Assembly Programs

1. **Program to multiply 16-bit number.**

//Multiply data 1122H with data 02H.

```
LXI H,0000H    //Load data 0000H to HL pair
LXI D, 1122H   //Load data 1122H to DE pair
MVI C, 02H     //Move 02H to C register
X: DAD D       //Multiply contents of DE pair with contents of HL pair
   DCR C       //Decrement C register
   JNZ X       //Jump with not zero to label X
SHLD 2055H     //Store the result in memory location 2055H and 2056H
HLT            //Terminate the program.
```

//Here 16-bit data will come from memory location 2050H and 2051H.

```
LHLD 2050H     //Load data FROM 2050H and 2051H to HL pair
XCHG           //Exchange the contents of HL pair with DE pair
LXI H,0000H    //Load data 0000H to HL pair
MVI C, 02H     //Move 02H to C register
X: DAD D       //Multiply contents of DE pair with contents of HL pair
   DCR C       //Decrement C register
   JNZ X       //Jump with not zero to label X
SHLD 2055H     //Store the result in memory location 2055H
HLT            //Terminate the program.
```

2. **Program to divide 16-bit number.**

//Multiply 16-bit data from 2050H and 2051H by 02H.

```
LXI B, 0000H       //initialize BC register as 0000H.
LXI H, 0002H       //load data 0002H to HL pair.
XCHG               //exchange the content of HL pair with DE pair register.
LHLD 2050H         //load the HL pair with address 2050.
X: MOV A, L        // move the content of register L into register A.
SBB E              //subtract the contents of register E with contents of accumulator.
MOV L, A           //move the content of register A into register L.
MOV A, H           //move the content of register H into register A.
SUB D              //subtract the contents of register D with contents of accumulator
                     with carry.
MOV H, A           //move the content of register A into register H.
JC  Y              //jump to address Y if there is carry.
INX B              //increment BC register by one.
JMP X              //jump to address X.
Y:  DAD D          //add the contents of DE and HL pair.
SHLD 2056          //stores the content of HL pair into memory address 2056 and 2057.
MOV L, C           //move the content of register C into register L.
MOV H, B           //move the content of register B into register H.
SHLD 2054          //stores the content of HL pair into memory address 2054 and 2055.
HLT                // terminates the execution of program.
```

3. **Program to find a number is odd or even.**

```
LDA 2050H      //loads the content of memory location 2050 in accumulator A
ANI 01H        //performs AND operation between accumulator A and 01 and store
                the result in A
JZ X        //jump to memory location X if ZF = 1
MVI A, 00H     //assign 00 to accumulator
JMP Y          //jump to memory location Y
X: MVI A , EEH      //assign EE to accumulator
Y: STA 2055H        //stores value of A in 2055H
HLT            //stops executing the program and halts any further execution
```

4. **Find the largest number in array.**

Statement: Find the largest element in a block of data. The length of the block is in the memory location 2200H and block itself starts from memory location 2201H. Store the maximum number in memory location 2300H.

```
LDA 2200H      //Load data from memory location 2200H
MOV C,A        //Initialize counter
MVI A, 00H     //Set max=0
LXI H,2201H    //Initialize the pointer
X: CMP M       //Is number>maximum
   JNC  Y      //If not carry, jump to Y
MOV A,M        //If carry then replace maximum
Y: INX H       //Increment memory location
   DCR C       //Decrement Counter
   JNZ  X      //Jump to X, if not zero
STA 2300H      //Store maximum number in 2300H
HLT            //Terminate the program.
```

5. **Find the smallest number in array.**

Statement: Find the smallest element in a block of data. The length of the block is in the memory location 2200H and block itself starts from memory location 2201H. Store the minimum number in memory location 2300H.

```
LDA 2200H      //Load data from memory location 2200H
MOV C,A        //Initialize counter
MVI A, 00H     //Set min=0
LXI H,2201H    //Initialize the pointer
X: CMP M       //Is number>minimum
   JC  Y       //If carry, jump to Y
MOV A,M        //If not carry then replace minimum
Y: INX H       //Increment memory location
   DCR C       //Decrement Counter
   JNZ  X      //Jump to X, if not zero
STA 2300H      //Store maximum number in 2300H
HLT            //Terminate the program.
```