M.I.T. LAB Assignment – 09

U19CS012

1) The Given String Is Stored At Memory Location 1000 Onwards:
"Microprocessor And Interface" Ended With '0dh'.
Write 8085 Program To Count Occurrences Of Each Character In Given String.
Output Is Displayed From Memory Location 2000

Notepad Code:

```
; Initialise [H-L] to Point to First Memory Location
LXI H, 1000H

MAIN:   MOV A, M          ; A <- [M]
        CALL ASCII        ; Convert it to ASCII
        MOV A, M          ; A <- [M]
        INX H             ; Point to Next Character
        CPI 0DH           ; Compare with 14
        JNZ MAIN
HLT

; Get the Extra Number from 'A' [Eg: 'B' = 1, a = '32']
ASCII:  CPI 41H           ; ASCII OF A = 65 = (41)H & 'a' = 97 =
        RC                ; If White Space [ASCII = 32 = 02H] is Encountered,
                          ; [02H - 41H < 0 & C = 1]  then Return
        SUI 41H           ; Otherwise Subtract (41)H
        JMP STORE         ; Store it in M
RET

STORE:  LXI D, 2000H      ; Intial Location for Output
        MOV E, A          ; Increment E <- A like 2000 D = 20 & E = 00 => A
        LDAX D            ; Store the Frequency [DE] -> A
        INR A             ; Increment the Frequency A++
        STAX D            ; Store the Frequency back [DE] <- A
        RET
```

Input:

Start 1000h

Address (Hex)	Address	Data	
1000	4096	77	M
1001	4097	105	i
1002	4098	99	c
1003	4099	114	r
1004	4100	111	o
1005	4101	112	p
1006	4102	114	r
1007	4103	111	o
1008	4104	99	c
1009	4105	101	e
100A	4106	115	s
100B	4107	115	s

Start 1000h

Address (Hex)	Address	Data	
100C	4108	111	o
100D	4109	114	r
100E	4110	32	
100F	4111	65	A
1010	4112	110	n
1011	4113	100	d
1012	4114	32	
1013	4115	73	I
1014	4116	110	n
1015	4117	116	t
1016	4118	101	e
1017	4119	114	r

1018	4120	102	f
1019	4121	97	a
101A	4122	99	c
101B	4123	101	e
101C	4124	13	0D = 13 to end

Output:

