XII Computer Science - Part II 8085 Programs

Addition of two 8-bit numbers.

2) Write a program to do addition between 8-bit numbers. Two numbers are stored in 7000H and 7001H. Store the result in 7002H.

Data Before execution

Memory Location	Data
7000	02
7001	03
7002	00

Data After execution

Memory Location	Data
7000	02
7001	03
7002	05

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7003	START	LXI	H,7000	21	Initialize HL as a Memory pointer
7004				00	Lower address byte
7005				70	Higher address byte
7006		MOV	A,M	7E	Move the first number in A reg.
7007		INX	H	23	Increment HL pointer by one
7008		ADD	M	86	Add accumulator data with HL pointer data
7009		STA	7002	32	Store result in memory location
700A				02	Lower address byte
700B				70	Higher address byte
700C	STOP	RST 1		CF	Stop program execution.

Addition of two 8-bit BCD numbers.

3) Write a program to do addition between 8-bit BCD numbers. Two numbers are stored in 7000H and 7001H. Store the result in 7002H onwards starting with least significant bit.

Data Before execution

Memory Location	Data
7000	56
7001	55
7002	00
7003	00

Data After execution

Memory Location	Data
7000	56
7001	55
7002	11
7003	01

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7004	START	LXI	H,7000	21	Initialize HL as a Memory pointer
7005				00	Lower address byte
7006				70	Higher address byte
7007		MVI	C,00 H	0E	Initialize C Reg. with Zero
7008				00	
7009		MOV	A,M	7E	Move the first number in A reg.
700A		INX	H	23	Increment HL pointer by one.
700B		ADD	M	86	Add accumulator data with HL pointer data.
700C		DAA		27	Decimal Adjust Accumulator.
700D		JNC	7011	D2	If carry not generated Jump to 7001H
700E				11	Lower address byte
700F				70	Higher address byte
7010		INR	C	0C	Increment C reg by one.
7011	L1	STA	7002	32	Store result in memory location
7012				02	Lower address byte
7013				70	Higher address byte
7014		MOV	A,C	79	Move C reg data to accumulator
7015		STA	7003	32	Store carry in memory location
7016				03	Lower address byte
7017				70	Higher address byte
7018	STOP	RST 1		CF	Stop program execution

4) Write a program that multiplies two 1-byte hex numbers stored in consecutive memory locations starting from 7015H. Store the two byte result in consecutive memory locations starting from 7017H beginning with lower order byte.

Data Before execution

Memory Location	Data
7015	09
7016	05
7017	00
7018	00

Data After execution

Memory Location	Data
7015	09
7016	05
7017	2D
7018	00

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7000	START	XRA	A	AF	Clear the a reg.
7001		MOV	B,A	47	Initialize B reg to zero to record the higher byte
7002		LXI	H,7015	21	Initialize HL as a Memory pointer
7003				15	Lower address byte
7004				70	Higher address byte
7005		MOV	C,M	4E	Move length in C reg.
7006		INX	H	23	Increment HL pointer by one.
7007	L1	ADD	M	86	Add memory content to accumulator
7008		JNC	700C	D2	If CY=0 then jump to label P1
7009				0C	Lower address byte
700A				70	Higher address byte
700B		INR	B	04	Increment B reg by one
700C	P1	DCR	C	0D	decrement C reg by one
700D		JNZ	7007	C2	If counter C is not zero goto label L1
700E				07	Lower address byte
700F				70	Higher address byte
7010		INX	H	23	Increment HL pointer by one.
7011		MOV	M,A	77	Move result from accumulator to memory location pointed by HL
7012		INX	H	23	Increment HL pointer by one.
7013		MOV	M,B	70	Move carry from B reg. to memory location pointed by HL
7014	STOP	RST 1		CF	Stop program execution

5) Write a program that divides two 1 byte hex numbers where the dividend is stored in 7030H and the divisor stored in 7031H. Store the quotient and the remainder in the next consecutive memory locations respectively.

