Problems on object program generation of SIC/XE programs :-

I Generate the object code for each statement and write the object program for the following SIC/XE programs.

Given that,

CLEAR = B4, LDA = 00, LDB = 68, ADD = 18, TIX = 2C, JLT = 38, STA = 0C } opcode

LOC				Object Code
0000	SUM	START	0	
0000	FIRST	CLEAR	X	B410
0002		LDA	#0	010000
0005		+LDB	#TOTAL	69101788
0009		BASE	TOTAL	
	LOOP	ADD	TABLE, X	1BA00C
000C		TIX	COUNT	2F2006
000F		JLT	LOOP	3B2FF7
0012		STA	TOTAL	0F4000
0015	COUNT	RESW	1	
0018	TABLE	RESW	2000	
1788	TOTAL	RESW	1	
178B		END	FIRST	

① CLEAR X (reg-to-reg assembly)

opcode (8)	r_1 (4)	r_2 (4)
B4	1	0

} opcode of CLEAR is B4.

} Mnemonic of register X is 1 and second register is not there, hence 0 is taken

2) LDA #0 (Immediate addressing)

As immediate addressing is considered,

disp = 000

opcode (6)	n	i	x	b	p	e	disp (12)
0000 00	0	1	0	0	0	0	0000 0000 0000
0	1		0				0 0 0

3) +LDB #TOTAL

As this is Format 4 instruction, relative addressing will not apply. Directly address of operand is considered.

address = 01788 [ie., address of operand TOTAL]

opcode (6)	n	i	x	b	p	e	address (20)
0110 10	0	1	0	0	0	1	0000 0001 0111 1000 1000
6 9		1					0 1 7 8 8

4) ADD TABLE, X [Indexed addressing]
(Use PC relative)

TA = 0018
PC = 000C

∴ disp = TA − PC
= 000C

opcode (6)	n	i	x	b	p	e	disp (12)
0001 10	1	1	1	0	1	0	0000 0000 1100
1 B			A				0 0 C

5) TIX COUNT (PC relative)

TA = 0015

PC = 000F

disp = 006

opcode(6)	n	i	x	b	p	e	disp (12)
0010 11	1	1	0	0	1	0	0000 0000 0110
2 F			2				0 0 6

6) JLT LOOP

TA = 0009

PC = 0012

disp = 0009 − 0012

= −009 = FF7$_H$

2's complement

```
   0    0    9
0000 0000 1001
1111 1111 0110   (1's)
                1
1111 1111 0111   (2's)
F    F    7
```

opcode(6)	n	i	x	b	p	e	disp(12)
001110	1	1	0	0	1	0	1111 1111 0111
3 B			2				F F 7

7) STA TOTAL

Try PC relative first

TA = 1788

PC = 0015

disp = 1773$_{(H)}$ ⇒ 6003$_D$ [Range for PC relative $-2048_{(D)} \leq 0 \leq 2048$]

So, the instruction should be assembled using base relative-addressing.

Hence, TA = 1788 disp = 000

Base = 1788

opcode (6)	n	i	x	b	p	e	disp (12)
0000 11	1	1	0	1	0	0	0000 0000 0000
0 F			4				0 0 0

Object Program :-

H^ SUM ^ 000000 ^ 00178B

T^ 000000 ^ 0C ^ B410 ^ 010000 ^ 691017 88 ^ 1BA00C

T^ 00000C ^ 09 ^ 2F2006 ^ 3B2FF7 ^ 0F4000

E^ 000000

II Generate the complete object program for the following SIC/XE assembly program.

