INSTITUTE OF COMPUTER TECHNOLOGY

B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

SUBJECT: MICROCONTROLLER & APPLICATION

NAME: Rahul Prajapati

ENRLL. NO: 23162171020

BRANCH: CYBER SECURITY

BATCH: 52

PRACTICAL_5

Aim:- Exercise for jump instruction.

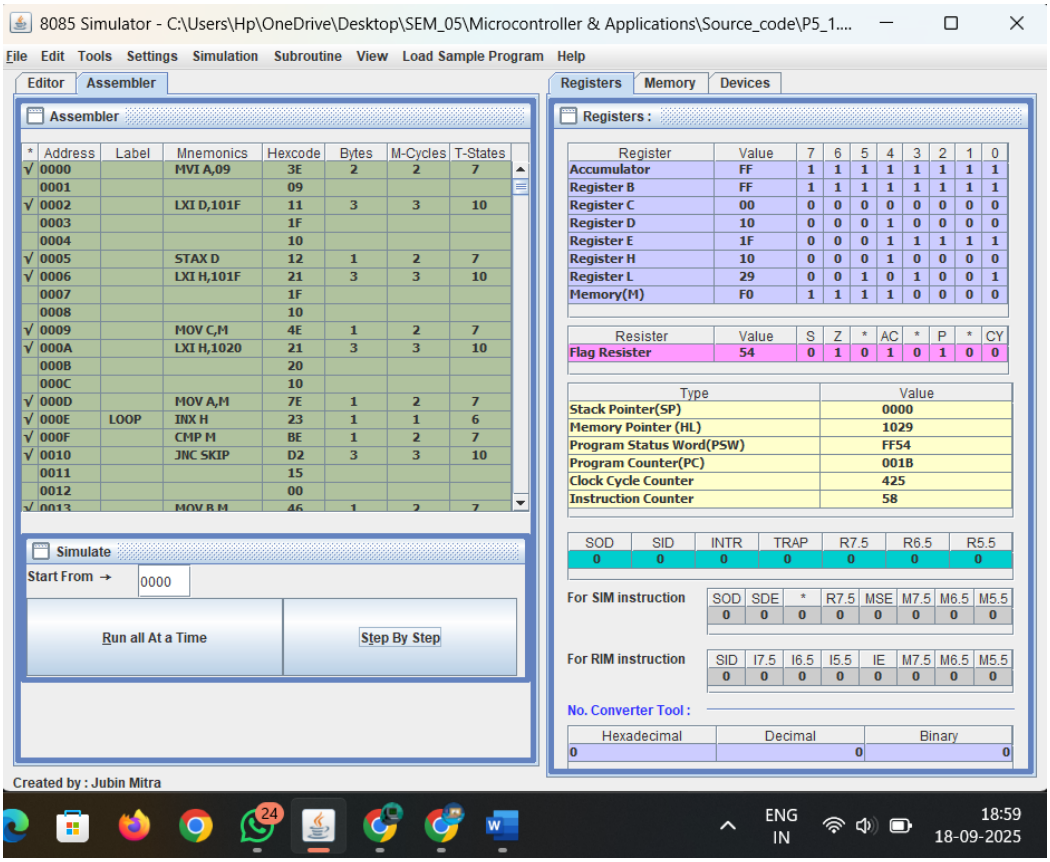
1. Find out the maximum number out of a block of data stored from memory location 1020H onwards. The size of block is stored at location 101FH. Display the maximum number on port 10H.

