Date: 20-01-2022 Exp.2 Operations on array

Aim: To find the sum and average of the array elements by combining elements of two arrays.

Tool Used:

Assembler - MASM 611

Algorithm:

- 5 elements are stored in data segment array.
- Array 2 is declared and each of the element of the ARRAY1 is firstly multiplied by 02h, followed by 10h individually and stored in it.
  First 5 elements are obtained by multiplying 02h and next 5 elements are obtained by multiplying 10h
- Initialize a counter variable and use the ADD command to add each of the
- element of the ARRAY2. The result of the addition is then stored in variable
- called SUMRES.
- Then the value stored in AX register is taken (sum of elements) and divided
- by the number of elements, i.e. 10. This result is then stored in the variable
- called AVGRES.

Program:

```
ASSUME CS:CODE,DS:DATA

DATA SEGMENT

ARRAY1 DB 97H,95H,98H,97H,98H

ARRAY2 DW 0AH DUP(0)

SUMRES DW ?

AVGRES DW ?

COUNT1 EQU 05H
```

```
COUNT2 EQU 0AH
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        XOR AX,AX
        MOV BL,02H
        MOV BH,10H
        LEA SI,ARRAY1
        LEA DI,ARRAY2
        MOV CX,COUNT1
RPT1 :
        MOV AL,[SI]
        MUL BL
        MOV [DI],AX
        INC SI
        ADD DI,02H
        XOR AX,AX
        LOOP RPT1
        LEA SI,ARRAY1
        MOV CX,COUNT1
RPT2 :
```

```
        MOV AL,[SI]
        MUL BH
        MOV [DI],AX
        INC SI
        ADD DI,02H
        XOR AX,AX
        LOOP RPT2
        MOV CX,COUNT2
        LEA SI,ARRAY2
RPT3:
        ADD AX,[SI]
        ADD SI,02H
        LOOP RPT3
        MOV SUMRES,AX
        MOV BX,COUNT2
        DIV BX
        MOV AVGRES,AX
HLT
CODE ENDS
END START
```

Sample Input:

ARRAY1 = [97H, 95H, 98H, 97H, 98H]
ARRAY2 = [12E, 12A, 130, 12E, 130, 970, 950, 980, 970, 980]

Sample Output:

SUM = 3516H
AVERAGE = 54F

Manual Verification:

Hex value:
012E + 012A = **258**

Hex value:
0258 + 0130 = **388**

Hex value:
0388 + 012E = **4B6**

Hex value:
04B6 + 0130 = **5E6**

Hex value:
05E6 + 0970 = **F56**

Hex value:
0F56 + 0950 = **18A6**

Hex value:
18A6 + 0980 = **2226**

Hex value:
2226 + 0970 = **2B96**

Hex value:
2B96 + 0980 = **3516**

Hex value:
3516 ÷ A = **54F**

Register/ Memory Contents for I/O:

```
-u
0766:000B 8D360000      LEA     SI,[0000]
0766:000F 8D3E0500      LEA     DI,[0005]
0766:0013 B90500        MOV     CX,0005
0766:0016 8A04          MOV     AL,[SI]
0766:0018 F6E3          MUL     BL
0766:001A 8905          MOV     [DI],AX
0766:001C 46            INC     SI
0766:001D 83C702        ADD     DI,+02
0766:0020 33C0          XOR     AX,AX
0766:0022 E2F2          LOOP    0016
0766:0024 8D360000      LEA     SI,[0000]
0766:0028 B90500        MOV     CX,0005
```

Snapshot of the Output:

```
C:\BIN>debug arrgen.exe
-t

AX=0764  BX=0000  CX=0073  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0754  ES=0754  SS=0763  CS=0766  IP=0003   NV UP EI PL NZ NA PO NC
0766:0003 8ED8          MOV     DS,AX
-t

AX=0764  BX=0000  CX=0073  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0764  ES=0754  SS=0763  CS=0766  IP=0005   NV UP EI PL NZ NA PO NC
0766:0005 33C0          XOR     AX,AX
-t

AX=0000  BX=0000  CX=0073  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0764  ES=0754  SS=0763  CS=0766  IP=0007   NV UP EI PL ZR NA PE NC
0766:0007 B302          MOV     BL,02
-t

AX=0000  BX=0002  CX=0073  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0764  ES=0754  SS=0763  CS=0766  IP=0009   NV UP EI PL ZR NA PE NC
0766:0009 B710          MOV     BH,10
```

```
-t

AX=0000  BX=1002  CX=0073  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0764  ES=0754  SS=0763  CS=0766  IP=000B   NV UP EI PL ZR NA PE NC
0766:000B 8D360000      LEA     SI,[0000]                       DS:0000=9597
```

```
-t

AX=054F  BX=000A  CX=0000  DX=0000  SP=0000  BP=0000  SI=0019  DI=0019
DS=0764  ES=0754  SS=0763  CS=0766  IP=0052   NV UP EI PL NZ NA PO NC
0766:0052 F4            HLT
-d 0764:0000 001c
0764:0000  97 95 98 97 98 2E 01 2A-01 30 01 2E 01 30 01 70   .......*.0...0.p
0764:0010  09 50 09 80 09 70 09 80-09 16 35 4F 05            .P...p....5O.
-
```

Result:

Hence we have verified the results and successfully executed the 10 16 bit numbers addition and average operations.