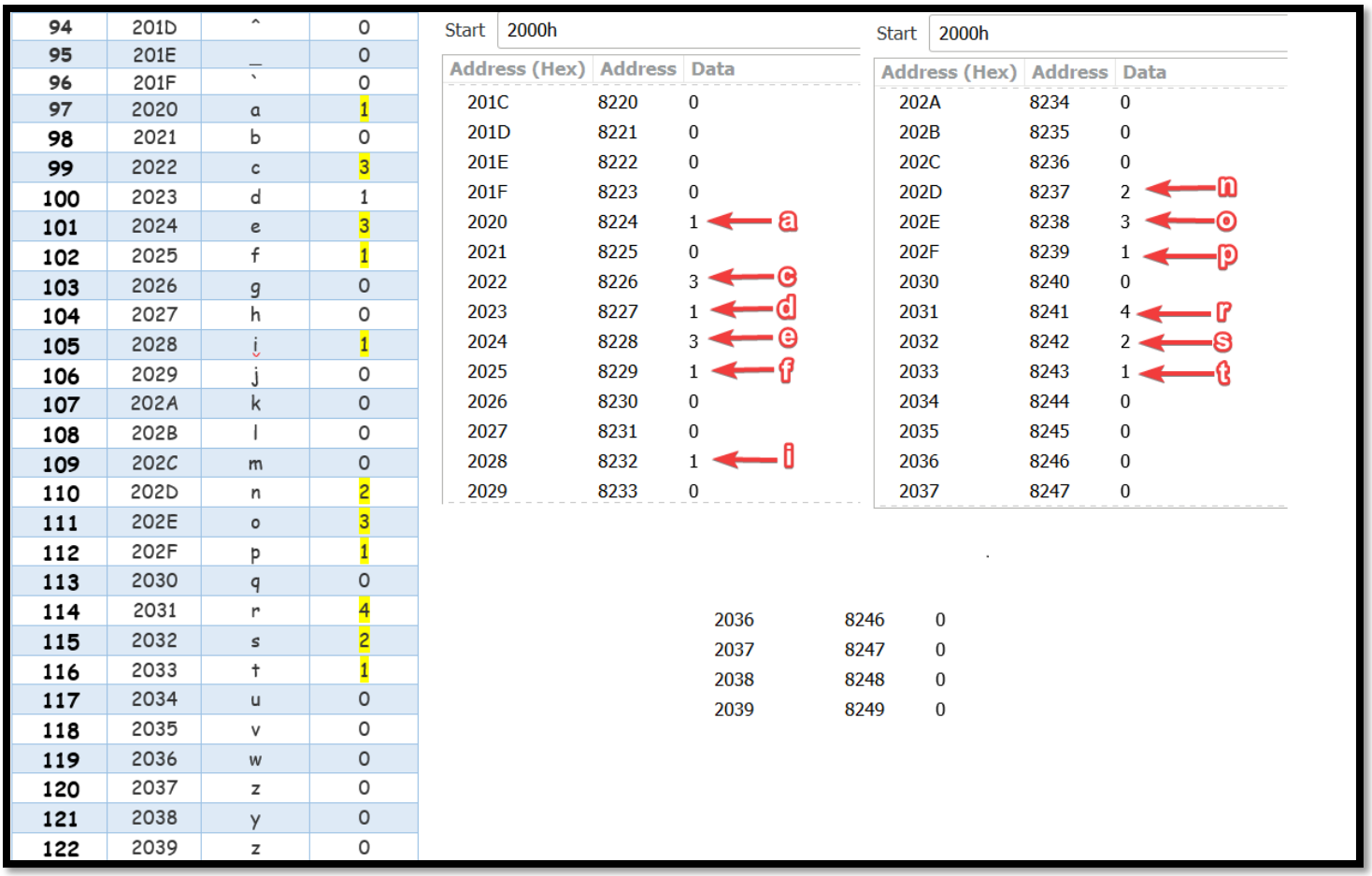
ASCII Code	Location [Hex]	Character	Frequency
65	2000	A	1
66	2001	B	0
67	2002	C	0
68	2003	D	0
69	2004	E	0
70	2005	F	0
71	2006	G	0
72	2007	H	0
73	2008	I	1
74	2009	J	0
75	200A	K	0
76	200B	L	0
77	200C	M	1
78	200D	N	0
79	200E	O	0
80	200F	P	0
81	2010	Q	0
82	2011	R	0
83	2012	S	0
84	2013	T	0
85	2014	U	0
86	2015	V	0
87	2016	W	0
88	2017	X	0
89	2018	Y	0
90	2019	Z	0
91	201A	[	0
92	201B	\	0
93	201C	]	0

Start 2000h

Address (Hex)	Address	Data
2000	8192	1 ← A
2001	8193	0
2002	8194	0
2003	8195	0
2004	8196	0
2005	8197	0
2006	8198	0
2007	8199	0
2008	8200	1 ← I
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	1 ← M
200D	8205	0

Start 2000h

Address (Hex)	Address	Data
200E	8206	0
200F	8207	0
2010	8208	0
2011	8209	0
2012	8210	0
2013	8211	0
2014	8212	0
2015	8213	0
2016	8214	0
2017	8215	0
2018	8216	0
2019	8217	0
201A	8218	0
201B	8219	0



94	201D	^	0
95	201E	_	0
96	201F	`	0
97	2020	a	1
98	2021	b	0
99	2022	c	3
100	2023	d	1
101	2024	e	3
102	2025	f	1
103	2026	g	0
104	2027	h	0
105	2028	i	1
106	2029	j	0
107	202A	k	0
108	202B	l	0
109	202C	m	0
110	202D	n	2
111	202E	o	3
112	202F	p	1
113	2030	q	0
114	2031	r	4
115	2032	s	2
116	2033	t	1
117	2034	u	0
118	2035	v	0
119	2036	w	0
120	2037	z	0
121	2038	y	0
122	2039	z	0

Start 2000h

Address (Hex)	Address	Data
201C	8220	0
201D	8221	0
201E	8222	0
201F	8223	0
2020	8224	1 ← a
2021	8225	0
2022	8226	3 ← c
2023	8227	1 ← d
2024	8228	3 ← e
2025	8229	1 ← f
2026	8230	0
2027	8231	0
2028	8232	1 ← i
2029	8233	0

Start 2000h

Address (Hex)	Address	Data
202A	8234	0
202B	8235	0
202C	8236	0
202D	8237	2 ← n
202E	8238	3 ← o
202F	8239	1 ← p
2030	8240	0
2031	8241	4 ← r
2032	8242	2 ← s
2033	8243	1 ← t
2034	8244	0
2035	8245	0
2036	8246	0
2037	8247	0