Data Before execution

Memory Location	Data
7030	0D
7031	03
7032	00
7033	00

Data After execution

Memory Location	Data
7030	0D
7031	03
7032	04
7033	01

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7000	START	XRA	A	AF	Clear the A reg.
7001		MOV	B,A	47	Initialize B reg to zero to record the higher byte
7002		LXI	H,7030	21	Initialize HL as a Memory pointer
7003				30	Lower address byte
7004				70	Higher address byte
7005		MOV	A,M	7E	Move length in A reg.
7006		INX	H	23	Increment HL pointer by one.
7007	PP	CMP	M	BE	Compare to check dividend is greater than divisor.
7008		JC	7010	DA	If A<M then goto label VP
7009				10	Lower address byte
700A				70	Higher address byte
700B		SUB	M	96	Subtract divisor from dividend
700C		INR	B	04	Increment quotient from dividend
700D		JMP	7007	C3	Jump unconditionally to label PP
700E				07	Lower address byte
700F				70	Higher address byte
7010	VP	INX	H	23	Increment HL contents by one.
7011		MOV	M,B	70	Store quotient at 7032H
7012		INX	H	23	Increment HL contents by one.
7013		MOV	M,A	77	Store remainder at 7033H
7014	STOP	RST1		CF	Stop the execution.

6) A block of data is stored in memory locations from 7000 H to 700C H. write a program to find the smallest as well as greatest number from this block using linear search. Store the results immediately after the end of the block.

Data Before execution

Memory Location	Data
7000	06
7001	56
7002	04
7003	01
7004	59
7005	64
7006	77
7007	3A
7008	05
7009	12
700A	3F
700B	00
700C	00

Data After execution

Memory Location	Data
7000	06
7001	56
7002	04
7003	01
7004	59
7005	64
7006	77
7007	3A
7008	05
7009	12
700A	3F
700B	77
700C	01

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	**Operand**		
700D	START	MVI	C,0B	0E	Store length of block in C reg.
700E				0B	
700F		LXI	H,7000	21	Initialize HL as a memory pointer.
7010				00	Lower address byte
7011				70	Higher address byte
7012		MOV	D,M	56	Assume D contents are greatest
7013		MOV	E,M	5E	Assume E contents are smallest
7014		DCR	C	0D	Decrement length by one.
7015	VP	INX	H	23	Increment HL contents by one
7016		MOV	A,M	7E	Store next memory contents in A
7017		CMP	D	BA	Compare D contents are greater than A
7018		JC	701C	DA	If D>A then goto label SML
7019				1C	Lower address byte
701A				70	Higher address byte
701B		MOV	D,A	57	Move greatest in reg D
701C	SML	CMP	E	BB	Compare E contents are smaller than A
701D		JNC	7021	D2	If E<A then go to NXT Label
701E				21	Lower address byte
701F				70	Higher address byte
7020		MOV	E,A	5F	Move smallest in E
7021	NXT	DCR	C	0D	Decrement length by 1
7022		JNZ	7015	C2	Of flag Z=0 then goto label VP
7023				15	Lower address byte
7024				70	Higher address byte
7025		INX	H	23	Increment HL contents by one
7026		MOV	M,D	72	Store greatest in memory
7027		INX	H	23	Increment HL contents by one
7028		MOV	M,E	73	Store smallest in memory
7029	STOP	RST1		CF	Stop the execution

7) A block of data is stored in memory locations from 7000H to 700CH. write a program to find the number of ODD as well as EVEN numbers from this block. Store the results immediately after the end of the block.

Data Before execution

Memory Location	Data
7000	23
7001	AE
7002	7B
7003	1E
7004	15
7005	06
7006	05
7007	29
7008	31
7009	29
700A	13
700B	11
700C	1D
700D	00
700E	00

Data After execution

Memory Location	Data
7000	23
7001	AE
7002	7B
7003	1E
7004	15
7005	06
7006	05
7007	29
7008	31
7009	29
700A	13
700B	11
700C	1D
700D	03
700E	0A

