Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai-400008

DEPARTMENT OF INFORMATION TECHNOLOGY

Subject in charge: ER. SHRINIDHI GINDI
Course: Microprocessor Lab

Course code: ITL403

SR NO.	EXPERIMENT NAME	DOP	DOC	GRADE	SIGN
1.	Study of different types of PC motherboard technology				
2.	To study & verify the truth table of logic gates				
3.	To realize half adder & full adder				
4.	To design & setup 4:1 MUX, 1:4 DEMUX				
5.	Write an ALP to add two 16-bit numbers				
6.	Write an ALP to convert two digit packed BCD to unpacked BCD.				
7.	Write an ALP to interchange a block of data				
8.	Write an ALP to count number of 1's and 0's in a given 8-bit number				
9.	Write an ALP to find whether given number is odd or even				
10.	Write an ALP to check whether a given string is a palindrome or not				
11.	Write an ALP to find Factorial of a given number				
12.	Study a program for seven segment LED display through 8255 (PCI BASED)				

Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai-400008

DEPARTMENT OF INFORMATION TECHNOLOGY

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: 5

Aim: Write an ALP to add TWO 16-bit numbers

- **AIM:** Write an ALP To Add two 16-bit numbers

- **SOFTWARE REQUIRED**: Masm 16 Bit

- **ALGORITHM:**

  1. Initialize the MSBs of sum to 0
  2. Get the first number.
  3. Add the second number to the first number.
  4. If there is any carry, increment MSBs of sum by 1.
  5. Store LSBs of sum.
  6. Store MSBs of sum.

- **PROGRAM**:

```
.model small
.stack 100h
.data
a dw 8111h
b dw 6222h
.code
main proc
mov ax, @data
mov ds, ax
mov ax, a
mov bx, b
adc ax, bx
mov ah, 4ch
int 21h
main endp
end main
```

- **<u>OUTPUT:</u>**

- **<u>RESULT</u>**: Program is executed without errors and the output is verified.

- **<u>VERIFICATION AND VALIDATION</u>**: Output is verified and is found correct.

- **<u>CONCLUSION</u>**: The addition of two 16-bit data is done and the output is verified.

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

Name of Student:

Roll Number:

Experiment Number: <u>06</u>

AIM: WRITE AN ALP TO CONVERT TWO DIGIT PACKED BCD TO UNPACKED BCD

<h1 style="text-align:center">EXPERIMENT NO.:06</h1>

AIM: WRITE AN ALP TO CONVERT TWO DIGIT PACKED BCD TO UNPACKED BCD

SOFTWARE REQUIRED: MASM 16 BIT

ALGORITHM:

STEP 1: INITIALIZE THE DATA MEMORY.

STEP 2: LOAD NUMBER INTO REGISTER AL.

STEP 3: MASK THE LOWER NIBBLE.

STEP 4: ROTATE 4 TIMES LEFT TO MAKE MSB DIGIT = LSB.

STEP 5: DISPLAY THE DIGIT.

STEP 6: LOAD NUMBER IN AL.
STEP 7: MASK UPPER NIBBLE.

STEP 8: DISPLAY THE RESULT.

STEP 9: STOP.

PROGRAM:

```
.model small
.stack 100h
.data
a db 52h
b db 02 dup(?)
.code
main proc
mov ax, @data
mov ds, ax
lea si, a
mov al, [si]
mov bl, al
and bl, 0fh
mov cl, 04h
ror al, cl
and al, 0fh
mov [si+1], al
```

```
mov [si+2], bl
int 3
main endp
end main
```

OUTPUT:

RESULT:

PROGRAM IS EXECUTED WITHOUT ERRORS AND THE

OUTPUT IS VERIFIED.

VERIFICATION AND VALIDATION:

OUTPUT IS VERIFIED AND IS FOUND CORRECT.

CONCLUSION:

THUS CONVERT TWO DIGIT PACKED BCD TO UNPACKED
BCD IN TASM.

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai-400008

DEPARTMENT OF INFORMATION TECHNOLOGY

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: 07

Aim: WRITE AN ALP TO INTERCHANGE A BLOCK OF DATA

EXPERIMENT NO.:7

AIM: WRITE AN ALP TO INTERCHANGE A BLOCK OF DATA

SOFTWARE REQUIRED: MASM 16 BIT

ALGORITHM:

1. DEFINE TWO SETS OF DATA.

2. LOAD ADDRESS OF SRC TO SI

3. LOAD ADDRESS OF DST TO DI

4. INITIALIZE COUNTER

5. INTERCHANGE DATA IN SRC AND DST

6. REPEAT STEP 5 UNTIL COUNTER = 0.

7. END

PROGRAM:

```
.model small
.stack 100h
.data
blk1 db 01,02,03,04,05,06,07,08,09,0Ah
blk2 db 0ah dup(?)
count dw 0ah
.code
main proc
mov ax, @data
mov ds, ax
mov es, ax
mov si, offset blk1;
mov di, offset blk2;
mov cx, count
cld
rep movsb
main endp
end main
```

OUTPUT:

RESULT:

PROGRAM IS EXECUTED WITHOUT ERRORS AND THE OUTPUT IS

VERIFIED.

