + dd)
CB dd 6E	BIT	5, (IY + dd)
CB dd 76	BIT	6, (IY + dd)
CB dd 7E	BIT	7, (IY + dd)
BE dd	CP	(IY + dd)
35 dd	DEC	(IY + dd)
2B	DEC	IY
E3	EX	(SP), IY
34 dd	INC	(IY + dd)
23	INC	IY
E9	JP	(IY)
77 dd	LD	(IY + dd), A
70 dd	LD	(IY + dd), B
71 dd	LD	(IY + dd), C
72 dd	LD	(IY + dd), D
73 dd	LD	(IY + dd), E
74 dd	LD	(IY + dd), H
75 dd	LD	(IY + dd), L
36 dd nn	LD	(IY + dd), n
22 nn nn	LD	(nn), IY
7E dd	LD	A, (IY + dd)
46 dd	LD	B, (IY + dd)
4E dd	LD	C, (IY + dd)
56 dd	LD	D, (IY + dd)
5E dd	LD	E, (IY + dd)
66 dd	LD	H, (IY + dd)
2A nn nn	LD	IY, (nn)
21 nn nn	LD	IY, nn
6E dd	LD	L, (IY + dd)
F9	LD	SP, IY
B6 dd	OR	(IY + dd)
E1	POP	IY
E5	PUSH	IY
CB dd 86	RES	0, (IY + dd)
CB dd 8E	RES	1, (IY + dd)
CB dd 96	RES	2, (IY + dd)
CB dd 9E	RES	3, (IY + dd)
CB dd A6	RES	4, (IY + dd)

CB dd AE	RES	5, (IY + dd)
CB dd B6	RES	6, (IY + dd)
CB dd BE	RES	7, (IY + dd)
CB dd 16	RL	(IY + dd)
CB dd 06	RLC	(IY + dd)
CB dd 1E	RR	(IY + dd)
CB dd 0E	RRC	(IY + dd)
9E dd	SBC	A, (IY + dd)
CB dd C6	SET	0, (IY + dd)
CB dd CE	SET	1, (IY + dd)
CB dd D6	SET	2, (IY + dd)
CB dd DE	SET	3, (IY + dd)
CB dd E6	SET	4, (IY + dd)
CB dd EE	SET	5, (IY + dd)
CB dd F6	SET	6, (IY + dd)
CB dd FE	SET	7, (IY + dd)
CB dd 26	SLA	(IY + dd)
CB dd 2E	SRA	(IY + dd)
CB dd 3E	SRL	(IY + dd)
96 dd	SUB	(IY + dd)
AE dd	XOR	(IY + dd)

1^{er} op - code = CB

46	BIT	0, (HL)
47	BIT	0, A
40	BIT	0, B
41	BIT	0, C
42	BIT	0, D
43	BIT	0, E
44	BIT	0, H
45	BIT	0, L
4E	BIT	1, (HL)
4F	BIT	1, A
48	BIT	1, B
49	BIT	1, C
4A	BIT	1, D
4B	BIT	1, E
4C	BIT	1, H
4D	BIT	1, L
56	BIT	2, (HL)
57	BIT	2, A
50	BIT	2, B
51	BIT	2, C
52	BIT	2, D
53	BIT	2, E
54	BIT	2, H
55	BIT	2, L
5E	BIT	3, (HL)

5F	BIT	3,A
58	BIT	3,B
59	BIT	3,C
5A	BIT	3,D
5B	BIT	3,E
5C	BIT	3,H
5D	BIT	3,L
66	BIT	4,(HL)
67	BIT	4,A
60	BIT	4,B
61	BIT	4,C
62	BIT	4,D
63	BIT	4,E
64	BIT	4,H
65	BIT	4,L
6E	BIT	5,(HL)
6F	BIT	5,A
68	BIT	5,B
69	BIT	5,C
6A	BIT	5,D
6B	BIT	5,E
6C	BIT	5,H
6D	BIT	5,L
76	BIT	6,(HL)
77	BIT	6,A
70	BIT	6,B
71	BIT	6,C
72	BIT	6,D
73	BIT	6,E
74	BIT	6,H
75	BIT	6,L
7E	BIT	7,(HL)
7F	BIT	7,A
78	BIT	7,B
79	BIT	7,C
7A	BIT	7,D
7B	BIT	7,E
7C	BIT	7,H
7D	BIT	7,L
86	RES	0,(HL)
87	RES	0,A
80	RES	0,B
81	RES	0,C
82	RES	0,D
83	RES	0,E
84	RES	0,H
85	RES	0,L
8E	RES	1,(HL)

8F	RES	1,A
88	RES	1,B
89	RES	1,C
8A	RES	1,D
8B	RES	1,E
8C	RES	1,H
8D	RES	1,L
96	RES	2,(HL)
97	RES	2,A
90	RES	2,B
91	RES	2,C
92	RES	2,D
93	RES	2,E
94	RES	2,H
95	RES	2,L
9E	RES	3,(HL)
9F	RES	3,A
98	RES	3,B
99	RES	3,C
9A	RES	3,D
9B	RES	3,E
9C	RES	3,H
9D	RES	3,L
A6	RES	4,(HL)
A7	RES	4,A
A0	RES	4,B
A1	RES	4,C
A2	RES	4,D
A3	RES	4,E
A4	RES	4,H
A5	RES	4,L
AE	RES	5,(HL)
AF	RES	5,A
A8	RES	5,B
A9	RES	5,C
AA	RES	5,D
AB	RES	5,E
AC	RES	5,H
AD	RES	5,L
B6	RES	6,(HL)
B7	RES	6,A
B0	RES	6,B
B1	RES	6,C
B2	RES	6,D
B3	RES	6,E
B4	RES	6,H
B5	RES	6,L
BE	RES	7,(HL)
BF	RES	7,A
B8	RES	7,B

