OPCODES TABLE OF INTEL 8085

Opcodes of Intel 8085 in Alphabetical Order

Sr. No.	Mnemonics, Operand	Opcode	Bytes
1.	ACI Data	CE	2
2.	ADC A	8F	1
3.	ADC B	88	1
4.	ADC C	89	1
5.	ADC D	8A	1
6.	ADC E	8B	1
7.	ADC H	8C	1
8.	ADC L	8D	1
9.	ADC M	8E	1
10.	ADD A	87	1
11.	ADD B	80	1
12.	ADD C	81	1
13.	ADD D	82	1
14.	ADD E	83	1
15.	ADD H	84	1
16.	ADD L	85	1
17.	ADD M	86	1
18.	ADI Data	C6	2
19.	ANA A	A7	1
20.	ANA B	A0	1
21.	ANA C	A1	1
22.	ANA D	A2	1
23.	ANA E	A3	1
24.	ANA H	A4	1
25.	ANA L	A5	1
26.	ANA M	A6	1
27.	ANI Data	E6	2
28.	CALL Label	CD	3
29.	CC Label	DC	3
30.	CM Label	FC	3
31.	CMA	2F	1
32.	CMC	3F	1
33.	CMP A	BF	1
34.	CMP B	B8	1
35.	CMP C	B9	1
36.	CMP D	BA	1
37.	CMP E	BB	1
38.	CMP H	BC	1
39.	CMP L	BD	1
40.	CMP M	BD	1
41.	CNC Label	D4	3
42.	CNZ Label	C4	3

Sr. No.	Mnemonics, Operand	Opcode	Bytes
43.	CP Label	F4	3
44.	CPE Label	EC	3
45.	CPI Data	FE	2
46.	CPO Label	E4	3
47.	CZ Label	CC	3
48.	DAA	27	1
49.	DAD B	09	1
50.	DAD D	19	1
51.	DAD H	29	1
52.	DAD SP	39	1
53.	DCR A	3D	1
54.	DCR B	05	1
55.	DCR C	0D	1
56.	DCR D	15	1
57.	DCR E	1D	1
58.	DCR H	25	1
59.	DCR L	2D	1
60.	DCR M	35	1
61.	DCX B	0B	1
62.	DCX D	1B	1
63.	DCX H	2B	1
64.	DCX SP	3B	1
65.	DI	F3	1
66.	EI	FB	1
67.	HLT	76	1
68.	IN Port-address	DB	2
69.	INR A	3C	1
70.	INR B	04	1
71.	INR C	0C	1
72.	INR D	14	1
73.	INR E	1C	1
74.	INR H	24	1
75.	INR L	2C	1
76.	INR M	34	1
77.	INX B	03	1
78.	INX D	13	1
79.	INX H	23	1
80.	INX SP	33	1
81.	JC Label	DA	3
82.	JM Label	FA	3
83.	JMP Label	C3	3
84.	JNC Label	D2	3
85.	JNZ Label	C2	3
86.	JP Label	F2	3
87.	JPE Label	EA	3
88.	JPO Label	E2	3
89.	JZ Label	CA	3

Sr. No.	Mnemonics, Operand	Opcode	Bytes
90.	LDA Address	3A	3
91.	LDAX B	0A	1
92.	LDAX D	1A	1
93.	LHLD Address	2A	3
94.	LXI B	01	3
95.	LXI D	11	3
96.	LXI H	21	3
97.	LXI SP	31	3
98.	MOV A, A	7F	1
99.	MOV A, B	78	1
100.	MOV A, C	79	1
101.	MOV A, D	7A	1
102.	MOV A, E	7B	1
103.	MOV A, H	7C	1
104.	MOV A, L	7D	1
105.	MOV A, M	7E	1
106.	MOV B, A	47	1
107.	MOV B, B	40	1
108.	MOV B, C	41	1
109.	MOV B, D	42	1
110.	MOV B, E	43	1
111.	MOV B, H	44	1
112.	MOV B, L	45	1
113.	MOV B, M	46	1
114.	MOV C, A	4F	1
115.	MOV C, B	48	1
116.	MOV C, C	49	1
117.	MOV C, D	4A	1
118.	MOV C, E	4B	1
119.	MOV C, H	4C	1
120.	MOV C, L	4D	1
121.	MOV C, M	4E	1
122.	MOV D, A	57	1
123.	MOV D, B	50	1
124.	MOV D, C	51	1
125.	MOV D, D	52	1
126.	MOV D, E	53	1
127.	MOV D, H	54	1
128.	MOV D, L	55	1
129.	MOV D, M	56	1
130.	MOV E, A	5F	1
131.	MOV E, B	58	1
132.	MOV E, C	59	1
133.	MOV E, D	5A	1
134.	MOV E, E	5B	1
135.	MOV E, H	5C	1
136.	MOV E, L	5D	1