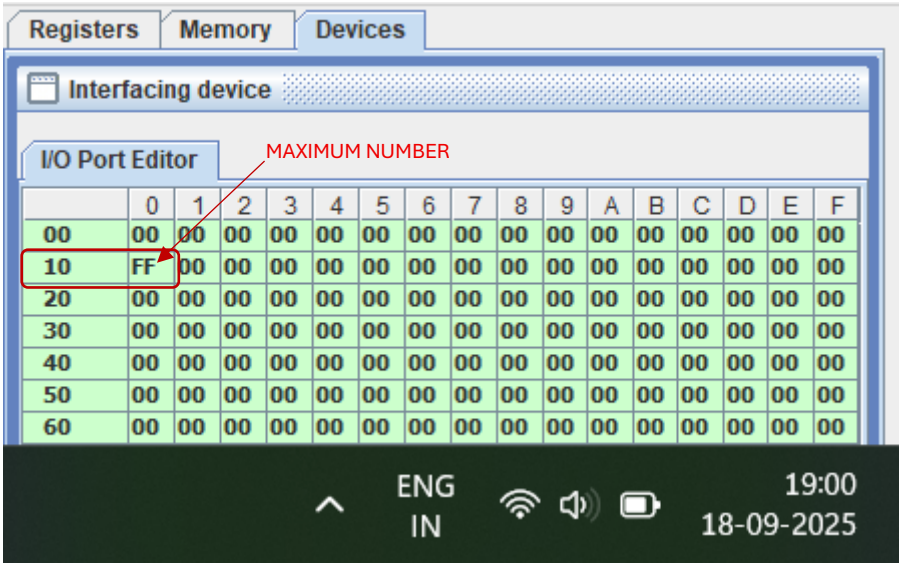
```
# ORG 1020H
# DB 45H,C4H,FFH,72H,A0H,E3H,99H,15H,85H,F0H
# ORG 0000H
                MVI A,09
                LXI D,101F
                STAX D
                LXI H,101F
                MOV C,M
                LXI H,1020 // SET HL PAIR FOR START FROM FIRST DATA
                MOV A,M    // LOAD FIRST DATA ASSUME ITS TEMPORARY
MAX

LOOP:           INX H      // INCREMENT HL PAIR FOR STARTING
COMARISON FROM SECOND DATA
                CMP M      // COMPARE MEMORY DATA WITH
ACCUMULATOR
                JNC SKIP   // JUMP ON SKIP IF MEMORY DATA IS NOT
GREATER THAN A
                MOV B,M    // IF DATA IS GREATER THEN MOVE IT INTO REG
B ITS CARRY OUR MAXIMUM NUMBER
                MOV A,M    // UPDATE TEMPORARY MAX

SKIP:           DCR C      // DECREMENT COUNTER
                JNZ LOOP   // JUMP ON LOOP
                OUT 10     // STORE OUR MAX DATA IN PORT 10H

HOLD:           HLT
```


8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\Source_code\P5_1....
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Assembler
Address Label Mnemonics Hexcode Bytes M-Cycles T-States
0000 MVI A,09 3E 2 2 7
0001 09
0002 LXI D,101F 11 3 3 10
0003 1F
0004 10
0005 STAX D 12 1 2 7
0006 LXI H,101F 21 3 3 10
0007 1F
0008 10
0009 MOV C,M 4E 1 2 7
000A LXI H,1020 21 3 3 10
000B 20
000C 10
000D MOV A,M 7E 1 2 7
000E LOOP INX H 23 1 1 6
000F CMP M BE 1 2 7
0010 JNC SKIP D2 3 3 10
0011 15
0012 00
Simulate
Start From →
0000
Run all At a Time
Step By Step
Created by : Jubin Mitra
Registers Memory Devices
Registers :
Register Value 7 6 5 4 3 2 1 0
Accumulator FF 1 1 1 1 1 1 1 1
Register B FF 1 1 1 1 1 1 1 1
Register C 00 0 0 0 0 0 0 0 0
Register D 10 0 0 0 1 0 0 0 0
Register E 1F 0 0 0 1 1 1 1 1
Register H 10 0 0 0 1 0 0 0 0
Register L 29 0 0 1 0 1 0 0 1
Memory(M) F0 1 1 1 1 0 0 0 0
Resister Value S Z * AC * P * CY
Flag Resister 54 0 1 0 1 0 1 0 0
Type Value
Stack Pointer(SP) 0000
Memory Pointer (HL) 1029
Program Status Word(PSW) FF54
Program Counter(PC) 001B
Clock Cycle Counter 425
Instruction Counter 58
SOD SID INTR TRAP R7.5 R6.5 R5.5
0 0 0 0 0 0 0
For SIM instruction
SOD SDE * R7.5 MSE M7.5 M6.5 M5.5
0 0 0 0 0 0 0 0
For RIM instruction
SID I7.5 I6.5 I5.5 IE M7.5 M6.5 M5.5
0 0 0 0 0 0 0 0
No. Converter Tool :
Hexadecimal Decimal Binary
0 0 0
ENG IN
18:59
18-09-2025