2036	8246	0
2037	8247	0
2038	8248	0
2039	8249	0

---

2) Write An 8085 Program To Check The Substring From Given String
Given String: "Hello World" & Substring: "Wor" => Output : Location of 'W'

Notepad Code:

```
; Q-2) Write An 8085 Program To Check The Substring From Given String
; Given String: "Hello World"      <= s
; Substring: "Wor"                 <= substr

LXI H, 2000H        ; Starting String Location
LDA 3000H           ; Substring to Find Location
MOV B, A            ; Store Starting Charater in B <- A

MAIN:   MOV A, M    ; Chech s[i] with 1st character of substr 'Wor' i.e. W
        CMP B
        CZ  CHECK   ; Check if it is Valid Substring
        MOV A, M
        INX H
        CPI 0DH     ; Check for End of String
        JNZ MAIN
HLT

CHECK:  PUSH H      ; Safely Store HL
        LXI D, 3000H ; Location of Sub-String

LOOP:   LDAX D          ; A = substr[j]
        CMP M           ; Compare the substr[j] & s[i]
        JNZ NOTEQUAL    ; If substr[j] != s[i] Then it can't be Substring
        INX D
        INX H
        LDAX D
        CPI 0DH
        JNZ LOOP
; Control Reaches Here -> Substring is Found
POP H
XCHG        ; [DE] <-> [HL]
INX H
INX H
MOV M, E    ; (lower nibble) Store s[i]'s Location to [end of substring + 2] Location
INX H
MOV M, D    ; (upper nibble)
HLT

NOTEQUAL:   POP H   ; Restore [HL] and Return
            RET
```

Input:

String

Start 2000h

Address (Hex)	Address	Data	
2000	8192	72	H
2001	8193	101	e
2002	8194	108	l
2003	8195	108	l
2004	8196	111	o
2005	8197	32	
2006	8198	87	W
2007	8199	111	o
2008	8200	114	r
2009	8201	108	l
200A	8202	100	d
200B	8203	13	0D = 13 to End
200C	8204	0	

Substring

Start 3000h

Address (Hex)	Address	Data	
3000	12288	87	W
3001	12289	111	o
3002	12290	114	r
3003	12291	13	0D = 13 to End

Output:

3003	12291	13	06H (lower bit)
3004	12292	0	
3005	12293	6	
3006	12294	32	
3007	12295	0	
3008	12296	0	20H (upper bit)
3009	12297	0	
300A	12298	0	FINAL ANS: 2006H

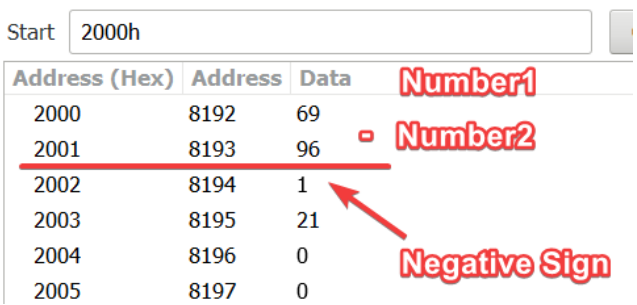
Explanation: 'Wor' Substring Starts at Location 2006 H Location in Main String

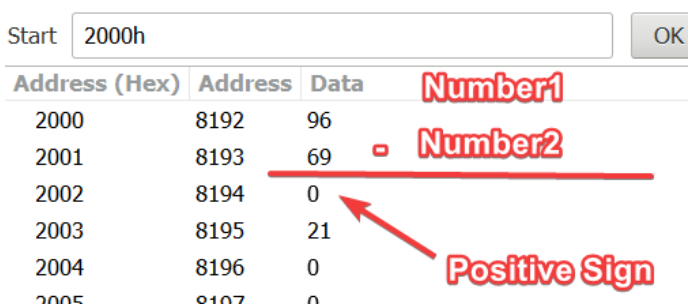
3) Write An Assembly Language Program in 8085 Microprocessor to Subtract Two 8 Bit BCD Numbers.

