**EXPERIMENT 5**

**MP LAB**

**NAME:AUGUSTAS NIXON**

**ROLL NO:19**

**(a)Aim: To find the smallest number of an array of unsigned and unordered array of**

1. **bit numbers.**

**ALGORITHM:**

Step 1: Start

Step 2: Set up a logical data segment by defining identifiers to represent the data stored in an array, count for length of the array and the result.

Step 3: Initialize the logical code segment. Initialize DS register with the

the starting address of the logical data segment.

Step 4: Load starting address of the array to SI.

Step 5: Move 0000H to AX.

Step 6: Move the content of (COUNT - 1) to CX.

Step 7: Move the contents of SI to AL.

Step 8: Compare AL and contents of (SI+1). If AL is smaller, go to step 10. Else move to step 9.

Step 9: Move the contents of (SI+1) to AL.

Step 10: Increment SI to obtain the address of the next element in the array.

Step 11: Decrement CX.

Step 12: If CX is not equal to zero, repeat steps 7-12, else go to step 13.

Step 13: Move the content of AL to a memory location (SMALL).

Step 14: Move 4CH to AH register to terminate execution and call the DOS function is called INT 21H.

Step 15: Stop

**PROGRAM:**

DATA SEGMENT

ARRAY DB 0AH, 02H, 0B2H

COUNT EQU ($-ARRAY)

SMALL DB 1 DUP(?)

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START: MOV AX, DATA

MOV DS, AX

LEA SI, ARRAY

MOV AX, 0000H

MOV CX, (COUNT-1)

MOV AL, [SI]

L1: CMP AL, [SI+1]

JBE L2

MOV AL, [SI+1]

L2: INC SI

DEC CX

JNZ L1

MOV SMALL, AL

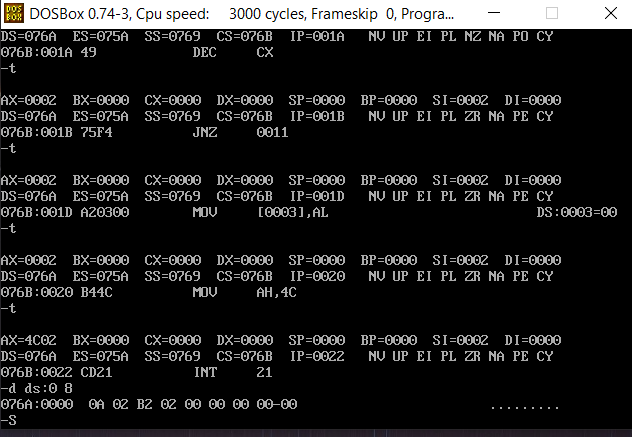
MOV AH, 4CH

INT 21H

CODE ENDS

END START

**OUTPUT:**

****

**Exp 5(b) - Largest number of an array of unsigned numbers**

**ALGORITHM:**

Step 1: Start

Step 2: Set up a logical data segment by defining identifiers to represent the data stored in an array, count for the length of the array and the result.

Step 3: Initialize the logical code segment. Initialize DS register with the

the starting address of the logical data segment.

Step 4: Load starting address of the array to SI.

Step 5: Move 0000H to AX.

Step 6: Move the content of (COUNT - 1) to CX.

Step 7: Move the contents of SI to AL.

Step 8: Compare AL and contents of (SI+1). If AL is greater, go to step10. Else move to step 9.

Step 9: Move the contents of (SI+1) to AL.

Step 10: Increment SI to obtain the address of the next element in the array.

Step 11: Decrement CX.

Step 12: If CX is not equal to zero, repeat steps 7-12, else go to step 13.

Step 13: Move the content of AL to a memory location (LARGE).

Step 14: Move 4CH to AH register to terminate execution and call the DOS function is called INT 21H.

Step 15: Stop

**PROGRAM:**

DATA SEGMENT

ARRAY DB 0AH, 00H, 0F5H

COUNT EQU ($-ARRAY)

SMALL DB 1 DUP(?)

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START: MOV AX, DATA

MOV DS, AX

LEA SI, ARRAY

MOV AX, 0000H

MOV CX, (COUNT-1)

MOV AL, [SI]

L1: CMP AL, [SI+1]

JAE L2

MOV AL, [SI+1]

L2: INC SI

DEC CX

JNZ L1

MOV SMALL, AL

MOV AH, 4CH

INT 21H

CODE ENDS

END START

**OUTPUT:**

