**Experiment 1**

1. Write a program to print the alphabets (A-Z)

**Code:**

.code

start:

mov cl,26D

mov dl,’A’

L:

mov ah,02h

int 21h #(INPUT OUTPUT ON THE SCREEN)

inc dl #dl is 8 bit register

loop L

end start

2. Write a program to print the alphabets (a-z)

**Code:**

.code

start:

mov cl,26D

mov dl,’a’

L:

mov ah,02h

int 21h

inc dl

loop L

end start

3. Write a program to print the numbers from 0-9

**Code:**

.code

start:

mov cl,10D

mov dl,’0’

L:

mov ah,02h

int 21h

inc dl

loop L

end start

**Experiment 2**

1. Write a program to print ASCII table

**Code:**

.code

start:

mov cx,256D

mov dl,0

L:

mov ah,02h

int 21h

inc dl

loop L

end start

2. Write a program to print AaBbCc………Zz

**Code:**

.code

start:

mov cl,26D

mov bl,’A’

mov bh,’a’

L:

mov dl,bl

mov ah,02h

int 21h

inc bl

mov dl,bh

mov ah,02h

int 21h

inc bh

loop L

end start

3. Write a program to print AaaBbbCcc………Zzz

**Code:**

.code

start:

mov cl,26D

mov bl,’A’

mov bh,’a’

L:

mov dl,bl

mov ah,02h

int 21h

inc bl

mov dl,bh

mov ah,02h

int 21h

mov ah,02h

int 21h

inc bh

loop L

end start

4. Write a program to print AbCdEf……….z

**Code:**

.code

start:

mov cl,13D

mov bl,’A’

mov bh,’b’

L:

mov dl,bl

mov ah,02h

int 21h

inc bl

inc bl

mov dl,bh

mov ah,02h

int 21h

inc bh

inc bh

loop L

end start

**Experiment 3**

1. Write a program to print the string using 09h function

**Code:**

.data

a db "Microprocessor$"

.code

start:

mov ax,@data

mov ds,ax

mov dl,offset a

mov ah,09h

int 21h

end start

2. Write a program to print the string character wise

**Code:**

.data

a db "Microprocessor LAB$"

.code

start:

mov ax,@data

mov ds,ax

mov si,offset a

L:

mov dl,[si]

mov ah,02h

int 21h

inc si

cmp [si],'$'

jnz L

end start

3. Repeat the program 1 for 16-bit string

**Code:**

.data

a dw "Microprocessor$"

.code

start:

mov ax,@data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

end start

4. Repeat the program 2 for 16-bit string

**Code:**

.data

a dw "Microprocessor Lab$"

.code

start:

mov ax,@data

mov ds,ax

mov si,offset a

L:

mov dx,[si]

mov ah,02h

int 21h

inc si

cmp [si],"$"

jne L

end start

**Experiment 4**

1. Write a program to convert the given string into reverse form.

**Code:**

.data

a db "Microprocessor$"

.code

start:

mov ax,@data

mov ds,ax

mov si,offset a

mov cx,0

L1:

inc si

inc cx

cmp [si],'$'

jne L1

L2:

dec si

mov dl,[si]

mov ah,02h

int 21h

loop L2

end start

2. Repeat the program for 16-bit string

**Code:**

.data

a dw "Micro processor$"

.code

start:

mov ax,@data

mov ds,ax

mov si,offset a

mov cx,0

L1:

inc si

inc cx

cmp [si],'$'

jne L1

L2:

dec si

mov dx,[si]

mov ah,02h

int 21h

loop L2

end start

**Experiment 5**

1. Write a program to check whether the given string is palindrome or not

**Code:**

.data

a db "AMDMA$"

b db "Palindrome$"

c db "Not Palindrome$"

.code

start:

mov ax,@data

mov ds,ax

lea si,a

lea di,a

mov cl,0

L1:

inc si

inc cl

cmp [si],'$'

jne L1

dec cl

L2:

dec si

mov al,[si]

mov bl,[di]

cmp al,bl

jne L3

inc di

loop L2

lea dx,b

mov ah,09h

int 21h

hlt

L3:

lea dx,c

mov ah,09h

int 21h

end start

2. Repeat the program for 16-bit string

**Code:**

.data

a dw "AMDMA$"

b dw "Palindrome$"

c dw "Not Palindrome$"

.code

start:

mov ax,@data

mov ds,ax

lea si,a

lea di,a

mov cl,0

L1:

inc si

inc cl

cmp [si],'$'

jne L1

dec cl

L2:

dec si

mov al,[si]

mov bl,[di]

cmp al,bl

jne L3

inc di

loop L2

lea dx,b

mov ah,09h

int 21h

hlt

L3:

lea dx,c

mov ah,09h

int 21h

end start

**Experiment 6**

1. Write a program to sum two 8-bit single digit numbers

**Code:**

.data

a db "Enter the first number:$"

b db "Enter the second number:$"

c db “Sum is:$"

.code

start:

mov ax,data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

mov ah,01h

int 21h

mov bl,al

mov dx,offset b

mov ah,09h

int 21h

mov ah,01h

int 21h

add al,bl

mov ah,0

aaa

mov bx,ax

add bx,3030h

mov dx,offset c

mov ah,09h

int 21h

mov dl,bh

mov ah,02h

int 21h

mov dl,bl

mov ah,02h

int 21h

end start

2. Write a program to sum two 16-bit single digit numbers

**Code:**

.data

a db "Enter the first number:$"

b db "Enter the second number:$"

c db "Sum is:$"