Given OPcodes :-

CLEAR - B4 , JEQ - 30 , WD - DC , JLT - 38 , TD - E0

LDT - 74 , LDCH - 50 , TIXR - B8 , RSUB - 4C

LOC				object code
405D	WRREC	START	405D	
405D		CLEAR	X	B410
405F		LDT	LENGTH	772FDH
4062	WLOOP	TD	OUTPUT	E32011
4065		JEQ	WLOOP	332FFA
4068		LDCH	BUFFER, X	53AFC8
406B		WD	OUTPUT	DF2008
406E		TIXR	T	B850
4070		JLT	WLOOP	3B2FEF
4073		RSUB		4F0000
4076	OUTPUT	BYTE	X'05'	05
4077		END		

Address of BUFFER - 4033

Address of LENGTH - 4036

1) CLEAR X

opcode (8)	r_1(4)	r_2(4)
1011 0100	1	0
B 4		

2) LDT LENGTH (PC relative)

disp = TA − (PC)
= 4036 − 4062
= −2C = FD4

```
  0 2 C
  0000 0010 1100
  1111 1101 0011
                 1
  ---------------
  1111 1101 0100
  ---------------
  F    D    4
```

opcode(6)	n	i	x	b	p	e	disp(12)
0111 01	1	1	0	0	1	0	1111 1101 0100
7 7			2				F D 4

3) TD OUTPUT (PC relative)

disp = TA − (PC)
= 4076 − 4065
= 011

opcode(6)	n	i	x	b	p	e	disp(12)
1110 00	1	1	0	0	1	0	0000 0001 0001
E 3			2				0 1 1

4) JEQ WLOOP (PC relative)

disp = TA - (PC)

= 4062 - 4068

= -6 = FFA

opcode(6)	n	i	x	b	p	e	disp(12)
0011 00	1	1	0	0	1	0	1111 1111 1001
3 3				2			F F A

object code - 332FFA.

5) LDCH BUFFER, X (PC relative & Indexed)

disp = TA - (PC)

= 4033 - 406B

= -38 = FC8

opcode(6)	n	i	x	b	p	e	disp(12)
0101 00	1	1	1	0	1	0	1111 1100 1000
5 3				A			F C 8

Object code - 53AFC8

6) WD OUTPUT

disp = TA - (PC)

= 4076 - 406E

= 8

opcode(6)	n	i	x	b	p	e	disp(12)
1101 11	1	1	0	0	1	0	0000 0000 1000
D F			2				0 0 8

object code - DF2008

7) TIXR T (reg-to-reg assembly)

opcode (8)	r_1 (4)	r_2 (4)
B8	5	0

object code - B850

8) JLT WLOOP (PC relative)

disp = TA - (PC)

= 4062 - 4073

= -11 = FEF

opcode (6)	n	i	x	b	p	e	disp (12)
0011 10	1	1	0	0	1	0	1111 1110 1111
3 B			2		F E F		

object code - 3B2FEF

9) RSUB (when no operand, disp is taken as 000)

opcode (6)	n	i	x	b	p	e	disp (12)
0100 11	1	1	0	0	0	0	0 000 0000 0000
4 F			0		0		0 0 0

object code - 4F0000

Object Program:-

H^ WRREC ^ 00405D ^ 00001A

T^ 00405D ^ 08 ^ 0B 410 ^ 772FD4 ^ E32011

T^ 004065 ^ 09 ^ 332FFA ^ 53AFC8 ^ DF2008

T^ 00406E ^ 09 ^ 00B850 ^ 3B2FEF ^ 4F0000 ^ 05

E^ 00405D

III Generate the object code for each statement and write the object program for the following SIC/XE program. Given that ADD - 18, LDA - 00, LDX - 04, STA - 0C, JLT - 38, LDB - 68, RSUB - 4C, TIX - 2C

LOCCTR	LABEL	OPCODE	OPERAND	OBJECT CODE
	SUM	START	0	
0000	FIRST	LDX	#0	050000
0003		LDA	#100	010064
0006		+LDB	# TABLE2	69101790
		BASE	TABLE2	
000A	LOOP	ADD	TABLE, X	1BA013
000D		ADD	TABLE2, X	1BC000
0010		TIX	COUNT	2F200A
0013		JLT	LOOP	3B2FF4
0016		+STA	TOTAL	0F102F00
001A		RSUB		4F0000
001D	COUNT	RESW	1	
0020	TABLE	RESW	2000	(3 x 2000 = 6000_D = 1770_H)
1790	TABLE2	RESW	2000	(∴ add 1770 to locctr)
2F00	TOTAL	RESW	1	
2F03		END	FIRST	