B9	RES	7,C
BA	RES	7,D
BB	RES	7,E
BC	RES	7,H
BD	RES	7,L
16	RL	(HL)
17	RL	A
10	RL	B
11	RL	C
12	RL	D
13	RL	E
14	RL	H
15	RL	L
06	RLC	(HL)
07	RLC	A
00	RLC	B
01	RLC	C
02	RLC	D
03	RLC	E
04	RLC	H
05	RLC	L
1E	RR	(HL)
1F	RR	A
18	RR	B
19	RR	C
1A	RR	D
1B	RR	E
1C	RR	H
1D	RR	L
0E	RRC	(HL)
0F	RRC	A
08	RRC	B
09	RRC	C
0A	RRC	D
0B	RRC	E
0C	RRC	H
0D	RRC	L
C6	SET	0,(HL)
C7	SET	0,A
C0	SET	0,B
C1	SET	0,C
C2	SET	0,D
C3	SET	0,E
C4	SET	0,H
C5	SET	0,L
CE	SET	1,(HL)
CF	SET	1,A
C8	SET	1,B
C9	SET	1,C
CA	SET	1,D

CB	SET	1,E
CC	SET	1,H
CD	SET	1,L
D6	SET	2,(HL)
D7	SET	2,A
D0	SET	2,B
D1	SET	2,C
D2	SET	2,D
D3	SET	2,E
D4	SET	2,H
D5	SET	2,L
DE	SET	3,(HL)
DF	SET	3,A
D8	SET	3,B
D9	SET	3,C
DA	SET	3,D
DB	SET	3,E
DC	SET	3,H
DD	SET	3,L
E6	SET	4,(HL)
E7	SET	4,A
E0	SET	4,B
E1	SET	4,C
E2	SET	4,D
E3	SET	4,E
E4	SET	4,H
E5	SET	4,L
EE	SET	5,(HL)
EF	SET	5,A
E8	SET	5,B
E9	SET	5,C
EA	SET	5,D
EB	SET	5,E
EC	SET	5,H
ED	SET	5,L
F6	SET	6,(HL)
F7	SET	6,A
F0	SET	6,B
F1	SET	6,C
F2	SET	6,D
F3	SET	6,E
F4	SET	6,H
F5	SET	6,L
FE	SET	7,(HL)
FF	SET	7,A
F8	SET	7,B
F9	SET	7,C
FA	SET	7,D
FB	SET	7,E
FC	SET	7,H
FD	SET	7,L

26	SLA	(HL)
27	SLA	A
20	SLA	B
21	SLA	C
22	SLA	D
23	SLA	E
24	SLA	H
25	SLA	L
2E	SRA	(HL)
2F	SRA	A
28	SRA	B
29	SRA	C
2A	SRA	D
2B	SRA	E
2C	SRA	H
2D	SRA	L
3E	SRL	(HL)
3F	SRL	A
38	SRL	B
39	SRL	C
3A	SRL	D
3B	SRL	E
3C	SRL	H
3D	SRL	L

1^{er} op - code = ED

4A	ADC	HL, BC
5A	ADC	HL, DE
6A	ADC	HL, HL
7A	ADC	HL, SP
A9	CPD	
B9	CP	
	DR	
A1	CPI	
B1	CPIR	
46	IM	0
56	IM	1
5E	IM	2
78	IN	A,(C)
40	IN	B,(C)
48	IN	C,(C)
50	IN	D,(C)
58	IN	E,(C)
60	IN	H,(C)
68	IN	L,(C)
AA	IND	
BA	INDR	
A2	INI	
B2	INIR	

43	nn nn	LD	(nn), BC
53	nn nn	LD	(nn), DE
73		LD	(nn), SP
57		LD	A, I
5F		LD	A, R
4B	nn nn	LD	BC, (nn)
5B	nn nn	LD	DE, (nn)
47		LD	I, A
4F		LD	R, A
7B	nn nn	LD	SP, (nn)
A8		LDD	
B8		LDDR	
A0		LDI	
B0		LDIR	
44		NEG	
BB		OTDR	
B3		OTIR	
79		OUT	(C), A
41		OUT	(C), B
49		OUT	(C), C
51		OUT	(C), D
59		OUT	(C), E
61		OUT	(C), H
69		OUT	(C), L
AB		OUTD	
A3		OUTI	
4D		RETI	
45		RETN	
6F		RLD	
67		RRD	
42		SBC	HL, BC
52		SBC	HL, DE
62		SBC	HL, HL
72		SBC	HL, SP

Remarques :

Dans un op-code,

→ n représente un nombre sur 8 bits (2 digit)

→ dd représente un déplacement sur 8 bits (2 digit)

→ nn représente un nombre sur 16 bits (4 digit)

→ quand des codes n, dd ou nn apparaissent dans le code opératoire et pas dans le code hexadécimal correspondant, ils suivent le dernier code hexadécimal donné.

Par exemple :

LD BC, (nn) est codé : ED 4B nn nn

Les données de type 1 et 2 peuvent être divisées en trois sous-types en fonction du nombre d'octets supplémentaires nécessaires pour le codage complet.

1°) Un octet supplémentaire.

Par exemple, DD 09 pour ADD IX,BC.

2°) Deux octets supplémentaires.

