Experiment Name:

Write a procedure BUBBLE to sort a byte array by the bubble sort algorithm. The procedure receives the offset address of the array in the SI and the number of elements in the BX. Write a program that lets the user type a list of single digit numbers, with one blank between numbers, calls BUBBLE to sort them and prints the sorted list on the next line.

Theory:

The objective of this program is to sorting the a list of single digit numbers using bubble sort algorithm where BUBBLE named procedure will be used. Stack and Array are needed to perform the I/O operation. For this program in assembly, While loop, CMP, Stack, XCHG, BUBBLE procedure were used as well as the required instructions and some registers to execute the solution.

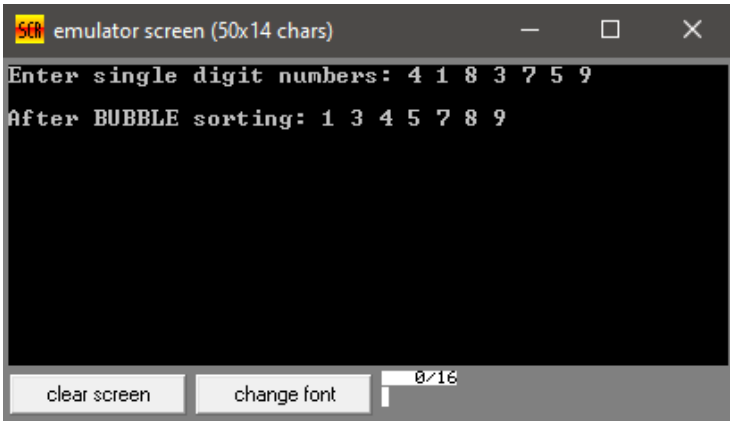
Code:

```
.MODEL SMALL
.STACK 100H

.DATA
   P1  DB  'Enter the time in seconds (0 to 65535) = $'
   P2  DB  0DH,0AH,'Time in hh:mm:ss format= $'
   COLON DB  ' : $'

.CODE
.MODEL SMALL
 .STACK 100H

 .DATA
   PROMPT_1  DB  'Enter single digit numbers: $'
   PROMPT_2  DB  0DH,0AH,'After BUBBLE sorting : $'

   ARRAY DB 100 DUB (0)

 .CODE
  MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    LEA DX, PROMPT_1
    MOV AH, 9
    INT 21H

    XOR CX,CX
    MOV AH,1
    INT 21H
    XOR SI,SI
```

```
  C:
    CMP AL,0DH
    JE END_WHILE

    MOV ARRAY[SI],AL
    INC SI
    INC CX

    MOV AH,2
    MOV DL,' '
    INT 21H

    MOV AH,1
    INT 21H
    JMP C

  END_WHILE:
    MOV AH,2
    MOV DL,0DH
    INT 21H
    MOV DL,0AH
    INT 21H

  JCXZ EXIT
    LEA SI,ARRAY
    MOV BX,CX
  CALL BUBBLE
  MOV AH,9
  LEA DX,PROMPT_2
  INT 21H
  XOR SI,SI
  TOP:
    MOV AH,2
    MOV DL,ARRAY[SI]
    INT 21H
    MOV DL,' '
    INT 21H
    INC SI
    LOOP TOP
  EXIT:
    MOV AH,4CH
    INT 21H
    MAIN ENDP


BUBBLE PROC

 PUSH AX
 PUSH BX
 PUSH CX
```

```
  PUSH DX
  PUSH DI

  MOV AX, SI
  MOV CX, BX
  DEC CX

  @OUTER_LOOP:
    MOV BX, CX

    MOV SI, AX
    MOV DI, AX
    INC DI

    @INNER_LOOP:
      MOV DL, [SI]

      CMP DL, [DI]
      JNG @SKIP_EXCHANGE

      XCHG DL, [DI]
      MOV [SI], DL

      @SKIP_EXCHANGE:
      INC SI
      INC DI

      DEC BX
    JNZ @INNER_LOOP
  LOOP @OUTER_LOOP

  POP DI
  POP DX
  POP CX
  POP BX
  POP AX

  RET
BUBBLE ENDP
END MAIN
```

Output:



Discussion:

In the above program, a list of single digit numbers was taken as input from the user and then in C loop block an ARRAY[] was used to take the list of numbers from the user until the user hits the ENTER and SI was used for indexing. Then the procedure BUBBLE was called. In the BUBBLE procedure, AX,BX,CX,DX,DI these registers were pushed into the stack, Then SI and BX were moved into AX,CX respectively and CX was decremented by 1. Then after moving CX,AX into BX,SI respectively in a @OUTER_LOOP , AX was again moved into DI and DI was incremented by 1. Then in a @INNER_LOOP, it was compared in between two consecutive numbers whether the left one is greater than the right one, if so then XCHG was used to swap them otherwise not swapped. At the end of BUBBLE procedure, DI,DX,DX,CX,BX,AX was popped from the stack and RET was used to return the stack into the main program. Then using DL proper output of bubble sorted array was printed. Thus the program was successfully executed.