VERIFICATION AND VALIDATION:

OUTPUT IS VERIFIED AND IS FOUND CORRECT.

CONCLUSION:

THE BLOCKS OF DATA DEFINED IN THE PROGRAM IS
INTERCHANGED AND OUTPUT IS VERIFIED

Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai-400008

DEPARTMENT OF INFORMATION TECHNOLOGY

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: 8

Aim: WRITE AN ALP TO COUNT NUMBER OF 1'S AND 0'S IN A GIVEN 8-BIT NUMBER

EXPERIMENT NO.8

- **AIM:** WRITE AN ALP TO COUNT NUMBER OF 1'S AND 0'S IN A GIVEN 8-BIT NUMBER

- **SOFTWARE REQUIRED**: Masm 16 Bit

  ALGORITHM:

  1. DEFINE DATA VARIABLES
  2. LOAD ADDRESS OF DATA SEGMENT TO DS
  3. LOAD VALUE OF NUM TO AL
  4. INITIALIZE COUNTER
  5. ROTATE AL REGISTER AND CHECK CARRY FLAG
  6. COUNT 0 AND 1 IN CY FLAG FOR EACH ROTATION
  7. REPEAT STEP 5 AND 6 UNTIL COUNTER = 0
  8. END

- **PROGRAM**:

```
.model small
.stack 100h
.data
num db 13h
ones db 0
zeros db 0
.code
main proc
mov ax, @data
mov ds, ax
mov al, num
mov cx, 08h
back: ror al, 1
jnc zerinc
inc ones
jmp next
zerinc: inc zeros
next: dec cx
jnz back
mov ah, 4ch
int 21h
```

```
main endp
end main
```

- **OUTPUT:**

- **RESULT**: Program is executed without errors and the output is verified.

- **VERIFICATION AND VALIDATION**: Output is verified and is found correct.

- **CONCLUSION**: The number of ones and zeros in a given data are found and the output is verified.

Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla,Mumbai-400008

<u>DEPARTMENT OF INFORMATION TECHNOLOGY</u>

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: <u>9</u>

Aim: <u>WRITE AN ALP TO FIND WHETHER GIVEN NUMBER IS ODD OR EVEN</u>

EXPERIMENT NO.:9

AIM: WRITE AN ALP TO FIND WHETHER GIVEN

NUMBER IS ODD OR EVEN

SOFTWARE REQUIRED: MASM 16 BIT

ALGORITHM:

1. DEFINE DATA VARIABLES
2. LOAD ADDRESS OF DATA SEGMENT TO DS
3. LOAD VALUE OF DATA TO AL
4. SHIFT AL RIGHT
5. CHECK IF CARRY FLAG IS SET
6. IF CY = 0, NUMBER IS EVEN OTHERWISE ODD
7. DISPLAY MESSAGE
8. END

PROGRAM:

```
.model small
.stack 100h
.data
a db 51h
msg1 db "number is even$"
msg2 db "given number odd$"
.code
mov ax, @data
mov ds, ax
mov al, a
shr al, 01h
jc L
mov ah, 09h
lea dx, msg1
int 21h
jmp stop
L: mov ah, 09h
```

```
lea dx, msg2
int 21h
stop: mov ah, 4ch
int 21h
end
```

OUTPUT:

RESULT: PROGRAM IS EXECUTED WITHOUT ERRORS AND

THE OUTPUT IS VERIFIED.

VERIFICATION AND VALIDATION: OUTPUT IS VERIFIED

AND IS FOUND CORRECT.

CONCLUSION: THE ODD AND EVEN NUMBERS ARE

SEPERATED AND OUTPUT IS VERIFIED.

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: 10.

Aim: WRITE AN ALP TO CHECK WHETHER A

GIVEN STRING IS A PALINDROME OR NOT

EXPERIMENT NO: 10

AIM: WRITE AN ALP TO CHECK WHETHER A GIVEN STRING

IS A PALINDROME OR NOT

SOFTWARE REQUIRED: MASM 16 BIT.