Par exemple, DD 8E dd pour ADC A,(IX + dd).

3°) Quatre octets supplémentaires.

Par exemple, DD CB dd EE pour SET 5,(IX + dd).

Les données du type 3 sont toutes adressées sur deux octets. Par exemple, CB D6 pour SET 2,(HL).

Les données du type 4 peuvent être divisées en deux sous-types selon le nombre d'octets supplémentaires nécessaires pour le codage complet.

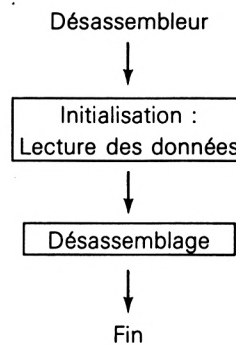
1°) Un octet supplémentaire.

Par exemple, ED 4A pour ADC HL,BC.

2°) Trois octets supplémentaires.

Par exemple, ED 43 nn nn pour LD (nnnn),BC.

Le programme que nous allons réaliser aura la structure suivante :

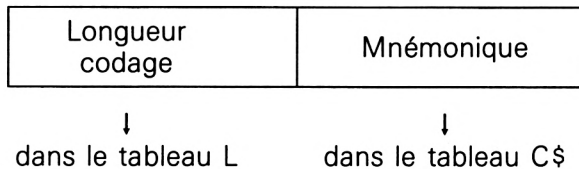


La phase d'initialisation mettra les codes opératoires du Z80 dans des tableaux pour faciliter leur manipulation et augmenter la vitesse d'affichage à l'écran. Nous distinguerons :

- a) les mnémoniques accessibles directement,
- b) les mnémoniques de type 1 (1^{er} octet = DD),
- c) les mnémoniques de type 2 (1^{er} octet = FD),
- d) les mnémoniques de type 3 (1^{er} octet = CB),
- e) les mnémoniques de type 4 (1^{er} octet = ED).

Les mnémoniques seront codés dans des lignes de DATA comme suit.

- a) Les mnémoniques accédés directement seront codés sur deux octets :



b et c) Les mnémoniques de type 1 ou 2 seront définis de trois manières différentes selon le nombre d'octets nécessaires pour qu'ils soient entièrement codés :

1 octet :

Code hexa de l'octet supplémentaire	Mnémonique
-------------------------------------	------------

2 octets :

Code hexa du 1 ^{er} octet supplémentaire	Début du mnémonique	Fin du mnémonique
---	---------------------	-------------------

4 octets :

Code hexa du 1 ^{er} octet supplémentaire	Code hexa du 3 ^e octet supplémentaire	Début du mnémonique	Fin du mnémonique
---	--	---------------------	-------------------

d) Les mnémoniques du type 3 sont définis par deux données :

Code hexa supplémentaire	Mnémonique
--------------------------	------------

e) Les mnémoniques du type 4 sont définis de deux manières différentes selon que le code-op. est en une ou deux parties :

— en une partie :

Code hexa supplémentaire	Mnémonique
--------------------------	------------

— en deux parties :

Code hexa supplémentaire	Début du mnémonique	Fin du mnémonique
--------------------------	---------------------	-------------------

La phase désassemblage consiste à :

- lire le code hexadécimal en mémoire,
- voir s'il s'agit d'un code spécial :
- dans ce cas, activer un sous-programme de traitement spécial,

- dans le cas contraire, afficher le mnémonique correspondant.

Les mnémoniques spéciaux sont repérés dans le programme dans la partie « codes opératoires et données d'adressage » par un mnémonique numérique. Ce nombre définit le numéro du traitement spécial à effectuer.

Le programme BASIC du désassembleur est le suivant :

```

10 GOSUB 3000 'Initialisation
20 GOSUB 4000 'Desassemblage
30 END

1000 REM =====
1010 REM          Codes-operatoires et donnees d'adressage
1020 REM          Codes accessibles directement
1030 REM =====
1040 DATA 1,NOP,3,"LD BC,",1,"LD (BC),A",1,INC BC,1,INC B,1,DEC B,2,"LD B,",1,RL
CA
1050 DATA 1,"EX AF,AF",1,"ADD HL,BC",1,"LD A,(BC)",1,DEC BC,1,INC C,1,DEC C,2,"
LD C,",1,RRCA
1060 DATA 3,EJNZ ,3,"LD DE,",1,"LD (DE),A",1,INC DE,1,INC D,1,DEC D,2,"LD D,",1,
RL A
1070 DATA 2,JR ,1,"ADD HL,DE",1,"LD A,(DE)",1,DEC DE,1,INC E,1,DEC E,2,"LD E,",1
,RRA
1080 DATA 2,"JR NZ,",3,"LD HL,",3,"5",1,INC HL,1,INC H,1,DEC H,2,"LD H,",1,DAA
1090 DATA 2,"JR Z,",1,"ADD HL,HL",3,"6",1,DEC HL,1,INC L,1,DEC L,2,"LD L,",1,CPL
1100 DATA 2,"JR NC,",3,"LD SP,",3,"7",1,INC SP,1,INC(HL),1,DEC(HL),2,"LD (HL),",
1,SCF
1110 DATA 2,"JR C,",1,"ADD HL,SP",3,"8",1,DEC SP,1,INC A,1,DEC A,2,"LD A,",1,CCF
1120 DATA 1,"LD B,B",1,"LD B,C",1,"LD B,D",1,"LD B,E",1,"LD B,H",1,"LD B,L",1,"L
D B,(HL)",1,"LD B,A"
1130 DATA 1,"LD C,B",1,"LD C,C",1,"LD C,D",1,"LD C,E",1,"LD C,H",1,"LD C,L",1,"L
D C,(HL)",1,"LD C,A"
1140 DATA 1,"LD D,B",1,"LD D,C",1,"LD D,D",1,"LD D,E",1,"LD D,H",1,"LD D,L",1,"L
D D,(HL)",1,"LD D,A"
1150 DATA 1,"LD E,B",1,"LD E,C",1,"LD E,D",1,"LD E,E",1,"LD E,H",1,"LD E,L",1,"L
D E,(HL)",1,"LD E,A"
1160 DATA 1,"LD H,B",1,"LD H,C",1,"LD H,D",1,"LD H,E",1,"LD H,H",1,"LD H,L",1,"L
D H,(HL)",1,"LD H,A"
1170 DATA 1,"LD L,B",1,"LD L,C",1,"LD L,D",1,"LD L,E",1,"LD L,H",1,"LD L,L",1,"L
D L,(HL)",1,"LD L,A"
1180 DATA 1,"LD (HL),B",1,"LD (HL),C",1,"LD (HL),D",1,"LD (HL),E",1,"LD (HL),H",
1,"LD (HL),L",1,HALT,1,"LD (HL),A"