Sr. No.	Mnemonics, Operand	Opcode	Bytes
137.	MOV E, M	5E	1
138.	MOV H, A	67	1
139.	MOV H, B	60	1
140.	MOV H, C	61	1
141.	MOV H, D	62	1
142.	MOV H, E	63	1
143.	MOV H, H	64	1
144.	MOV H, L	65	1
145.	MOV H, M	66	1
146.	MOV L, A	6F	1
147.	MOV L, B	68	1
148.	MOV L, C	69	1
149.	MOV L, D	6A	1
150.	MOV L, E	6B	1
151.	MOV L, H	6C	1
152.	MOV L, L	6D	1
153.	MOV L, M	6E	1
154.	MOV M, A	77	1
155.	MOV M, B	70	1
156.	MOV M, C	71	1
157.	MOV M, D	72	1
158.	MOV M, E	73	1
159.	MOV M, H	74	1
160.	MOV M, L	75	1
161.	MVI A, Data	3E	2
162.	MVI B, Data	06	2
163.	MVI C, Data	0E	2
164.	MVI D, Data	16	2
165.	MVI E, Data	1E	2
166.	MVI H, Data	26	2
167.	MVI L, Data	2E	2
168.	MVI M, Data	36	2
169.	NOP	00	1
170.	ORA A	B7	1
171.	ORA B	B0	1
172.	ORA C	B1	1
173.	ORA D	B2	1
174.	ORA E	B3	1
175.	ORA H	B4	1
176.	ORA L	B5	1
177.	ORA M	B6	1
178.	ORI Data	F6	2
179.	OUT Port-Address	D3	2
180.	PCHL	E9	1
181.	POP B	C1	1
182.	POP D	D1	1
183.	POP H	E1	1

Sr. No.	Mnemonics, Operand	Opcode	Bytes
184.	POP PSW	F1	1
185.	PUSH B	C5	1
186.	PUSH D	D5	1
187.	PUSH H	E5	1
188.	PUSH PSW	F5	1
189.	RAL	17	1
190.	RAR	1F	1
191.	RC	D8	1
192.	RET	C9	1
193.	RIM	20	1
194.	RLC	07	1
195.	RM	F8	1
196.	RNC	D0	1
197.	RNZ	C0	1
198.	RP	F0	1
199.	RPE	E8	1
200.	RPO	E0	1
201.	RRC	0F	1
202.	RST 0	C7	1
203.	RST 1	CF	1
204.	RST 2	D7	1
205.	RST 3	DF	1
206.	RST 4	E7	1
207.	RST 5	EF	1
208.	RST 6	F7	1
209.	RST 7	FF	1
210.	RZ	C8	1
211.	SBB A	9F	1
212.	SBB B	98	1
213.	SBB C	99	1
214.	SBB D	9A	1
215.	SBB E	9B	1
216.	SBB H	9C	1
217.	SBB L	9D	1
218.	SBB M	9E	1
219.	SBI Data	DE	2
220.	SHLD Address	22	3
221.	SIM	30	1
222.	SPHL	F9	1
223.	STA Address	32	3
224.	STAX B	02	1
225.	STAX D	12	1
226.	STC	37	1
227.	SUB A	97	1
228.	SUB B	90	1
229.	SUB C	91	1
230.	SUB D	92	1

Sr. No.	Mnemonics, Operand	Opcode	Bytes
231.	SUB E	93	1
232.	SUB H	94	1
233.	SUB L	95	1
234.	SUB M	96	1
235.	SUI Data	D6	2
236.	XCHG	EB	1
237.	XRA A	AF	1
238.	XRA B	A8	1
239.	XRA C	A9	1
240.	XRA D	AA	1
241.	XRA E	AB	1
242.	XRA H	AC	1
243.	XRA L	AD	1
244.	XRA M	AE	1
245.	XRI Data	EE	2
246.	XTHL	E3	1