Registers Memory Devices
Interfacing device
I/O Port Editor
MAXIMUM NUMBER
0 1 2 3 4 5 6 7 8 9 A B C D E F
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ENG IN
19:00
18-09-2025

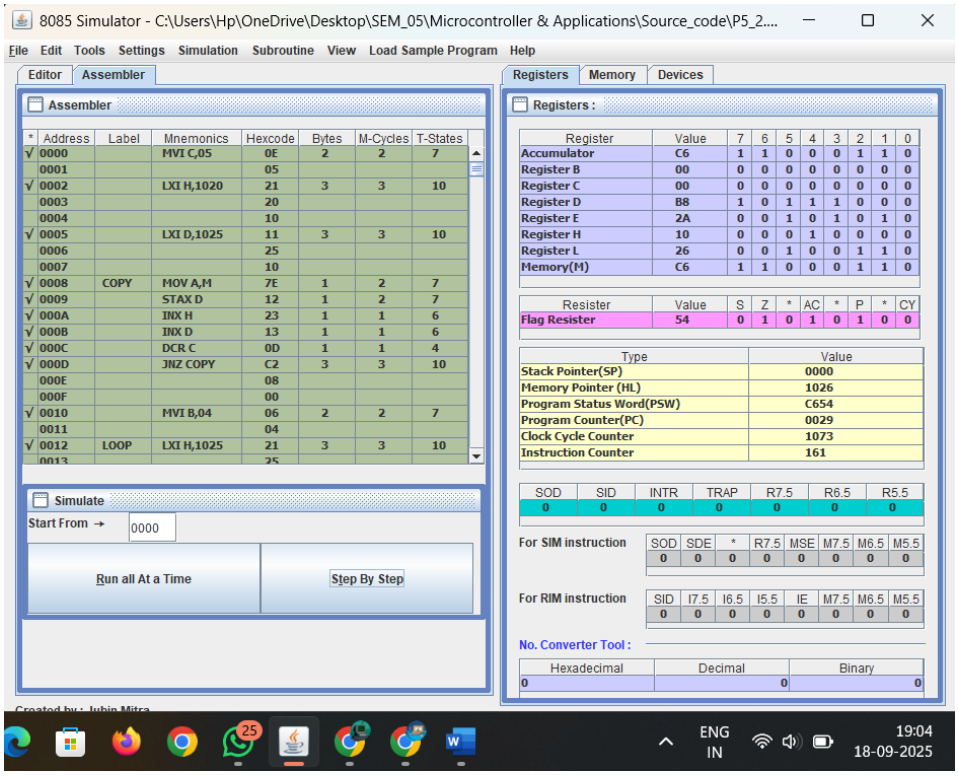
2. Arrange 5 bytes of data into ascending order. The bytes are to be loaded in memory using assembler directives. The sorted data must be stored in the memory location next to the last byte.