```

```

1190 DATA 1,"LD A,B",1,"LD A,C",1,"LD A,D",1,"LD A,E",1,"LD A,H",1,"LD A,L",1,"LD
D A,(HL)",1,"LD A,A"

1200 DATA 1,"ADD A,B",1,"ADD A,C",1,"ADD A,D",1,"ADD A,E",1,"ADD A,H",1,"ADD A,L
",1,"ADD A,(HL)",1,"ADD A,A"

1210 DATA 1,"ADC A,B",1,"ADC A,C",1,"ADC A,D",1,"ADC A,E",1,"ADC A,H",1,"ADC A,L
",1,"ADC A,(HL)",1,"ADC A,A"

1220 DATA 1,SUB B,1,SUB C,1,SUB D,1,SUB E,1,SUB H,1,SUB L,1,SUB (HL),1,SUB A

1230 DATA 1,"SBC A,B",1,"SBC A,C",1,"SBC A,D",1,"SBC A,E",1,"SBC A,H",1,"SBC A,L
",1,"SBC A,(HL)",1,"SBC A,A"

1240 DATA 1,AND B,1,AND C,1,AND D,1,AND E,1,AND H,1,AND L,1,AND (HL),1,AND A

1250 DATA 1,XOR B,1,XOR C,1,XOR D,1,XOR E,1,XOR H,1,XOR L,1,XOR (HL),1,XOR A

1260 DATA 1,OR B,1,OR C,1,OR D,1,OR E,1,OR H,1,OR L,1,OR (HL),1,OR A

1270 DATA 1,CP B,1,CP C,1,CP D,1,CP E,1,CP H,1,CP L,1,CP (HL),1,CP A

1280 DATA 1,RET NZ,1,POP BC,2,"JP NZ,",3,"JP ",3,"CALL NZ,",1,PUSH BC,2,"ADD A,"
,1,RST 00

1290 DATA 1,RET Z,1,RET,3,"JP Z,",255,3,3,"CALL Z,",3,CALL ,2,"ADC A,",1,RST 0B

1300 DATA 1,RET NC,1,POP DE,3,"JP NC,",2,"9",3,"CALL NC,",1,PUSH DE,2,SUB ,1,RST
10H

1310 DATA 1,RET C,1,EXX,3,"JP C,",2,"10",3,"CALL C,",255,1,2,"SBC A,",1,RST 18H

1320 DATA 1,RET PO,1,POP HL,3,"JP PO,",1,"EX (SP),HL",3,"CALL PO,",1,"PUSH HL",2
,AND ,1,RST 20H

1330 DATA 1,RET PE,1,JP (HL),3,"JP PE,",1,"EX DE,HL",3,"CALL PE,",255,4,2,XOR ,1
,RST 28H

1340 DATA 1,RET P,1,POP AF,3,"JP P,",1,DI,3,"CALL P,",1,PUSH AF,2,OR ,1,RST 30H

1350 DATA 1,RET M,1,"LD SP,HL",3,"JP M,",1,EI,3,"CALL M,",255,2,2,CP ,1,RST 38H

1360 REM -----

1370 REM                               Codes de type 1

1380 REM -----

1390 DATA 9,"ADD IX,BC",&19,"ADD IX,DE",&29,"ADD IX,IX",&39,"ADD IX,SP",&2B,DEC
IX,&E3,"EX (SP),IX",&23,INC IX,&E9,JP (IX),&E1,POP I
X,&E5,PUSH IX

1400 DATA &BE,"ADC A,(IX+,)",&86,"ADD A,(IX+,)",&A6,AND (IX+,),&BE,CP (IX+,),&35
,DEC (IX+,),&34,INC (IX+,),&77,LD (IX+,),A",&70,LD
(IX+,),B"

1410 DATA &71,LD (IX+,),C",&72,LD (IX+,),D",&73,LD (IX+,),E",&74,LD (IX+,),H
",&75,LD (IX+,),L",&7E,"LD A,(IX+,)",&46,"LD B,(IX+
",),&56,"LD C,(IX+,)