Memory Location	**Code**	**Opcode**
8000	MVI A, 48	3E
8001		48
8002	MVI B, 48	06
8003		48
8004	ADD B	80
8005	STA 8500	32
8006		00
8007		85
8008	RST 5	EF

addition of two 8-bit numbers

Memory Location	**Code**	**Opcode**
8000	LDA 8500	3A
8001		00
8002		85
8003	MOV B,A	47
8004	LDA 8501	3A
8005		01
8006		85
8007	ADD B	80
8008	STA 8502	32
8009		02
800A		85
800B	RST 5	EF

Direct addressing INPUT [8500] = 88, [8501] = 88
OUTPUT [8502] = 10

Memory Location	Code	Opcode
8000	LXI H,8500	21
8001		00
8002		85
8003	MOV A,M	7E
8004	INX H	23
8005	ADD M	86
8006	INX H	23
8007	MOV M,A	77
8008	RST 5	EF

Indirect addressing

INPUT [8500] = 88, [8501] = 88
OUTPUT [8502] = 10

Memory Location	Code	Opcode
8000	LHLD 8500	2A
8001		00
8002		85
8003	XCHG	EB
8004	LHLD 8502	2A
8005		02
8006		85
8007	DAD D	19
8008	SHLD 8504	22
8009		04
800A		85
800B	RST 5	EF

16 bit direct

Memory Location	Code	Opcode
8000	LXI B, 8500	01
8001		00
8002		85
8003	LDAX B	0A
8004	MOV D, A	57
8005	INX B	03
8006	LDAX B	0A

8007	ADD D	82
8008	STA 8504	32
8009		04
800A		85
800B	INX B	03
800C	LDAX B	0A
800D	MOV D, A	57
800E	INX B	03
800F	LDAX B	0A
8010	ADC D	8A
8011	STA 8505	32
8012		05
8013		85
8014	RST 5	EF

16 bit indirect

Input [8500] = 34, [8501] = 48, [8502] = 54, [8503] = 78

Output [8504] = 7C [8505] = CC

Add using carry

Memory Location	**Code**	**Opcode**
8000	MVI C, 00	0E
8001		00
8002	LXI H, 8500	21
8003		00
8004		85
8005	MOV A, M	7E
8006	INX H	23
8007	ADD M	86
8008	JNC Next	D2
8009		0C
800A		80
800B	INR C	0C
800C	Next: INX H	23
800D	MOV M, A	77
800E	INX H	23
800F	MOV M,C	71
8010	RST 5	EF

Add using carry

Memory Location	Code	Opcode
8000	LDA 8500	3A
8001		00
8002		85
8003	CMA	2F
8004	STA 8501	32
8005		01
8006		85
8007	INR A	3C
8008	STA 8502	32
8009		02
800A		85
800B	RST 5	EF

1,2 complemet

INPUT [8500] = 48
OUTPUT [8501] = B7, [8502] = B8

Memory Location	Code	Opcode
8000	LDA 8200	3A
8001		00
8002		82
8003	MOV C, A	4F
8004	SUB A	97
8005	LXI H 8201	21
8006		01
8007		82
8008	Back: ADD M	86
8009	INX H	23
800A	DCR C	0D
800B	JNZ Back	C2
800C		08
800D		80
800E	STA 8300	32
800F		00
8010		83
8011	RST 5	EF

Sum of series of no

INPUT [8200] = 04, [8201] = 9A, [8202] = 52, [8203] = 89, [8204] = 3EOUTPUT [8300] = B3, CARRY=1

Memory Location	**Code**	**Opcode**
8000	MVI C, 0A	0E
8001		0A
8002	LXI H, 8200	21
8003		00
8004		82
8005	LXI D, 8300	11
8006		00
8007		83
8008	Back: MOV A,M	7E
8009	STAX D	12
800A	INX H	23
800B	INX D	13
800C	DCR C	0D
800D	JNZ Back	C2
800E		08
800F		80
8010	RST 5	EF