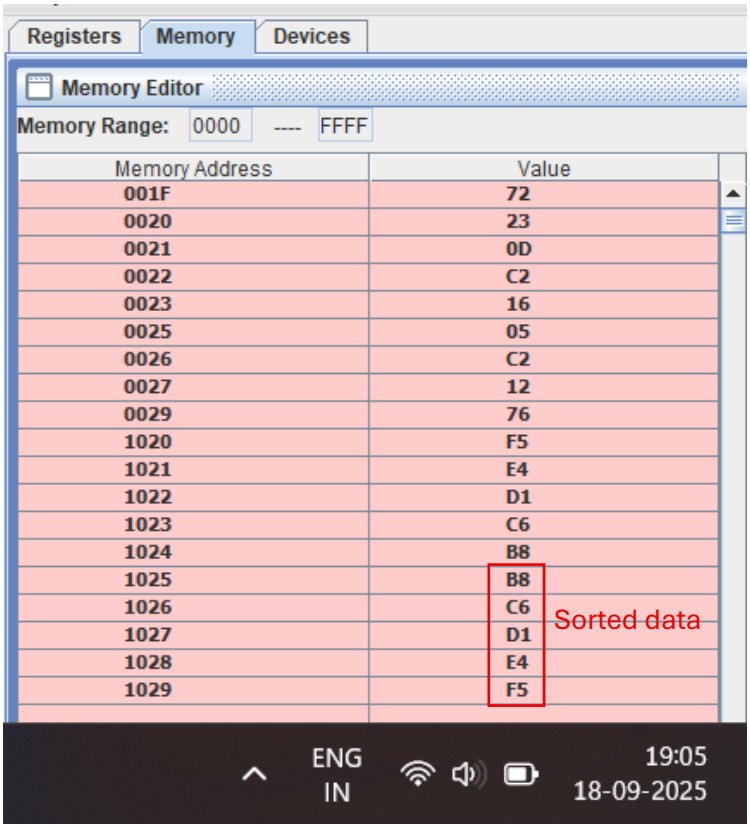
Editor | Assembler

8085 Assembly Language Editor

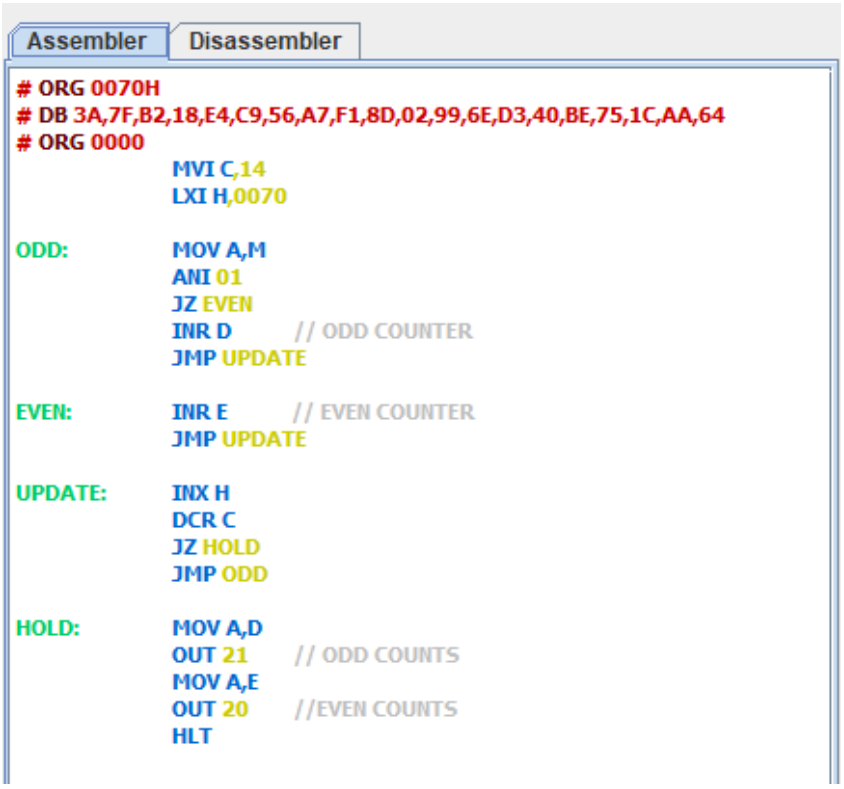
Assembler | Disassembler

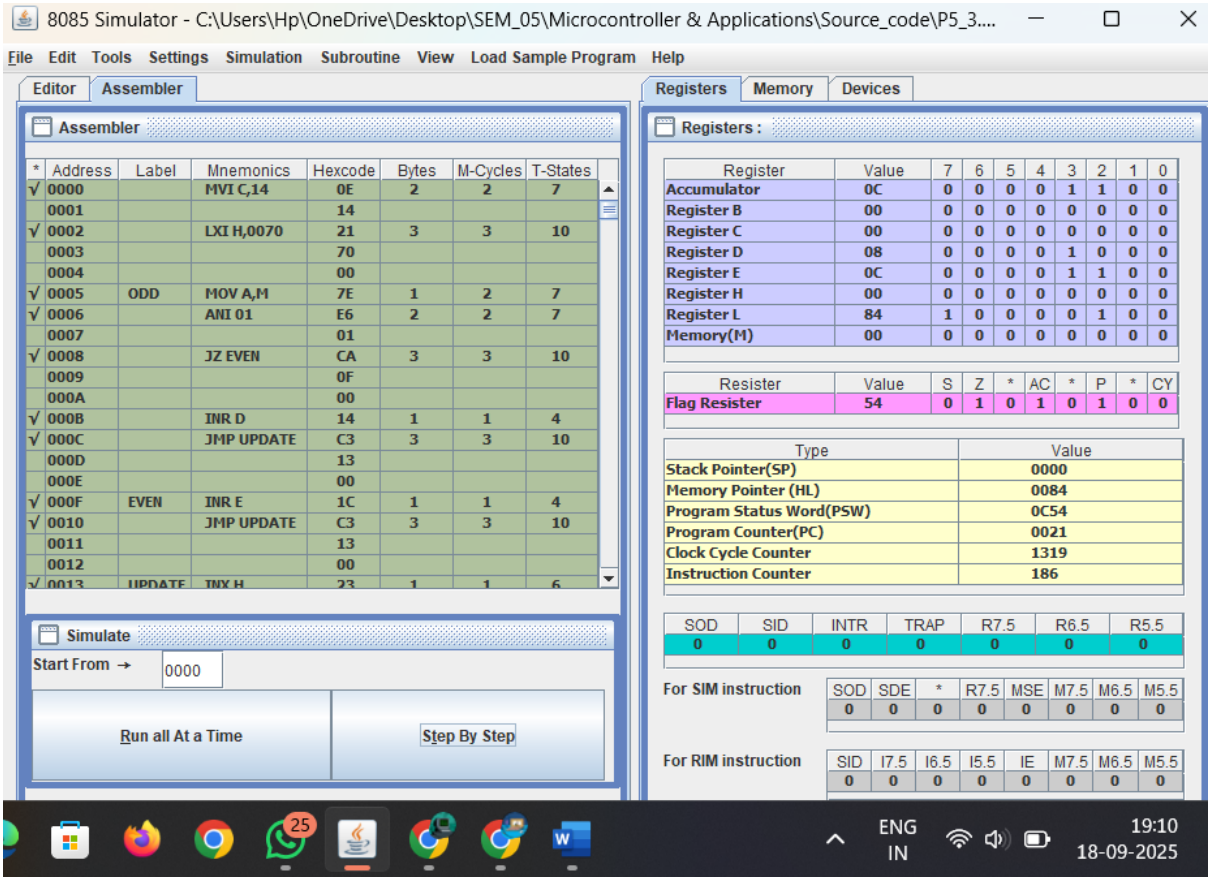
```
# ORG 1020H
# DB F5H,E4H, D1H, C6H, B8H
# ORG 0000H
        MVI C,05
        LXI H,1020
        LXI D,1025
COPY:   MOV A,M//COPY DATA FROM ACTUAL TO TEMPORARY
LOCATION
        STAX D
        INX H
        INX D
        DCR C
        JNZ COPY
        MVI B,04
LOOP:   LXI H,1025//OUTER LOOP
        MOV C,B
AGAIN:  MOV A,M//INNER LOOP
        INX H
        CMP M
        JC NO_SWAP
        MOV D,M
        MOV M,A
        DCX H
        MOV M,D
        INX H
NO_SWAP: DCR C
        JNZ AGAIN
        DCR B
        JNZ LOOP
HOLD:   HLT
```