```

```

1420 DATA &5E,"LD D,(IX+,),&5E,"LD E,(IX+,),&66,"LD H,(IX+,),&6E,"LD L,(IX+,
),&B6,OR (IX+,),&9E,"SBC A,(IX+,),&96,SUB (IX+,),&A

E,XOR (IX+,)

1430 DATA &CB,&46,"BIT 0,(IX+,),&CB,&4E,"BIT 1,(IX+,),&CB,&56,"BIT 2,(IX+,),&
CB,&5E,"BIT 3,(IX+,),&CB,&66,"BIT 4,(IX+,),&CB,&6E
,"BIT 5,(IX+,)

1440 DATA &CB,&76,"BIT 6,(IX+,),&CB,&7E,"BIT 7,(IX+,),&CB,&86,"RES 0,(IX+,),&
CB,&8E,"RES 1,(IX+,),&CB,&96,"RES 2,(IX+,),&CB,&9E
,"RES 3,(IX+,)

1450 DATA &CB,&A6,"RES 4,(IX+,),&CB,&AE,"RES 5,(IX+,),&CB,&B6,"RES 6,(IX+,),&
CB,&BE,"RES 7,(IX+,),&CB,&16,RL (IX+,),&CB,&6,RLC (
IX+,)

1460 DATA &CB,&1E,RR (IX+,),&CB,&E,RRC (IX+,),&CB,&C6,"SET 0,(IX+,),&CB,&CE,"SE
T 1,(IX+,),&CB,&D6,"SET 2,(IX+,),&CB,&DE,"SET 3,(I
X+,),&CB,&E6,"SET 4,(IX+,)

1470 DATA &CB,&EE,"SET 5,(IX+,),&CB,&F6,"SET 6,(IX+,),&CB,&FE,"SET 7,(IX+,),&
CB,&26,SLA (IX+,),&CB,&2E,SRA (IX+,)

1480 REM -----
1490 REM                               Codes de type 2
1500 REM -----

1510 DATA &8E,"ADC A,(IY+,),&86,"ADD A,(IY+,),9,"ADD IY,BC",&19,"ADD IY,DE",&29,"A
DD IY,IX",&39,"ADD IY,SP",&2B,DEC IY,&E3,"EX (SP),IY
",&23,INC IY,&E9,JP (IY)

1520 DATA &F9,"LD SP,IY",&E1,POP IY,&E5,PUSH IY

1530 DATA &8E,"ADC A,(IY+,),&86,"ADD A,(IY+,),&A6,"AND (IY+,),&BE,CP (IY+,),&
35,DEC (IY+,),&34,INC (IY+,),&77,LD (IY+,),A",&70,L
D (IY+,),B"

1540 DATA &71,LD (IY+,),C",&72,LD (IY+,),D",&73,LD (IY+,),E",&74,LD (IY+,),H
",&75,LD (IY+,),L"

1550 DATA &CB,&86,"RES 0,(IY+,),&CB,&8E,"RES 1,(IY+,),&CB,&96,"RES 2,(IY+,),&
CB,&9E,"RES 3,(IY+,),&CB,&A6,"RES 4,(IY+,),&CB,&AE
,"RES 5,(IY+,)

1560 DATA &CB,&B6,"RES 6,(IY+,),&CB,&BE,"RES 7,(IY+,),&CB,&16,RL (IY+,),&CB,&6,
RLC (IY+,),&CB,&1E,&RR (IY+,),&CB,&E,&RRC (IY+,),&CB
,&C6,"SET 0,(IY+,)

1570 DATA &CB,&CE,"SET 1,(IY+,),&CB,&D6,"SET 2,(IY+,),&CB,&DE,"SET 3,(IY+,),&
CB,&E6,"SET 4,(IY+,),&CB,&EE,"SET 5,(IY+,),&CB,&F6
,"SET 6,(IY+,),&CB,&FF,"SET 7,(IY+,)