1) LDX #0 (Immediate addressing)

disp = 000

opcode(6)	n	i	x	b	p	e	disp(12)
000001	0	1	0	0	0	0	0000 0000 0000
0 5			0				0 0 0

object code - 050000

2) LDA #100 (Immediate addressing)

$100_D = 064_H$

opcode(6)	n	i	x	b	p	e	disp (12)
0000 00	0	1	0	0	0	0	0000 0110 0100

object code - 010064

3) +LDB #TABLE2 [Format 4, b=0, P=0, e=1]

address / disp = TA = 01790

opcode(6)	n	i	x	b	p	e	address (20)
0110 10	0	1	0	0	0	1	0000 0001 0111 1001 0000

object code - 69101790

4) ADD TABLE, X (PC relative, Indexed)

disp = TA - (PC)
= 0020 - 000D
= 013

opcode(6)	n	i	x	b	p	e	disp(12)
0001 10	1	1	1	0	1	0	0000 0001 0011

object code - 1BA013

5) ADD TABLE2, X [As TABLE2 address is associated with Base register, base relative assembly is used]

disp = TA - [B] [If PC relative is tried, disp would be out of range]
= 1790 - 1790
= 000

opcode(6)	n	i	x	b	p	e	disp(12)
0001 10	1	1	1	1	0	0	0000 0000 0000

object code → 1BC000

6) TIX COUNT (PC relative assembly)

disp = TA - [PC]

= 001D - 0013

= 00A

opcode(6)	n	i	x	b	p	e	disp(12)
0010 11	1	1	0	0	1	0	0000 0000 1010

object code - 2F200A

7) JLT LOOP [PC relative assembly]

disp = TA - [PC]

= 000A - 0016

= -00C = FF4

opcode(6)	n	i	x	b	p	e	disp(12)
0011 10	1	1	0	0	1	0	1111 1111 0100

object code - 3B2FF4

8) + STA TOTAL [Format 4 instruction, ∴ b=0, P=0, e=1]

address = 02F00

opcode(6)	n	i	x	b	p	e	address (20)
0000 11	1	1	0	0	0	1	0000 0010 1111 0000 0000

object code - 0F102F00

9) RSUB [As no operand, disp is 000] & [b=0, P=0]

opcode (6)	n	i	x	b	p	e	disp(12)
0100 11	1	1	0	0	0	0	0000 0000 0000

object code - 4F0000

Object Program :-

H^SUM^000000^002F03

T^000000^1D^050000^010000^691017 90^1BA013^
1BC000^2F200A^3B2FF4^0F102F00^4F0000

E^000000

IV Generate the object Program for the following SIC/XE program.

LOCCTR	LABEL	OPCODE	OPERAND	OBJECT CODE
1000	COPY	START	1000	
1000	CLOOP	+JSUB	RDREC	4B101157
1004		LDA	LENGTH	03214A
1007		COMP	ZERO	932141
100A		JEQ	EXIT	B32003
100D		J	CLOOP	BB2FF0
1010	EXIT	STA	BUFFER	532009
1013		LDA	THREE	032138
1016		STA	TOTAL_LENGTH	53213B
1019		RSUB		4F0000
101C	BUFFER	RESW	100 $(3*100=300_d=12C_H)$	-
1148	EOF	BYTE	C'EOF'	454F46
114B	ZERO	WORD	0	000000
114E	THREE	WORD	3	000003
1151	LENGTH	RESW	1	-
1154	TOTAL_LENGTH	RESW	1	-
1157	RDREC	LDX	ZERO	632FF1
115A		END	COPY	-

Given opcodes :-

JSUB - 48, LDA - 00, LDX - 60, STA - 50, COMP - 90, RSUB - 4C,
JEQ - B0, J - B8

1) +JSUB RDREC (Format 4, b=0, p=0, e=1)

address = TA = 01157

opcode (6)	n	i	x	b	p	e	address (20)
0100 10	1	1	0	0	0	1	0000 0001 0001 0101 0111

Object Code → 4B101157

2) LDA LENGTH (PC relative)

disp = TA − (PC)

= 1151 − 1007

= 14A

opcode(6)	n	i	x	b	p	e	disp(12)
0000 00	1	1	0	0	1	0	0001 0100 1010

object code - 03214A

3) COMP ZERO

disp = TA − (PC) (PC relative)

= 114B − 100A

= 141

opcode(6)	n	i	x	b	p	e	disp(12)
1001 00	1	1	0	0	1	0	0001 0100 0001

object code - 932141

4) JEQ EXIT (PC relative)

disp = TA − (PC)

= 1010 − 1000

= 003

opcode(6)	n	i	x	b	p	e	disp(12)
1011 00	1	1	0	0	1	0	0000 0000 0011

object code - B32003

5) J CLOOP (PC relative)

disp = TA − (PC)

= 1000 − 1010

= FF0

opcode(6)	n i x b p e	disp(12)
1011 10	1 1 0 0 1 0	1111 1111 0000

object code − BB2FF0

6) STA BUFFER

disp = TA − (PC)

= 101C − 1013 = 009

opcode(6)	n i x b p e	disp(12)
0101 00	1 1 0 0 1 0	0000 0000 1001

object code − 532009

7) LDA THREE

disp = TA − (PC)

= 114E − 1016 = 138

opcode(6)	n i x b p e	disp(12)
0000 00	1 1 0 0 1 0	0001 0011 1000

object code − 032138

8) STA TOTAL_LENGTH

disp = TA − (PC)

= 1154 − 1019 = 13B

opcode(6)	n i x b p e	disp(12)
0101 00	1 1 0 0 1 0	0001 0011 1011

object code − 53213B

9) RSUB

opcode (6)	n	i	x	b	p	e	disp (12)
0100 11	1	1	0	0	0	0	0000 0000 0000
4 F			0				0 0 0

object code - 4F0000

10) LDX ZERO

disp = TA - (PC)

= 114B - 115A = FF1

opcode(6)	n	i	x	b	p	e	disp(12)
0110 00	1	1	0	0	1	0	1111 1111 0001

object code - 632FF1

Object Program :-

H^ COPY ^ 001000 ^ 00015A

T^ 001000 ^ 19 ^ 4B101157 ^ 03214A ^ 932141 ^ B32003 ^

BB2FF0 ^ 532009 ^ 032138 ^ 53213B

T^ 001019 ^ 0F ^ 4F0000 ^ 454F46 ^ 000000 ^ 000003 ^ 632FF1

E^ 001000

Write a SIC/XE program to copy the string "COMPUTER SCIENCE ENGINEERING" from STR1 to another string STR2.

```
        LDT   #28
        LDX   #0
        LDCH  STR1,X
        STCH  STR2,X
        TIXR  T
        JLT   LOOP
        ⋮
STR1    BYTE  C'COMPUTER SCIENCE ENGINEERING'
STR2    RESB  28
```

Write a SIC/XE program to read 100byte record from a device 'F5' into BUFFER. Use immediate and register-to-register instructions.

```
        LDT   #100
        LDX   #0
CLOOP   TD    INDEV
        JEQ   CLOOP
        RD    INDEV
        STCH  BUFFER,X
        TIXR  T
        JLT   CLOOP
        ⋮
INDEV   BYTE  X'F5'
BUFFER  RESB  100
```

Generate the target address for the following object codes.

Given, X = 000090 PC = 003000
B = 006000

i) 032600

opcode(6)	n	i	x	b	p	e	disp(12)
0000 00	1	1	0	0	1	0	0010 0000 0000

P = 1, hence PC relative

TA = disp + (PC)
= 600 + 003000

TA = 003600 //

ii) 010030

opcode(6)	n	i	x	b	p	e	disp(12)
0000 00	0	1	0	0	0	0	0011 0000

n = 0, i = 1, b = 0, p = 0, hence neither PC relative nor base-relative

As i = 1 & n = 0, immediate addressing is used.

TA = disp

TA = 030 //

iii) 03C300

opcode(6)	n	i	x	b	p	e	disp(12)
0000 00	1	1	1	1	0	0	0011 0000 0000

n = 1, i = 1, x = 1, b = 1 (Base relative & Indexed addressing)

TA = disp + (B) + (X)
= 300 + 006000 + 000090

TA = 006390