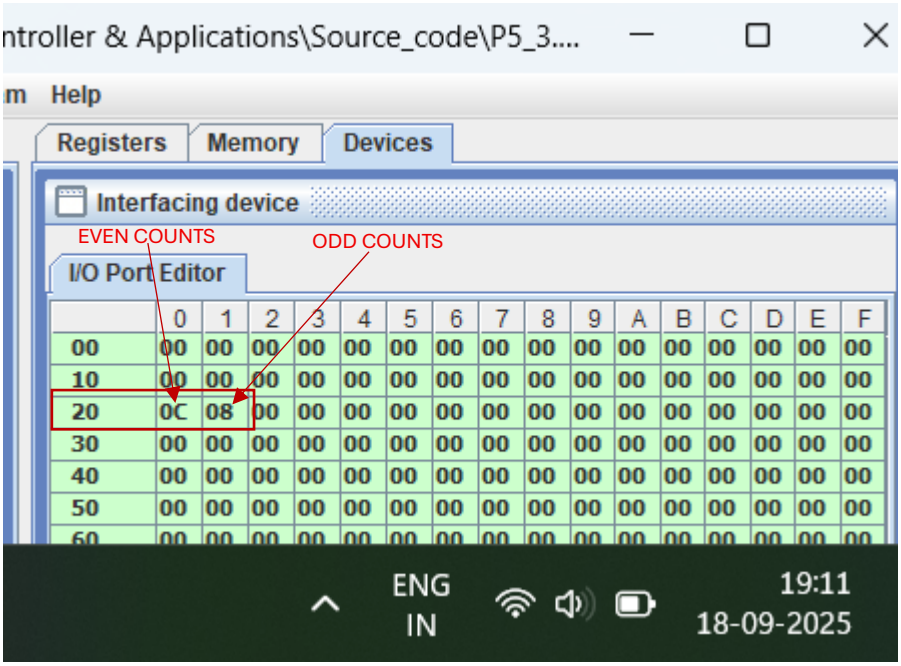




3. Assume there are 20 bytes stored in memory starting from location 0070H. Count the even and odd numbers in that bytes. Display the even count on port 20H and odd count on port 21H.




8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\Source_code\P5_3....
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Assembler
Simulate
Start From → 0000
Run all At a Time
Step By Step
Registers Memory Devices
Registers :
Register Value 7 6 5 4 3 2 1 0
Accumulator 0C 0 0 0 0 1 1 0 0
Register B 00 0 0 0 0 0 0 0 0
Register C 00 0 0 0 0 0 0 0 0
Register D 08 0 0 0 0 1 0 0 0
Register E 0C 0 0 0 0 1 1 0 0
Register H 00 0 0 0 0 0 0 0 0
Register L 84 1 0 0 0 0 1 0 0
Memory(M) 00 0 0 0 0 0 0 0 0
Resister Value S Z * AC * P * CY
Flag Resister 54 0 1 0 1 0 1 0 0
Type Value
Stack Pointer(SP) 0000
Memory Pointer (HL) 0084
Program Status Word(PSW) 0C54
Program Counter(PC) 0021
Clock Cycle Counter 1319
Instruction Counter 186
SOD SID INTR TRAP R7.5 R6.5 R5.5
For SIM instruction SOD SDE * R7.5 MSE M7.5 M6.5 M5.5
For RIM instruction SID I7.5 I6.5 I5.5 IE M7.5 M6.5 M5.5
ENG IN
19:10
18-09-2025


ntroller & Applications\Source_code\P5_3....
m Help
Registers Memory Devices
Interfacing device
EVEN COUNTS
ODD COUNTS
I/O Port Editor
0 1 2 3 4 5 6 7 8 9 A B C D E F
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 0C 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ENG IN
19:11
18-09-2025

