Experiment 4: Learn assembly programming by practicing simple programs including average calculation of 3/4/5/more numbers, calculate area of a rectangle and a triangle, temperature conversion from °C to °F, conversion from °F to °C, conversion from °C to °K, conversion from °K to °C and counting tiles problems.

Problem 1: Temperature conversion from °C to °K let, temperature = 39°C 1°K=1°C + 273

MOV AX, 39
MOV BX, 273
ADD AX, BX
INT 3

Output:	AX=0138 CX=0000	BX=0111 DX=0000
Result Verification:	K= 39+273= 312	

Discussion: We know, °K= °C + 273
At first, 27 loaded in AX register and the address is 0404 and 111 replaced in BX register and its address is 0407. Now, AX, BX are added in address 041A. After pressing STP and REG, it shows the result.

INT 3: INT 3 is a special one-byte instruction having op-code is CCH. that is inserted by debuggers at the instruction where the user has set a breakpoint to occur. When it's hit, the interrupt handler breaks into the debugger and then replaces the original instruction so that execution can proceed when the user is ready.

Merge Problem 1 and Problem 2 and show the students about the task done by the INT3

Problem 2: Temperature conversion from °K to °C
let, temperature = 270°K

```
MOV   AX, 270
MOV   BX, 273
SUB   AX, BX
INT    3
```

Output:	AX=FFFD BX=0111 CX=0000 DX=0000
Result Verification:	C= 270-273= -3 = FFFDH

Discussion: We know, °C = °K – 273
At first, 10E replaced in Ax register and the address is 0404 and 111 replaced in Bx register and its address is 0407. Now, Ax, Bx are subtract in address040A.After pressing STP and REG, it shows the result.

Problem 5: Floor size 20*20, Tiles size 2*2. How many tiles are needed to cover up the floor?

```
MOV   AX, 20
MOV   BX, 20
MUL   BL
MOV   CX, AX
MOV   AX, 2
MOV   BX, 2
MUL   BL
MOV   BX, AX
MOV   AX, CX
DIV    BL
INT 3
```

Output:	AX=0064 BX=0004 CX=0190 DX=0000
Result Verification:	Tiles = (20*20)/ (2*2) = 400/4= 100 = 64H

Problem 6: Factorial Operation: 5! – 3!

```
MOV   AX, 1
MOV   CL, 5
L1:      MUL CL
LOOP L1
MOV  DX, AX
MOV  AX, 1
MOV  CL, 3
L2:   MUL CL
LOOP   L2
MOV   BX, AX
```

```
MOV   AX, DX
SUB    AX, BX
INT    3
```

Output:	AX=0072 BX=0006 CX=0000 DX=0078
Result Verification:	5! – 3! = 114 = 72H; AH = 00, AL = 72

Discussion:
At first, we load 1 in AX register and load 5 in CL register then do multiply by giving loop with CL address and move AX value in DX register. Now, again entered value 1 in AX register and 3 replaced in CL register then do multiply by giving loop with CL address and move AX value in BX register. Then move the DX value in AX register and do subtraction of AX and BX. After pressing STP and REG, it produces the result.

Problem 7: (5! / 3!) + 4!

```
MOV   AX, 1
MOV  CL, 5
L1:MUL   CL
LOOP  L1
MOV   DX, AX
MOV   AX, 1
MOV   CL, 3
L2:MUL   CL
LOOP  L2
MOV   BX, AX
MOV   AX, DX
DIV    BL
MOV  DX, AX
MOV  AX, 1
MOV  CL, 4
L3:      MUL   CL
LOOP  L3
ADD   AX,DX
INT    3
```

Output:	AX=002C BX=0006 CX=0000 DX=0014

Result Verification:	(5! / 3!)+4!=(120/6)+24=20+24=44=2C H, AH=00, AL=2C

Discussion:
At first, load 1 in AX register and load 5 in CL register then do multiply by giving loop with CL register and move AX value in DX register. Again, load value 1 in AX register and 3 in CL register then do multiply by giving loop with CL register and move AX value in BX register. Then move the DX value in AX register and do division by BL. Now, move AX value in DX and again entered value 1 in AX register and 4 replaced in CL register then do multiply by giving loop with CL address. At last, we do addition of DX and AX. After pressing STP and REG, we get the result.

Problem 10: Byte with Byte Division

```
ORG 100h
.MODEL SMALL
.DATA
NUM_1 DB 0F2H
NUM_2 DB 4H
.CODE
MOV BH, NUM_2     ;Load numerator in BH
MOV AL, NUM_1     ;Load denominator in AL
DIV BH            ;Divide BH by AL
RET
```

Output: The DIV instruction divides BH by AL. F2 divided by 04 gives quotient of 3C and give 02 as a remainder. AL stores the quotient and remainder is stored in AH register.	AX=023C

- ORG (abbr. for ORiGin) is an assembly directive (not an instruction). It defines where the machine code (translated assembly program) is to place in memory. As for ORG 100H this deals with 80x86 COM program format (COMMAND) which consist of only one segment of max. 64k bytes. 100H says that the machine code starts from address (offset) 100h in this segment, effective address is CS:100H.
- With .model small you get a program where CS points to a 64k bytes code segment and DS point to 64k bytes data segment. Thus, code and data both use 64k bytes maximum space.

```
.MODEL MEDIUM     ;the data must fit into 64K bytes
                  ;but the code can exceed 64K bytes of memory
.MODEL COMPACT    ;the data can exceed 64K bytes
                  ;but the code cannot exceed 64K bytes
.MODEL LARGE      ;both data and code can exceed 64K
                  ;but no single set of data should exceed 64K
.MODEL HUGE       ;both code and data can exceed 64K
                  ;data items (such as arrays) can exceed 64K
.MODEL TINY       ;used with COM files in which data and code
                  ;must fit into 64K bytes
```

Problem 11: Word with Word Division

```
ORG 100h
.MODEL SMALL
.DATA
NUM_1 DW 0F213H
NUM_2 DW 41A8H
.CODE
MOV AX, NUM_1     ;Load numerator in AX
DIV NUM_2         ;Divide AX by NUM_2
RET
```

Output:

The output window shows that the division of F213H by 41A8 gives the remainder of 2D1B into DX register and 03 as a quotient into AX.

AX=0003

DX=2D1B

Conclusion:

In this experiment, we have learnt conversion from °C to °F, conversion from °F to °C, conversion from °C to °K, Conversion from °K to °C, Average of 3 numbers, average of 5 numbers, area of rectangle, area of triangle, find how many tiles. After performing those operation, we use assembly language in 8086 microprocessors which results in getting the correct output.