ALGORITHM:

1. DEFINE DATA VARIABLES
2. LOAD ADDRESS OF DATA SEGMENT TO DS
3. LOAD ADDRESS OF STRING
4. REVERSE STRING
5. COMPARE ORIGINAL STRING AND REVERSED STRING
6. IF BOTH STRINGS ARE EQUAL, IT IS PALINDROME
OTHERWISE NOT PALINDROME
7. DISPLAY MESSAGE
8. END

PROGRAM:

```
.model small
.stack 100h
.data
str1 db "madam"
L dw $-str1
rstr db 10 dup(?)
msg1 db "its palindrome$"
```

```
msg2 db "its not palindrome$"
.code
mov ax, @data
mov ds, ax
mov es, ax
mov cx, L
lea si, str1
lea di, rstr
add di, cx
back: mov al, [si]
mov [di], al
inc si
dec di
loop back
lea si, str1
lea di, rstr
mov cx, 1
cld
cmpsb
jne notpal
lea dx, msg1
jmp disp
notpal: lea dx, msg2
disp: mov ah, 9
int 21h
mov ah, 4ch
int 21h
end
```

OUTPUT:

RESULT: PROGRAM EXECUTED WITHOUT ERRORS & THE

OUTPUT IS VERIFIED.

<u>VERIFICATION & VALIDATION:</u>OUTPUT IS VERIFIED &

FOUND CORRECT.

<u>CONCLUSION:</u>THE STRING IS FOUND TO BE A

PALINDROME & THE OUTPUT IS VERIFIED.

Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai-400008

DEPARTMENT OF INFORMATION TECHNOLOGY

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: 11.

Aim: WRITE AN ALP TO FIND FACTORIAL OF A GIVEN NUMBER

AIM: WRITE AN ALP TO FIND FACTORIAL OF A GIVEN NUMBER

SOFTWARE REQUIRED: MASM 16 BIT.

ALGORITHM:

1. DEFINE DATA VARIABLES
2. LOAD DATA INTO REGISTERS
3. CALL FACTORIAL PROCEDURE
4. IF NUMBER = 1, RETURN 1 OTHERWISE RETURN 'NUMBER * FACT(NUMBER-1)'
5. STORE ANSWER IN 'RESULT' VARIABLE
6. END

PROGRAM:

```
.model small
.stack 100h
.data
num dw 5
result dw (?)
.code
main proc
mov ax, @data
mov ds, ax
mov ax, 01
mov cx, num
cmp cx, 00
je loop1
mov bx, cx
call fact
loop1: mov result, ax
mov ah, 4ch
int 21h
main endp

fact proc
```

```
cmp bx, 01
jz loop2
push bx
dec bx
call fact
pop bx
mul bx
ret
loop2: mov ax, 01
ret
fact endp
end
```

OUTPUT:

RESULT: PROGRAM IS EXECUTED WITHOUT ERRORS AND

THE OUTPUT IS VERIFIED.

VERIFICATION AND VALIDATION: OUTPUT IS VERIFIED

AND IS FOUND CORRECT.

CONCLUSION:THE FACTORIAL OF A GIVEN NUMBER IS
FOUND AND OUTPUT IS VERIFIED

Anjuman-i-Islam's

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, SabooSiddik Polytechnic Road, Byculla,Mumbai-400008

<u>DEPARTMENT OF INFORMATION TECHNOLOGY</u>

Course: Microprocessor Lab

Course code: ITL403

Name of Student:

Roll Number:

Experiment Number: <u>12</u>

Aim: <u>TO STUDY A PROGRAM FOR SEVEN SEGMENT LED DISPLAY THROUGH 8255 (PCI BASED)</u>

EXPERIMENT NO.:12

AIM: TO STUDY A PROGRAM FOR SEVEN SEGMENT LED

DISPLAY THROUGH 8255 (PCI BASED)

SOFTWARE REQUIRED: MASM 16 BIT

PROGRAM:

```
.model small
.stack 100h
.data
porta equ 0d400h
portc equ 0d402h
cr equ 0d403h
fire db 79h,77h,06h,71h,00,00
help db 00,00,73h,38h,79h,76h
.code
mov ax, @data
mov ds, ax
mov al, 80h
mov dx, cr
out dx, al
mov cx, 02h
again: mov di, 50
disp1: lea si, fire
call display
dec di
jnz disp1
mov di, 50
disp2: lea si, help
call display
dec di
jnz disp2
loop again
mov ah, 4ch
int 21h

display proc
mov ah, 0
back: mov al, ah
```

```
mov dx, portc
out dx, al
lodsb
mov dx, porta
out dx, al
call delay
inc ah
cmp ah, 6
jnz back
ret
display endp

delay proc
push bx
push cx
mov bx, 0ffh
loop2:
loop1: mov cx, 0fffh
dec bx
jnz loop2
pop cx
pop bx
ret
delay endp
end
```

RESULT: PROGRAM IS EXECUTED WITHOUT ERRORS AND THE OUTPUT IS VERIFIED.

VERIFICATION AND VALIDATION: OUTPUT IS VERIFIED AND IS FOUND CORRECT.

CONCLUSION: THE 7 SEGMENT DISPLAY IS PROGRAMMED SUCCESSFULLY AND OUTPUT IS VERIFIED.