2316217020 Rahul Prajapati Batch-52

18-September-2025

Practical 5

AIM Exercise for Jump Instruction

1. Maximum number find.

```
- MVI A,09H ; move 09 into accumulator
- LXI D,101FH ; load 101F in register Pair D
- STAX D ; store accumulator data in reg. Pair DE
- LXI H,101FH; Load 101F address in HL Pair
- MOV C,M ; move data of memory to register C
- LXI H,1020H; load 1020 address in HL Pair which is
              indicate the starting address of Data.
- MOV A,M ; Load first data in accumulator and assume
            it temporary maximum number.
- LOOP: INX H ; start loop and increment HL Pair.
-      CMP M ; Compare M with Accumulator.
-      JNC SKIP ; Jump on skip if carry flag is reset.
-      MOV B,M ; if data is greater than move it into
                 Register B its carry our real max num.
-      MOV A,M ; update temporary max number
- SKIP: DCR C ; Decrement counter.
-      JNZ LOOP ; if Counter is not Zero then Jump on loop
-      OUT 10H ; Store our max number on port 10H.
- HLT ; stop our execution.
```

2316217020 Rahul Prajapati

Date
Page

2. Sort 5 Data in ascending order. (Bubble Sort)

```
- MVI C,05 ; move imidiate data in reg. C // counter.
- LXI H,1020 ; load 1020H address in HL Pair
- LXI D,1025; loud 1025H address in DE pair.
- COPY: MOV A,M ; move memory data in Accumulator.
-       STAX D ; store Accu. data in address which is
                 Stored at DE Pair.
-       INX H ; increment HL Pair
-       INX D ; increment DE Pair
-       DCR C ; Decrement Counter.
-       JNZ COPY ; Jump on COPY if Counter is not zero
-       MVI B,04 ; Counter for outer loop
- LOOP: LXI H,1025 ; load 1025H address at HL Pair.
-       MOV C,B ; inner loop Counter.
- AGain: MOV A,M ; move memory data in Accumulator
-       INX H ; increment HL Pair
-       CMP M ; compare memory data with Accumula
-       JC NOSWAP ; Jump on NO_SWAP if carry flag set
-       MOV D,M ; move memory's data in reg. D
-       MOV M,A ; move Accumulator data in memory
-       DCX H ; Decrement HL Pair
-       MOV M,D ; move reg D's data in memory
-       INX H ; Increment HL pair.
- NO_SWAP: DCR C ; decrement Counter
-       JNZ AGAIN ; Jump on AGain in counter is not 0
-       DCR B ; Decrement outer loop Counter
-       JNZ LOOP ; if counter is not zero the Jump lo
- HLT ; Stop execution.
```

Performing Swapping

3. Count Even and odd numbers.

```
MVI C,14 ; move 14H in reg.c as counter.
LXI H,0070 ; load address 0070H in HL pair
ODD: MOV A,M ; move memory data in accumulator
     ANI 01H ; Perfor AND operation with Accumulator
               data to Hentify LSB
     JZ EVEN ; if LSB is 0 then AND operation set
               zero flag and if its 1 then Jump on EVEN
     INR D ; increment odd counter.
     JMP UPDate ; JMP on Update for Update
                  memory address and counter.
EVEN: INR E ; increment Even counter.
      JMP UPDATE; Jump on Update.
UPDATE: INX H ; increment HL pair
        DCR C ; Decrement counter
        JZ HOLD ; if counter is zero the Jump on HOLD
        JMP ODD ; Jump on ODD
HOLD: move A,D ; move odd counts in Accumulator
      OUT 21H ; Store Odd counts in Port 21H
      MOV A,E ; move Even counts in Accumulator
      OUT 20H ; store Even counts on Port 20H.
      HLT ; Stop Execution.
```