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	**Operand**		
700F	START	MVI	C,0D	0E	Store length of block in c reg.
7010				0D	
7011		MVI	D,00	16	Set D contents to 0
7012				00	
7013		MOV	E,D	5A	Set E reg to 0
7014		LXI	H,7000	21	Store even counter
7015				00	Lower address byte
7016				70	Higher address byte
7017	VP	MOV	A,M	7E	Load A with contents of M
7018		RRC		0F	Rotate contents of A to right by one.
7019		JNC	7020	D2	If carry flag do not set then go to Label ODD
701A				20	Lower address byte
701B				70	Higher address byte
701C		INR	E	1C	Increment even counter by 1
701D		JMP	7021	C3	Jump unconditionally
701E				21	Lower address byte
701F				70	Higher address byte
7020	ODD	INR	D	14	Increment D contents by 1
7021	NXT	INX	H	23	Increment HL by one
7022		DCR	C	0D	Decrement C by one
7023		JNZ	7017	C2	If contents not zero then goto label VP
7024				17	Lower address byte
7025				70	Higher address byte
7026		MOV	M,D	72	Move ODD count at memory 700D
7027		INX	H	23	Increment HL by one
7028		MOV	M,E	73	Move EVEN count at memory 700E
7029	STOP	RST1		CF	Stop program execution

8) Write a program that subtracts the number stored in 7000H from the number stored in 7001H. Store the absolute difference in memory location 7002H as result.

Data Before execution

Memory Location	Data
7000	06
7001	0A
7002	00

Data After execution

Memory Location	Data
7000	06
7001	0A
7002	04

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7003	START	LXI	H,7000	21	Initialize HL as a memory pointer
7004				00	Lower address byte
7005				70	Higher address byte
7006		MOV	B,M	46	Move subtractor to B reg
7007		INX	H	23	Increment HL pointer by 1
7008		MOV	A,M	7E	Move minuend to A reg
7009		SUB	B	90	Subtract A minus B
700A		JP	7010	F2	If result is positive go to label VP
700B				10	Lower address byte
700C				70	Higher address byte
700D		CMA		2F	Take 1's complement of A reg
700E		ADI	01	C6	Add 1 to A reg to get 2's complement
700F				01	
7010	VP	INX	H	23	Increment HL pointer by 1
7011		MOV	M,A	77	Move A contents to memory
7012	STOP	RST1		CF	Stop program execution

9)Write a program to find how many times data AD H appears in memory block starting from 7001 H and length of block is at 7000H. Store the count in 7005H.

Data Before execution

Memory Location	Data
7000	04
7001	12
7002	AD
7003	34
7004	AD
7005	00

Data After execution

Memory Location	Data
7000	04
7001	12
7002	AD
7003	34
7004	AD
7005	02

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7006	START	MVI	C,00	0E	Move 00 to C reg
7007				00	
7008		MVI	A,AD H	3E	Move AD data to A reg
7009				AD	
700A		LXI	H,7000	21	Initialize HL with 7000H
700B				00	Lower address byte
700C				70	Higher address byte
700D		MOV	B,M	46	Move memory data to B reg
700E		INX	H	23	
700F	BACK	CMP	M	BE	Compare accumulator & memory data
7010		JNZ	7014	C2	If zero flag is not set, goto label NEXT
7011				14	Lower address byte
7012				70	Higher address byte
7013		INR	C	0C	Increment counter by 1
7014	NEXT	INX	H	23	Increment HL pair by 1
7015		DCR	B	05	Decrement B reg
7016		JNZ	700F	C2	If zero flag not set, goto label BACK
7017				0F	Lower address byte
7018				70	Higher address byte
7019		MOV	A,C	79	Move memory data to A reg
701A		STA	7005	32	Store accumulator data to memory location 7005H
701B				05	Lower address byte
701C				70	Higher address byte
701D	STOP	RST1		CF	Stop the program

10) A 8-bit number is stored in memory location 7000H. Write an assembly language program to count the zero in the given number. Store count in memory location 7001H.

