Microprocessor Fundamentals and Programming

Assignment 3　　　CE092: Nevil Parmar

Task1:

Write a program to find the smallest number in an array of words and store it in the variable named small of the data segment.

```
DATA SEGMENT
      A DB 5,2,5,6,4,3
      SMALLEST DB ?
DATA ENDS

CODE SEGMENT
      ASSUME DS:DATA,CS:CODE
START:
      MOV AX,DATA
      MOV DS,AX
      MOV CX,0000
      MOV CL,06
      LEA BX,A
      MOV AX,0000
      MOV AL,0FFH
      MOV AH,BYTE PTR[BX]
CHECK:
      CMP AL,BYTE PTR[BX]
      JNC SAVE
      JMP NEXT
SAVE:
      MOV AL,BYTE PTR[BX]
      JMP NEXT
NEXT:
      INC BX
      DEC CL
      CMP CL,00
      JNZ CHECK
      MOV SMALLEST,AL
      INT 03
```

```
CODE ENDS
END START
```

Output:

Task2:

Write a program to sort an array of signed numbers given in the data segment.

```
DATA SEGMENT
    ARRAY DB 99H,12H,56H,45H,36H
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    MOV DS,AX
    MOV CH,04H

UP2:
    MOV CL,04H
    LEA SI,ARRAY

UP1:
    MOV AL,[SI]
    MOV BL,[SI+1]
    CMP AL,BL
    JC DOWN
    MOV DL,[SI+1]
    XCHG [SI],DL
```

```
    MOV [SI+1],DL
```

```
DOWN:
    INC SI
    DEC CL
    JNZ UP1
    DEC CH
    JNZ UP2
    INT 3
CODE ENDS
END START
```

Output:


```
AX=0756 BX=0099 CX=0000 DX=0045 SP=0000 BP=0000 SI=0004 DI=0000
DS=0744 ES=0734 SS=0743 CS=0745 IP=0027 NV UP EI PL ZR NA PE CY
0745:0027 00F8              ADD     AL,BH
-D 0000
0744:0000  12 36 45 56 99 00 00 00-00 00 00 00 00 00 00 00  .6EV............
0744:0010  B8 44 07 8E D8 B5 04 B1-04 BE 00 00 8A 04 8A 5C  .D.............\
0744:0020  01 3A C3 72 08 8A 54 01-86 14 88 54 01 46 FE C9  .:.r..T....T.F..
0744:0030  75 EA FE CD 75 E1 CC 00-F8 26 80 0E 01 00 04 26  u...u....&.....&
0744:0040  A1 16 00 99 26 A3 0A 00-26 89 16 0C 00 26 88 16  ....&...&....&..
0744:0050  24 00 33 C0 26 A3 14 00-26 A3 16 00 26 80 26 05  $.3.&...&...&.&.
0744:0060  00 FC 26 80 0E 05 00 01-E8 5E 3B E8 43 3B E8 82  ..&......^;.C;..
0744:0070  3B E9 DD 04 BF 3C 01 57-8B D8 E8 87 EE 26 C7 06  ;....<.W.....&..
```

Task3:

Write a program to find the square root of a given number.

```
DATA SEGMENT
    NUMBER DW 19H
    ANS DW ?
DATA ENDS

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AX,NUMBER
    MOV CX,0000H
```

```
        MOV BX,0FFFFH
REPEAT:
        ADD BX,02H
        INC CX
        SUB AX,BX
        JNZ REPEAT
        MOV ANS,CX
        INT 03H
CODE ENDS
END START
```

Output:


```
AX=0000 BX=0009 CX=0005 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0744 ES=0734 SS=0743 CS=0745 IP=001B NV UP EI PL ZR NA PE NC
0745:001B 26F606040004      TEST    BYTE PTR ES:[0004],04        ES:0004=00
-d 0000
0744:0000  19 00 05 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0744:0010  B8 44 07 8E D8 A1 00 00-B9 00 00 BB FF FF 83 C3  .D..............
0744:0020  02 41 2B C3 75 F8 89 0E-02 00 CC 26 F6 06 04 00  .A+.u......&....
0744:0030  04 74 0C 26 80 26 01 00-F8 26 80 0E 01 00 04 26  .t.&.&...&.....&
0744:0040  A1 16 00 99 26 A3 0A 00-26 89 16 0C 00 26 88 16  ....&...&....&..
0744:0050  24 00 33 C0 26 A3 14 00-26 A3 16 00 26 80 26 05  $.3.&...&...&.&.
0744:0060  00 FC 26 80 0E 05 00 01-E8 5E 3B E8 43 3B E8 82  ..&......^;.C;..
0744:0070  3B E9 DD 04 BF 3C 01 57-8B D8 E8 87 EE 26 C7 06  ;....<.W.....&..
```

Task4:

Consider an array(array1) of 20 random numbers ranging between 1 to 4. Write a program to Count the number of 1's, 2's, 3's and 4's in the array1 and store the result of the count in one more array(array 2) of size 4 elements. At first location of array2 it has to store the count of 1's, at second location it has to store count of 2's and so on...

```
DATA SEGMENT
        ARRAY2 DB 4 DUP(0)
        ARRAY1 DB 01h,01h,02h,04,03h,02h,01h,00h,04h,02h,03h
,00h,01h,02h,03h,04h,01h,03h,00h,02h
DATA ENDS

CODE SEGMENT
        ASSUME DS:DATA,CS:CODE
START:
        MOV AX,DATA
        MOV DS,AX
        LEA SI,ARRAY1
```

```
        LEA DI,ARRAY2
        MOV CX,14H
        MOV BX,0000H
COMPUTE:
        MOV BL,[SI]
        INC [DI+BX-1]
        INC SI
        DEC CX
        JCXZ STOP
        JMP COMPUTE
    STOP:
        INT 03H
CODE ENDS
END START
```

Output: