

**EXPERIMENT NO : 13****ADDITION OF TWO 16 BIT NUMBERS**  
**USING 8086 TRAINER KIT****AIM**

To add two 16-bit numbers using 8086 trainer kit.

**ALGORITHM**

1. Clear the AX by performing AND operation with 0000
2. Move the location where result is to be stored to BX
3. Move the location of operand 1 to SI
4. Move the location of operand 2 to DI
5. Move the contents of SI to AX
6. Add the contents of DI to AX
7. Move the result to the location stored in BX
8. Move 0000H to AX
9. Add the carry flag to AX
10. Move the result to the location stored in [BX + 2]
11. Halt

**PROGRAM**

ADDRESS	MNEMONICS
0400	AND AX,0000H
0403	MOV BX,0600H
0406	MOV SI,0500H
0409	MOV DI,0550H
040C	MOV AX,[SI]
040E	ADD AX,[DI]
0410	MOV [BX],AX
0412	MOV AX,0000H
0415	ADC AX,0000H
0418	MOV [BX+2],AX
041B	HLT

**INPUT**

0500 - B5

0501 - 7A

0550 - 2A

0551 – E5

**OUTPUT**

0600 - DF

0601 - 5F

0602 - 01

**RESULT**

**EXPERIMENT NO : 14**

**SUBTRACTION OF TWO 16 BIT NUMBERS**  
**USING 8086 TRAINER KIT**

**AIM**

To subtract two 16-bit numbers using 8086 trainer kit.

**ALGORITHM**

1. Clear the carry flag
2. Move the location where result is to be stored to BX
3. Move the location of operand 1 to SI
4. Move the location of operand 2 to DI
5. Move the contents of SI to AX
6. Subtract the contents of DI from AX including the borrow value
7. Move the result to the location stored in BX
8. Halt

**PROGRAM**

ADDRESS	MNEMONICS
0400	CLC
0401	MOV BX,0900H
0404	MOV SI,0700H
0407	MOV DI,0800H
040A	MOV AX,[SI]
040C	SBB AX,[DI]
040E	MOV [BX],AX
0410	HLT

**INPUT**

0700 - 18  
 0701 - 08  
 0800 - 40  
 0801 - 10

**OUTPUT**

0900 - D8

0901 - F7

**RESULT**

**EXPERIMENT NO : 15****MULTIPLICATION OF TWO 16 BIT NUMBERS**  
**USING 8086 TRAINER KIT****AIM**

To multiply two 16-bit numbers using 8086 trainer kit.

**ALGORITHM**

1. Clear the carry flag
2. Move the location where result is to be stored to BX
3. Move the location of operand 1 to SI
4. Move the location of operand 2 to DI
5. Move the contents of SI to AX
6. Move the contents of DI to CX
7. Multiply CX to AX
8. Move the result from AX to the location stored in BX
9. Move the higher bits of result from DX to the location stored in [BX+2]
10. Halt

**PROGRAM**

ADDRESS	MNEMONICS
0400	CLC
0401	MOV BX,0700H
0404	MOV SI,0750H
0407	MOV DI,0800H
040A	MOV AX,[SI]
040C	MOV CX,[DI]
040E	MUL CX
0410	MOV [BX],AX
0412	MOV [BX+2],DX
0415	HLT

**INPUT**

0750 - 1A

0751 - 2B

0800 - 4B

0801 - 12

**OUTPUT**

0700 - 9E

0701 - 74

0702 - 14

0703 - 03

**RESULT**

**EXPERIMENT NO : 16****DIVISION OF A 16 BIT NUMBER BY AN 8 BIT NUMBER  
USING 8086 TRAINER KIT****AIM**

To divide a 16-bit number by an 8 bit number using 8086 trainer kit.

**ALGORITHM**

1. Clear the carry flag
2. Move the location where result is to be stored to BX
3. Move the location of operand 1 to SI
4. Move the location of operand 2 to DI
5. Move the contents of SI to AX
6. Move the contents of DI to CX
7. Move 00 to CH
8. Divide CL from AX
9. Move the result from AX to the location stored in BX
10. Halt

**PROGRAM**

ADDRESS	MNEMONICS
0400	CLC
0401	MOV BX,0700H
0404	MOV SI,0750H
0407	MOV DI,0800H
040A	MOV AX,[SI]
040C	MOV CX,[DI]
040E	MOV CH,00H
0410	DIV CL
0412	MOV [BX],AX
0414	HLT

**INPUT**

0750 - 43

0751 - 12

0800 - 21

### **OUTPUT**

0700 - 8D     (Quotient)

0701 - 16     (Remainder)

### **RESULT**



**EXPERIMENT NO : 17****MAXIMUM OF N NUMBERS**  
**USING 8086 TRAINER KIT****AIM**

To find the maximum of n numbers using the 8086 trainer kit.

**ALGORITHM**

1. Clear the carry flag
2. Move the location where the result has to be stored to BX
3. Move the starting location of array to SI
4. Move the total number of elements in the array to CX
5. Move 00 to AL
6. Compare the contents of SI with AL
7. Jump to step 9 if above instruction satisfies
8. Else move the contents of SI to AL
9. Move 00 to CH
10. Increment SI
11. Continue the loop of comparing the contents of SI and AL till the counter reaches zero (LOOPNZ only loops when the zero flag is not set)
12. Move the result, ie, maximum number from AL to the location stored in BX
13. Halt

**PROGRAM**

ADDRESS	MNEMONICS
0400	CLC
0401	MOV BX,0700H
0404	MOV SI,0800H
0407	MOV CX,0005H
040A	MOV AL,00H
040C	CMP AL,[SI]
040E	JA 0412H
0410	MOV CH,00H
0412	INC SI

0413	LOOPNZ 040CH
0415	MOV [BX],AL
0417	HLT

### **INPUT**

0800 - 77  
0801 - 81  
0802 - B4  
0803 - F1  
0804 - AB

### **OUTPUT**

0700 - F1

### **RESULT**

**EXPERIMENT NO : 18**

**SORTING NUMBERS IN ASCENDING ORDER**  
**USING 8086 TRAINER KIT**

**AIM**

To sort the numbers in ascending order using 8086 trainer kit.

**ALGORITHM**

1. Set the value of SI to 500.
2. Load data from offset SI to register CL.
3. Decrease value of register CL by 1.
4. Set the value of SI to 500.
5. Load data from offset SI to register CH. Decrease value of register CH by 1
6. Increase the value of SI by 1.
7. Load value from offset SI to register AL.
8. Increase the value of SI by 1.
9. Compare the value of register AL and [SI] ,ie,(AL-[SI]).
10. Jump to address 41C if carry is generated.
11. Exchange the contents of register AL and SI.
12. Decrease the value of SI by 1.
13. Exchange the contents of register AL and SI
14. Increase the value of SI by 1.
15. Decrease the value of register CH by 1.
16. Jump to address 40F if zero flat reset
17. Decrease the value of register CL by 1.
18. Jump to address 407 if zero flat reset.
19. Stop

**PROGRAM**

ADDRESS	MNEMONICS
0400	MOV SI,500
0403	MOV CL,[SI]
0405	DEC CL

0407	MOV SI,500
0409	MOV CH,[SI]
040C	DEC CH
040E	INC SI
040F	MOV AL,[SI]
0411	INC SI
0412	CMP AL,[SI]
0414	JC 041C
0416	XCHG AL,[SI]
0418	DEC SI
0419	XCHG AL,[SI]
041B	INC SI
041C	DEC CH
041E	JNZ 40F
0420	DEC CL
0422	JNZ 407
0424	HLT

### **INPUT**

0500 - 5  
0501 - 6  
0502 - 8  
0503 - 3  
0504 - 5  
0505 - 4

### **OUTPUT**

0500 - 5  
0501 - 3  
0502 - 4  
0503 - 5  
0504 - 6  
0505 - 8

### **RESULT**