```

```

1580 DATA &CB,&26,SLA (IY+,&CB,&2E,SRA (IY+,&CB,&3E,SRL (IY+,&
1590 REM -----
1600 REM                               Codes de type 3
1610 REM -----
1620 DATA &46,"BIT 0,(HL)",&47,"BIT 0,A",&40,"BIT 0,B",&41,"BIT 0,C",&42,"BIT 0,
D",&43,"BIT 0,E",&44,"BIT 0,H",&45,"BIT 0,L"
1630 DATA &4E,"BIT 1,(HL)",&4F,"BIT 1,A",&48,"BIT 1,B",&49,"BIT 1,C",&4A,"BIT 1,
D",&4B,"BIT 1,E",&4C,"BIT 1,H",&4D,"BIT 1,L"
1640 DATA &56,"BIT 2,(HL)",&57,"BIT 2,A",&50,"BIT 2,B",&51,"BIT 2,C",&52,"BIT 2,
D",&53,"BIT 2,E",&54,"BIT 2,H",&55,"BIT 2,L"
1650 DATA &5E,"BIT 3,(HL)",&5F,"BIT 3,A",&58,"BIT 3,B",&59,"BIT 3,C",&5A,"BIT 3,
D",&5B,"BIT 3,E",&5C,"BIT 3,H",&5D,"BIT 3,L"
1660 DATA &66,"BIT 4,(HL)",&67,"BIT 4,A",&60,"BIT 4,B",&61,"BIT 4,C",&62,"BIT 4,
D",&63,"BIT 4,E",&64,"BIT 4,H",&65,"BIT 4,L"
1670 DATA &6E,"BIT 5,(HL)",&6F,"BIT 5,A",&68,"BIT 5,B",&69,"BIT 5,C",&6A,"BIT 5,
D",&6B,"BIT 5,E",&6C,"BIT 5,H",&6D,"BIT 5,L"
1680 DATA &76,"BIT 6,(HL)",&77,"BIT 6,A",&70,"BIT 6,B",&71,"BIT 6,C",&72,"BIT 6,
D",&73,"BIT 6,E",&74,"BIT 6,H",&75,"BIT 6,L"
1690 DATA &7E,"BIT 7,(HL)",&7F,"BIT 7,A",&78,"BIT 7,B",&79,"BIT 7,C",&7A,"BIT 7,
D",&7B,"BIT 7,E",&7C,"BIT 7,H",&7D,"BIT 7,L"
1700 DATA &86,"RES 0,(HL)",&87,"RES 0,A",&80,"RES 0,B",&81,"RES 0,C",&82,"RES 0,
D",&83,"RES 0,E",&84,"RES 0,H",&85,"RES 0,L"
1710 DATA &8E,"RES 1,(HL)",&8F,"RES 1,A",&88,"RES 1,B",&89,"RES 1,C",&8A,"RES 1,
D",&8B,"RES 1,E",&8C,"RES 1,H",&8D,"RES 1,L"
1720 DATA &96,"RES 2,(HL)",&97,"RES 2,A",&90,"RES 2,B",&91,"RES 2,C",&92,"RES 2,
D",&93,"RES 2,E",&94,"RES 2,H",&95,"RES 2,L"
1730 DATA &9E,"RES 3,(HL)",&9F,"RES 3,A",&98,"RES 3,B",&99,"RES 3,C",&9A,"RES 3,
D",&9B,"RES 3,E",&9C,"RES 3,H",&9D,"RES 3,L"
1740 DATA &A6,"RES 4,(HL)",&A7,"RES 4,A",&A0,"RES 4,B",&A1,"RES 4,C",&A2,"RES 4,
D",&A3,"RES 4,E",&A4,"RES 4,H",&A5,"RES 4,L"
1750 DATA &AE,"RES 5,(HL)",&AF,"RES 5,A",&AB,"RES 5,B",&A9,"RES 5,C",&AA,"RES 5,
D",&AB,"RES 5,E",&AC,"RES 5,H",&AD,"RES 5,L"
1760 DATA &B6,"RES 6,(HL)",&B7,"RES 6,A",&B0,"RES 6,B",&B1,"RES 6,C",&B2,"RES 6,
D",&B3,"RES 6,E",&B4,"RES 6,H",&B5,"RES 6,L"
1770 DATA &BE,"RES 7,(HL)",&BF,"RES 7,A",&BB,"RES 7,B",&B9,"RES 7,C",&BA,"RES 7,
D",&BB,"RES 7,E",&BC,"RES 7,H",&BD85,"RES 7,L"
1780 DATA &16,RL (HL),&17,RL A,&10,RL B,&11,RL C,&12,RL D,&13,RL E,&14,RL H,&15,
RL L
1790 DATA 6,RLC (HL),7,RLC A,0,RLC B,1,RLC C,2,RLC D,3,RLC E,4,RLC H,5,RLC L