Notepad Code:

```
; (3) WAP to Subtract Two 8 Bit BCD Numbers.
; Input: Numbers are Stored at Location 2000H & 2001H

LXI H, 2000H
MOV B, M            ; First Number in 'B'
INX H
MOV C, M            ; Second Number in 'C'
INX H

; Procedure to Find B - C
MVI A, 99H          ; A <- 99 [Max BCD]
SUB C               ; A <- A - C
DAA                 ; Decimal Adjust [8 Bit in A -> 2 (4 Bit) BCD]
ADD B               ; A <- A + B
DAA                 ; Adjust to BCD
JNC COMP            ; If No Carry => Negative Number [Since A + (B - C) ]
ADI 01H             ; A <- A + 1
DAA                 ; Adjust to BCD
MVI M, 00H          ; Positive Sign
INX H
JMP STORE           ; Store the Final Answer

COMP:   MOV D, A
        MVI A, 99H
        SUB D
        MVI M, 01H      ; Negative Sign
        INX H
STORE: MOV M, A
HLT
```

Test Case:





Explanation:

Case1: Negative Result

Number1: 45 (in BCD) = $(0100\ 0101)_2$ = $(69)_{10}$ [Decimal]

Number2: 60 (in BCD) = $(0110\ 0000)_2$ = $(96)_{10}$ [Decimal]

Num1-Num2: -15 (in BCD) = $(0001\ 0101)_2$ = $-(21)_{10}$ [Decimal]

Case2: Positive Result

Number1: 60 (in BCD) = $(0110\ 0000)_2$ = $(96)_{10}$ [Decimal]

Number2: 45 (in BCD) = $(0100\ 0101)_2$ = $(69)_{10}$ [Decimal]

Num1-Num2: 15 (in BCD) = $(0001\ 0101)_2$ = $(21)_{10}$ [Decimal]

4) Write an Assembly Level Language Program to Convert 8 Bit BCD Number to its Respective ASCII Code.

Notepad Code:

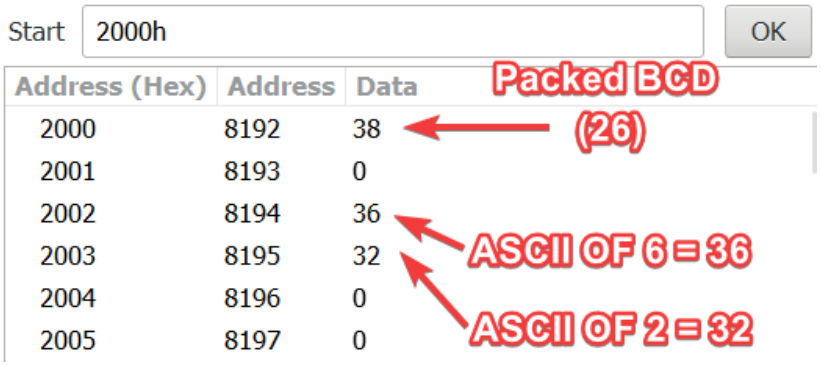
```
;(4) - WAP to Convert 8 Bit BCD Number to its Respective ASCII Code.

; Input Memory Location : 2000H
; Output Memory Location : 3000H & 3001

LXI H, 2000H        ; Packed BCD Number's Location
CALL BCD2BIN
INX H
INX H
MOV D, A            ; Storing Binary Number in 'D' [For Reference]

; Ones and Ten's Digit are in 'B' & 'C'
MOV A, C            ; A <- C
ADI 1EH             ; 1EH = 30 & DIGIT + 30 -> ASCII
MOV M, A            ; Tens Digit Stored
INX H               ; Next Location
MOV A, B            ; A <- B
ADI 1EH             ; Get ASCII by Adding 1EH
MOV M, A            ; Ones Digit Stored
HLT
```

```
BCD2BIN:    MOV A, M      ; [abcd efgh]
            ANI 0FH       ; Mask Upper 4 Bits [abcd efgh & 0000 1111]
            MOV C, A      ; Store it in C [0000 efgh]
            MOV A, M      ; [abcd efgh]
            ANI 0F0H      ; Mask Lower 4 Bits [abcd efgh & 1111 0000]
            RRC           ; [0abc d000]
            RRC           ; [00ab cd00]
            RRC           ; [000a bcd]
            RRC           ; [0000 abcd]
            MOV B, A      ; Tens Digit
            MVI D, 09H    ; A <- A*10
MUL:        ADD B
            DCR D
            JNZ MUL
            ADD C         ; Add Ones Digit at Last
RET
```

Test Case:

Input: Packed BCD = 26 = $(0010\ 0110)_2$ = $(38)_{10}$ [Decimal]

Output: ASCII OF 6 = 36 & ASCII OF 2 = 32



SUBMITTED BY:

BHAGYA VINOD RANA

[U19CS012]