Data transfer from b1 to b2

INPUT	[8200] TO [8209] = 01 TO 0A
OUTPUT	[8300] TO [8309] = 01 TO 0A

Memory Location	**Code**	**Opcode**
8000	LDA 8200	3A
8001		00
8002		82
8003	MOV E, A	5F
8004	MVI D, 00	16
8005		00
8006	LDA 8201	3A
8007		01
8008		82
8009	MOV C,A	4F
800A	LXI H, 0000	21
800B		00
800C		00
800D	Back: DAD D	19
800E	DCR C	0D

800F	JNZ Back	C2
8010		0D
8011		80
8012	SHLD 8300	22
8013		00
8014		83
8015	RST 5	EF

Multiply 2 8 bit numbers

INPUT [8200] = B2, [8201] = 03
OUTPUT [8300] = 16, [8301] = 02

Memory Location	Code	Opcode
8000	MVI C,00	0E
8001		00
8002	MVI B,09	06
8003		09
8004	LXI H,8500	21
8005		00
8006		85
8007	MOV A,M	7E
8008	Back: INX H	23
8009	ADD M	86
800A	JNC Next	D2
800B		0E
800C		80
800D	INR C	0C
800E	Next: DCR B	05
800F	JNZ Back	C2
8010		08
8011		80
8012	INX H	23
8013	MOV M,A	77
8014	INX H	23
8015	MOV M,C	71
8016	RST 5	EF

Write a program to add ten 8-bit numbers. Assume the numbers are stored in8500-8509. Store the result in 850A and 850B memory address**.**

Memory Location	Code	Opcode
8000	LDA 8200	3A
8001		00
8002		82
8003	MOV C,A	4F

8004	MVI B,00	06
8005		00
8006	LXI H,8201	21
8007		01
8008		82
8009	Back: MOV A,M	7E
800A	ANI 80	E6
800B		80
800C	JZ Skip	CA
800D		10
800E		80
800F	INR B	04
8010	Skip: INX H	23
8011	DCR C	0D
8012	JNZ Back	C2
8013		09
8014		80
8015	MOV A,B	78
8016	STA 8300	32
8017		00
8018		83
8019	RST 5	EF

Negative no in a block of data

INPUT [8200] = 04, [8201] = 56, [8202] = A9, [8203] =73, [8204] = 82
OUTPUT [8300] = 02

Memory Location	Code	Opcode
8000	LDA 8500	3A
8001		00
8002		85
8003	MVI B,08	06
8004		08
8005	MVI D,00	16
8006		00
8007	Loop1: RLC	07
8008	JNC Loop2	D2
8009		0C
800A		80
800B	INR D	14
800C	Loop2: DCR B	05
800D	JNZ Loop1	C2
800E		07
800F		80
8010	MOV A,D	7A
8011	STA 9000	32

8012		00
8013		90
8014	RST 5	EF

No of ones in a number

INPUT [8500] = 25

OUTPUT [9000] = 03

Memory Location	**Code**	**Opcode**
8000	LXI H,9000	21
8001		00
8002		90
8003	MOV C,M	4E
8004	DCR C	0D
8005	Repeat: MOV D,C	51
8006	LXI H,9001	21
8007		01
8008		90
8009	Loop: MOV A,M	7E
800A	INX H	23
800B	CMP M	BE
800C	JC Skip	DA
800D		14
800E		80
800F	MOV B,M	46
8010	MOV M,A	77
8011	DCX H	2B
8012	MOV M,B	70
8013	INX H	23
8014	Skip: DCR D	15
8015	JNZ Loop	C2
8016		09
8017		80
8018	DCR C	0D
8019	JNZ Repeat	C2
801A		05
801B		80
801C	RST 5	EF

Numbers in ascending ordre

INPUT [9000]=05,[9001]=05, [9002]=04, [9003]=03, [9004]=02, [9005]=01

OUTPUT [9000]=05, [9001]=01, [9002]=02, [9003]=03, [9004]=04, [9005]=05

Memory Location	**Code**	**Opcode**
8000	LDA 8200	3A
8001		00
8002		82
8003	MOV C,A	4F
8004	MVI B,00	06
8005		00