1800 DATA &1E,RR (HL),&1F,RR A,&18,RR B,&19,RR C,&1A,RR D,&1B,RR E,&1C,RR H,&1D,
RR L

```

```

1810 DATA &E,RRC (HL),&F,RRC A,B,RRC B,9,RRC C,&A,RRC D,&B,RRC E,&C,RRC H,&D,RRC
L
1820 DATA &C6,"SET 0,(HL)",&C7,"SET 0,A",&C8,"SET 0,B",&C1,"SET 0,C",&C2,"SET 0,
D",&C3,"SET 0,E",&C4,"SET 0,H",&C5,"SET 0,L"
1830 DATA &CE,"SET 1,(HL)",&CF,"SET 1,A",&CB,"SET 1,B",&C9,"SET 1,C",&CA,"SET 1,
D",&CB,"SET 1,E",&CC,"SET 1,H",&CD,"SET 1,L"
1840 DATA &D6,"SET 2,(HL)",&D7,"SET 2,A",&D8,"SET 2,B",&D1,"SET 2,C",&D2,"SET 2,
D",&D3,"SET 2,E",&D4,"SET 2,H",&D5,"SET 2,L"
1850 DATA &DE,"SET 3,(HL)",&DF,"SET 3,A",&DB,"SET 3,B",&D9,"SET 3,C",&DA,"SET 3,
D",&DB,"SET 3,E",&DC,"SET 3,H",&DD,"SET 3,L"
1860 DATA &E6,"SET 4,(HL)",&E7,"SET 4,A",&E8,"SET 4,B",&E1,"SET 4,C",&E2,"SET 4,
D",&E3,"SET 4,E",&E4,"SET 4,H",&E5,"SET 4,L"
1870 DATA &EE,"SET 5,(HL)",&EF,"SET 5,A",&EB,"SET 5,B",&E9,"SET 5,C",&EA,"SET 5,
D",&EB,"SET 5,E",&EC,"SET 5,H",&ED,"SET 5,L"
1880 DATA &F6,"SET 6,(HL)",&F7,"SET 6,A",&F8,"SET 6,B",&F1,"SET 6,C",&F2,"SET 6,
D",&F3,"SET 6,E",&F4,"SET 6,H",&F5,"SET 6,L"
1890 DATA &FE,"SET 7,(HL)",&FF,"SET 7,A",&FB,"SET 7,B",&F9,"SET 7,C",&FA,"SET 7,
D",&FB,"SET 7,E",&FC,"SET 7,H",&FD,"SET 7,L"
1900 DATA &26,SLA (HL),&27,SLA A,&20,SLA B,&21,SLA C,&22,SLA D,&23,SLA E,&24,SLA
H,&25,SLA L
1910 DATA &2E,SRA (HL),&2F,SRA A,&28,SRA B,&29,SRA C,&2A,SRA D,&2B,SRA E,&2C,SRA
H,&2D,SRA L
1920 DATA &3E,SRL (HL),&3F,SRL A,&38,SRL B,&39,SRL C,&3A,SRL D,&3B,SRL E,&3C,SRL
H,&3D,SRL L
1930 REM -----
1940 REM                      Codes de type 4
1950 REM -----
1960 DATA &4A,"ADC HL,BC",&5A,"ADC HL,DE",&6A,"ADC HL,HL",&7A,"ADC HL,SP",&A9,CP
D,&B9,&CPDR,&A1,CPI,&B1,CPIR,&46,IM 0,&56,IM 1,&5E,I
M 2
1970 DATA &7B,"IN A,(C)",&40,"IN B,(C)",&48,"IN C,(C)",&50,"IN D,(C)",&58,"IN E,
(C)",&60,"IN H,(C)",&68,"IN L,(C)"
1980 DATA &AA,IND,&BA,INDR,&A2,INI,&B2,INIR,&57,"LD A,I",&5F,"LD A,R"
1990 DATA &47,"LD I,A",&4F,"LD R,A",&A8,LDD,&BB,LDDR,&A0,LDI,&B0,LDIR,&44,NEG,&B
B,OTDR
2000 DATA &B3,OTIR,&79,"OUT (C),A",&41,"OUT (C),B",&49,"OUT (C),C",&51,"OUT (C),
D",&59,"OUT (C),E",&61,"OUT (C),H",&69,"OUT (C),L"
2010 DATA &AB,OUT D,&A3,OUTI,&4D,RETI,&45,RETN,&6F,RLD,&67,RRD,&42,"SBC HL,BC",&
52,"SBC HL,DE",&62,"SBC HL,HL",&72,"SBC HL,SP"
2020 DATA &43,LD (,),BC,&53,LD (,),SP,&4B,"LD BC,(,)",&5B,"LD DE,(,)",&7B,"L
D SP,(,)"