Data Before execution

Memory Location	Data
7000	75
7001	00

Data After execution

Memory Location	Data
7000	75
7001	03

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7002	STOP	MVI	C,08 H	0E	Set counter to 08 H
7003				08	
7004		MVI	B,00 H	06	Set B to 00 to store number of zeros
7005				00	
7006		LDA	7000H	3A	Get number in A
7007				00	Lower address byte
7008				70	Higher address byte
7009	BACK	RRC		0F	Shift least significant bit in carry
700A		JC	700E	DA	Is carry? yes go to NEXT
700B				0E	Lower address byte
700C				70	Higher address byte
700D		INR	B	04	Increment B reg by 1
700E	NEXT	DCR	C	0D	Decrement C by 1
700F		JNZ	7009 H	C2	Is counter=0? No go to BACK
7010				09	Lower address byte
7011				70	Higher address byte
7012		MOV	A,B	78	Get contents of B reg in A reg
7013		STA	7001 H	32	Store result in 7001 H
7014				01	Lower address byte
7015				70	Higher address byte
7016	STOP	RST1		CF	Stop the program

11) Write a program that separates the two nibbles of a number stored in 7060H and stores the same in memory locations 7061H and 7062H. The program must also multiply the nibbles and stores the result in 7063H.

Data Before execution

Memory Location	Data
7060	62
7061	00
7062	00
7063	00

Data After execution

Memory Location	Data
7060	62
7061	02
7062	06
7063	0C

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
7000	START	LXI	H,7060H	21	Initialize HL with 7060H
7001				60	Lower address byte
7002				70	Higher address byte
7003		MOV	A,M	7E	Move memory data to accumulator
7004		MOV	B,M	46	Move memory data to B reg.
7005		ANI	0FH	E6	And immediate accumulator data with 0F
7006				0F	
7007		INX	H	23	Increment HL by 1
7008		MOV	M,A	77	Move accumulator data to memory
7009		MOV	C,A	4F	Move accumulator data to C reg
700A		MOV	A,B	78	Move B reg data to A reg.
700B		RLC		07	Rotate left contents of A reg.
700C		RLC		07	Rotate left contents of A reg.
700D		RLC		07	Rotate left contents of A reg.
700E		RLC		07	Rotate left contents of A reg.
700F		ANI	0F	E6	And immediate accumulator data with 0F
7010				0F	
7011		INX	H	23	Increment HL by 1
7012		MOV	M,A	77	Move accumulator data to memory
7013		XRA	A	AF	Make accumulator data 0
7014	S1	ADD	M	86	Add memory data to accumulator
7015		DCR	C	0D	Decrement C reg by 1
7016		JNZ	7014H	C2	Jump to label S1, if C reg is not zero
7017				14	Lower address byte
7018				70	Higher address byte
7019		INX	H	23	Increment HL by 1
701A		MOV	M,A	77	Move accumulator data to memory
701B	STOP	RST1		CF	Stop the program

12) A block of data is stored in memory locations from 7000H to 7004H. Write a program to transfer the data in reverse order to memory location starting from 7005H.

Data Before execution

Memory Location	Data
7000	05
7001	04
7002	03
7003	02
7004	01

Data After execution

Memory Location	Data
7005	01
7006	02
7007	03
7008	04
7009	05

Address	Label	Mnemonics		Hexcode	Comment
		Opcode	Operand		
700A	START	MVI	C,05H	0E	Store block length in C reg
700B				05	
700C		LXI	H,7004H	21	Initialize HL to last address of source data at 7004H
700D				04	Lower address byte
700E				70	Higher address byte
700F		LXI	D,7005H	11	Initialize DE to starting address of destination block 7005H
7010				05	Lower address byte
7011				70	Higher address byte
7012	LOOP	MOV	A,M	7E	Move memory data to accumulator
7013		STAX	D	12	Store accumulator data to memory pointed by DE
7014		INX	H	23	Decrement HL by 1
7015		DCX	D	1B	Increment DE by 1
7016		DCR	C	0D	Decrement counter C by 1
7017		JNZ	7012H	C2	Jump to 7012H if C data is not zero
7018				12	Lower address byte
7019				70	Higher address byte
701A	STOP	RST1		CF	Stop program