8006	LXI H,8201	21
8007		01
8008		82
8009	Back: MOV A,M	7E
800A	ANI 01	E6
800B		01
800C	JNZ Skip	C2
800D		12
800E		80
800F	MOV A,B	78
8010	ADD M	86
8011	MOV B,A	47
8012	Skip: INX H	23
8013	DCR C	0D
8014	JNZ Back	C2
8015		09
8016		80
8017	MOV A,B	78
8018	STA 820A	32
8019		0A
801A		82
801B	RST 5	EF

Sum of series of even numbers

INPUT [8200]=04, [8201]=20, [8202]=15, [8203]=13,
[8204]=22OUTPUT [820A]=42

Memory Location	Code	Opcode
8000	MVI B, 0A	06
8001		0A
8002	MVI D, AD	16
8003		AD
8004	MVI C, 00	0E
8005		00
8006	LXI H, 8500H	21
8007		00
8008		85
8009	Back: MOV A, M	7E
800A	CMP D	BA
800B	JNZ Next	C2
800C		0F
800D		80
800E	INR C	0C
800F	Next: INX H	23

8010	DCR B	05
8011	JNZ Back	C2
8012		09
8013		80
8014	MOV A, C	79
8015	STA 8600H	32
8016		00
8017		86
8018	RST 5	EF

program to verify how many bytes are presentin a given set, which resembles 10101101 in 8085

INPUT [8500] = AD, [8501] = 01, [8502] = 01, [8503] = 01, [8504] = 01, [8505] = 01, [8506] = 01, [8507] = 01, [8508] = 01, [8509] = 01
OUTPUT [8600] = 01

Memory Location	**Code**	**Opcode**
8000	MVI B, 0A	06
8001		0A
8002	MVI C, 00	0E
8003		00
8004	LXI H, 8500H	21
8005		00
8006		85
8007	Back: MOV A, M	7E
8008	ANI FF	E6
8009		FF
800A	JPO Next	E2
800B		0E
800C		80
800D	INR C	0C
800E	Next: INX H	23
800F	DCR B	05
8010	JNZ Back	C2

8011		07
8012		80
8013	MOV A, C	79
8014	STA 8600H	32
8015		00
8016		86
8017	RST 5	EF

to find the numbers of even parity inten consecutive memory locations in 8085.

INPUT [8500] = 01, [8501] = 03, [8502] = 01, [8503] = 03, [8504] = 01, [8505] = 03, [8506] = 01, [8507] = 03, [8508] = 01, [8509] = 03

OUTPUT [8600] = 05

Memory Location	**Code**	**Opcode**
8000	LDA 8500H	3A
8001		00
8002		85
8003	MOV B, A	47
8004	ANI 0F	E6
8005		0F
8006	MOV C, A	4F
8007	MOV A, B	78
8008	ANI F0	E6
8009		F0
800A	RRC	0F
800B	RRC	0F
800C	RRC	0F
800D	RRC	0F
800E	MOV B, A	47
800F	XRA A	AF
8010	MVI D, 0A	16
8011		0A
8012	Sum: ADD D	82
8013	DCR B	05
8014	JNZ Sum	C2
8015		12
8016		80
8017	ADD C	81
8018	STA 8600H	32
8019		00
801A		86
801B	RST 5	EF

Bcd into binaryINPUT [8500] = 67

OUPUT [8600] = 43

Memory Location	Code	Opcode
8000	MVI B, 09	06
8001		09
8002	LXI H, 8500H	21
8003		00
8004		85
8005	MOV A, M	7E
8006	INX H	23
8007	Back: CMP M	BE
8008	JNC Next	D2
8009		0C
800A		80
800B	MOV A, M	7E
800C	Next: INX H	23
800D	DCR B	05
800E	JNZ Back	C2
800F		07
8010		80
8011	STA 850AH	32
8012		0A
8013		85
8014	RST 5	EF

Largest in array of 10 elements
Input [8500] 01, [8501] TO [8509] = 02 TO 0A
Output [850A] 0A

Memory Location	Code	Opcode
8000	LDA 8500H	3A
8001		00
8002		85
8003	MOV B, A	47
8004	LDA 8600H	3A
8005		00
8006		86
8007	STA 8500H	32
8008		00
8009		85
800A	MOV A, B	78
800B	STA 8600H	32
800C		00
800D		86
800E	RST 5	EF

assembly language program for exchange the contents of memorylocation.