```



```
3000 REM =====
3010 REM                      Initialisation
3020 REM =====
3030 DIM L(256),C$(256) 'Longueur et Code operatoire
3040 FOR I=0 TO 255
3050   READ L(I),C$(I)
3060 NEXT I
3070 REM -----
3080 DIM D1(10),D2(24),D4(30),D5(30),D6(248),D7(50),D8(5),E1(13),E2(13),E4(23),E
5(23)
3090 DIM C1$(10),C2$(24),C3$(24),C4$(30),C5$(30),C6$(248),C7$(50),C8$(5),C9$(5),
D1$(13),D2$(13),D3$(13),D4$(23),D5$(23)
3100 REM Lecture des donnees speciales de type 1
3110 FOR I=1 TO 10
3120   READ D1(I):READ C1$(I)
3130 NEXT I
3140 FOR I=1 TO 24
3150   READ D2(I):READ C2$(I):READ C3$(I)
3160 NEXT I
3170 FOR I=1 TO 30
3180   READ D4(I):READ D5(I):READ C4$(I):READ C5$(I)
3190 NEXT I
3200 REM -----
3210 REM Lecture des donnees speciales de type 2
3220 FOR I=1 TO 13
3230   READ E1(I):READ D1$(I)
3240 NEXT I
3250 FOR I=1 TO 13
3260   READ E2(I):READ D2$(I):READ D3$(I)
3270 NEXT I
3280 FOR I=1 TO 23
3290   READ E4(I):READ E5(I):READ D4$(I):READ D5$(I)
3300 NEXT I
```

```
3310 REM -----
3320 REM Lecture des donnees speciales de type 3
3330 FOR I=1 TO 248
3340   READ D6(I):READ C6$(I)
3350 NEXT I
3360 REM -----
3370 REM Lecture des donnees speciales de type 4
3380 FOR I=1 TO 50
3390   READ D7(I):READ C7$(I)
3400 NEXT I
3410 FOR I=1 TO 5
3420   READ D8(I):READ C8$(I):READ C9$(I)
3430 NEXT I
3440 RETURN
4000 REM =====
4010 REM               Desassemblage
4020 REM =====
4030 MODE 2:INPUT "Adresse de debut de desassemblage ";ADR
4040 INPUT "Adresse de fin   de desassemblege ";FIN:PRINT
4050 A=PEEK(ADR)
4060 PRINT HEX$(ADR),
4070 IF VAL(C$(A))<>0 THEN GOTO 5000 'Traitement special
4080 PRINT C$(A);
4090 IF L(A)=1 THEN ADR=ADR+1
4100 IF L(A)=2 THEN ADR=ADR+1:PRINT"#";HEX$(PEEK(ADR));:ADR=ADR+1
4110 IF L(A)=3 THEN ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*256:ADR=ADR+1:
PRINT "#";HEX$(A);
4120 PRINT
4130 IF ADR<=FIN GOTO 4050 'Boucle de desassemblage
4140 RETURN
5000 REM =====
5010 REM               Traitements speciaux
5020 REM =====
```

```
5030 ON VAL(C$(A)) GOTO 5050,5280,5530,5640,5780,5790,5800,5810,5820,5830
5040 REM -----
5050 REM                               Traitement special No 1
5060 REM -----
5070 ADR=ADR+1:P=PEEK(ADR)
5080 IF P=&36 THEN PRINT"LD (IX+#;HEX$(PEEK(ADR+1));"),";HEX$(PEEK(ADR+2)):ADR=
ADR+3:GOTO 5260
5090 IF P=&22 THEN PRINT"LD (";V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V);"
),IX":ADR=ADR+3:GOTO 5260
5100 IF P=&2A THEN PRINT"LD IX,(":V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V)
;")":ADR=ADR+3:GOTO 5260
5110 IF P=&21 THEN PRINT"LD IX,":V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V):
ADR=ADR+3:GOTO 5260
5120 T=0 'Indicateur de code trouve
5130 FOR I=1 TO 10
5140   IF P=D1(I) THEN T=I
5150 NEXT I
5160 IF T<>0 THEN PRINT C1$(T):GOTO 5260
5170 FOR I=1 TO 24
5180   IF P=D2(I) THEN T=I
5190 NEXT I
5200 IF T<>0 THEN PRINT C2$(T);HEX$(PEEK(ADR+1));C3$(T):ADR=ADR+2:GOTO 5260
5210 FOR I=1 TO 30
5220   IF P=D4(I) AND PEEK(ADR+2)=D5(I) THEN T=I
5230 NEXT I
5240 IF T<>0 THEN PRINT C4$(T);HEX$(PEEK(ADR+1));C5$(T):ADR=ADR+3:GOTO 5260
5250 PRINT HEX$(PEEK(ADR))
5260 GOTO 4050
5270 REM -----
5280 REM                               Traitement special No 2
5290 REM -----
5300 ADR=ADR+1:P=PEEK(ADR)
5310 IF P=&36 THEN PRINT"LD (IY+#;HEX$(PEEK(ADR+1));"),";HEX$(PEEK(ADR+2)):ADR=
ADR+3:GOTO 5490
```

```
5320 IF P=&22 THEN PRINT"LD (" ;V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V);"  
) ,IY":ADR=ADR+3:GOTO 5490  
  
5330 IF P=&2A THEN PRINT"LD IY, (" ;V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V)  
;")":ADR=ADR+3:GOTO 5490  
  
5340 IF P=&21 THEN PRINT"LD IY, " ;V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT"#";HEX$(V) :  
ADR=ADR+3:GOTO 5490  
  
5350 T=0 'Indicateur de code trouve  
  
5360 FOR I=1 TO 13  
  
5370   IF P=E1(I) THEN T=I  
  
5380 NEXT I  
  
5390 IF T<>0 THEN PRINT D1$(T):GOTO 5490  
  
5400 FOR I=1 TO 13  
  
5410   IF P=E2(I) THEN T=I  
  
5420 NEXT I  
  
5430 IF T<>0 THEN PRINT D2$(T);HEX$(PEEK(ADR+1));D3$(T):ADR=ADR+2:GOTO 5490  
  
5440 FOR I=1 TO 23  
  
5450   IF P=E4(I) AND PEEK(ADR+2)=E5(I) THEN T=I  
  
5460 NEXT I  
  
5470 IF T<>0 THEN PRINT D4$(T);HEX$(PEEK(ADR+1));D5$(T):ADR=ADR+3:GOTO 5490  
  
5480 PRINT HEX$(PEEK(ADR))  
  
5490 GOTO 4050  
  
5500 REM -----  
  
5510 REM                               Traitement special No 3  
  
5520 REM -----  
  
5530 ADR=ADR+1:P=PEEK(ADR):T=0 'Initialisation  
  
5540 FOR I=1 TO 248  
  
5550   IF P=D6(I) THEN T=I  
  
5560 NEXT I  
  
5570 IF T<>0 THEN PRINT C6$(T):GOTO 5590  
  
5580 PRINT HEX$(PEEK(T))  
  
5590 GOTO 4050  
  
5600 REM -----  
  
5610 REM                               Traitement special No 4  
  
5620 REM -----
```

```
5630 REM Traitement special No 4
5640 ADR=ADR+1:P=PEEK(ADR):T=0 'Initialisation
5650 FOR I=1 TO 50
5660 IF P=D7(I) THEN T=I
5670 NEXT I
5680 IF T<>0 THEN PRINT C7$(T):ADR=ADR+1:GOTO 5740
5690 FOR I=1 TO 5
5700 IF P=D8(I) THEN T=I
5710 NEXT I
5720 IF T<>0 THEN PRINT C8$(I);V=PEEK(ADR+1)+PEEK(ADR+2)*256:PRINT HEX$(V);C9$(
T):ADR=ADR+3:GOTO 5740
5730 PRINT HEX$(P)
5740 GOTO 4050
5750 REM -----
5760 REM Traitements speciaux No 5 a 10
5770 REM -----
5780 PRINT"LD (";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1:PR
INT"#";HEX$(A);"),HL":GOTO 4050
5790 PRINT"LD HL,(";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1
:PRINT"#";HEX$(A);")":GOTO 4050
5800 PRINT"LD (";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1:PR
INT"#";HEX$(A);"),A":GOTO 4050
5810 PRINT"LD A,(";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1:
PRINT"#";HEX$(A);")":GOTO 4050
5820 PRINT"OUT (";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1:P
RINT"#";HEX$(A);"),A":GOTO 4050
5830 PRINT"IN A,(";ADR=ADR+1:A=PEEK(ADR):ADR=ADR+1:A=A+PEEK(ADR)*255:ADR=ADR+1:
PRINT"#";HEX$(A);")":GOTO 4050
```

Lignes 10 à 30 : Programme principal.

Lignes 1000 à 2020 : Données.

Lignes 3000 à 3440 : Initialisation.

Lignes 4000 à 4140 : Désassemblage.

Lignes 5000 à 5830 : Traitements spéciaux.