.code

start:

mov ax,data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

mov ah,01h

int 21h

mov bl,al

mov dx,offset b

mov ah,09h

int 21h

mov ah,01h

int 21h

add al,bl

mov ah,0

aaa

mov bx,ax

add bx,3030h

mov dx,offset c

mov ah,09h

int 21h

mov dl,bh

mov ah,02h

int 21h

mov dl,bl

mov ah,02h

int 21h

end start

3. Write a program to sum two 8-bit multi digit numbers

**Code:**

**.data**

a db "Enter the first 2 digit number: $"

b db "Enter the second 2 digit number: $"

c db "Sum is: $"

n db 10

lea dx,a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov bh,al

int 21h

sub al,30h

mov bl,al ; bh contains higher digit and bl contains lower digit

; 2nd number

mov ah,02h

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,b

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov ch,al

int 21h

sub al,30h

mov cl,al ; ch contains higher digit and cl contains lower digit

mov ah,02h

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,c

mov ah,09h

int 21h

; Perform addition

mov ax,cx

add ax,bx

aaa

mov bl,al

mov al,ah

mov ah,0h

aaa

mov cx,ax

mov ah,02h

mov dl,ch

add dl,30h

int 21h

mov dl,cl

add dl,30h

int 21h

mov dl,bl

add dl,30h

int 21h

hlt

4. Write a program to subtract two 8-bit single digit numbers

**Code:**

.data

a db "Enter the first number:$"

b db “Enter the second number:$"

c db “Difference is:$"

.code

start:

mov ax,data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov bl,al

mov dx,offset b

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov cl,al

mov al,bl

sub al,cl

mov ah,0

aas

mov bl,al

add bl,30h

mov dx,offset c

mov ah,09h

int 21h

mov dl,bl

mov ah,02h

int 21h

end start

5. Write a program to subtract two 8-bit multi digit numbers

**Code:**

lea dx, a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov bh,al

mov ah,01h

int 21h

sub al,30h

mov bl,al

; 2nd number

mov ah,02h

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,b

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov ch,al

mov ah,01h

int 21h

sub al,30h

mov cl,al

mov ah,02h

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,c

mov ah,09h

int 21h

cmp bh,ch

JL exchange

JMP subtract

exchange:

mov ah,02h

mov dl,'-'

int 21h

xchg bx,cx

subtract:

mov ax,bx

sub al,cl

aas

sub ah,ch

aas

add ax,3030h

mov dx,ax

mov ah,02h

xchg dh,dl

int 21h

xchg dh,dl

int 21h

hlt

a db "Enter the first 2 digit number: $"

b db "Enter the second 2 digit number: $"

c db "Difference is: $"

n db 10

6. Write a program to multiply two 8-bit single digit numbers

**Code:**

.data

a db "Enter the first number:$"

b db “Enter the second number:$"

c db “Product is:$"

.code

start:

mov ax,data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov bl,al

mov dx,offset b

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mul bl

mov ah,0

aam

mov bx,ax

add bx,3030h

mov dx,offset c

mov ah,09h

int 21h

mov dl,bh

mov ah,02h

int 21h

mov dl,bl

mov ah,02h

int 21h

end start

7. Write a program to divide two 8-bit single digit numbers

**Code:**

.data

a db "Enter the first number:$"

b db “Enter the second number:$"

c db "Division is:$"

.code

start:

mov ax,@data

mov ds,ax

mov dx,offset a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov bl,al

mov dx,offset b

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov cl,al

mov al,bl

mov ah,0

div cl

mov bx,ax

add bx,3030h

mov dx,offset c

mov ah,09h

int 21h

mov dl,bh

mov ah,02h

int 21h

mov dl,bl

mov ah,02h

int 21h

end start

**Experiment 7**

1. Write a program to convert single digit decimal number into hexadecimal

**Code:**

lea dx,a

mov ah,09h

int 21h

mov ah,01h

int 21h

sub al,30h

mov ah,0

div hex

mov bh,ah

mov ah,02h

mov dl,10

int 21h

mov dl,13

int 21h

mov ah,09h

lea dx,b

int 21h

mov dl,bh

add dl,30h

mov ah,02h

int 21h

a db "Enter the single digit decimal number: $"

b db "Hexadecimal number is: $"

hex db 16

**Experiment 8**

1. Write a program to find that 8-bit number is positive or negative

**Code:**

.DATA

MSG1 DW "ENTER A NUMBER:$"

MSG2 DW "NUMBER IS POSITIVE$"

MSG3 DW "NUMBER IS NEGATIVE$"

NUM1 DW 9925H

NUM2 DW 2851H

.CODE

START:

MOV AX, @DATA

MOV DS, AX

MOV DX, OFFSET MSG1

MOV AH, 09H

INT 21H

MOV AH, 01H

INT 21H

MOV BL, AL

MOV AH, 01H

INT 21H

CMP BL, '-'

JZ l

;PRINT POSITIVE

MOV DX, OFFSET MSG2

MOV AH, 09H

INT 21H

HLT

l:

;NEGATIVE

MOV DX, OFFSET MSG3

MOV AH, 09H

INT 21H

END START

**Experiment 9**

1. Write a program to find that 8-bit number is even or odd

**Code:**

.DATA

MSG1 DW "ENTER A NUMBER:$"

MSG2 DW "NUMBER IS EVEN$"

MSG3 DW “NUMBER IS ODD$"

.CODE

START:

MOV AX, @DATA

MOV DS, AX

MOV DX, OFFSET MSG1

MOV AH, 09H

INT 21H

MOV AH, 01H

INT 21H

MOV DX, 0H

MOV BX, 02H

DIV BX

CMP DX, 0H

JNZ LABEL

;PRINT EVEN

MOV DX, OFFSET MSG2

MOV AH, 09H

INT 21H

HLT

LABEL:

;ODD

MOV DX, OFFSET MSG3

MOV AH, 09H

INT 21H

END START

**Experiment 10**

1. Write a program to find the factorial of a given number

**Code:**

.STACK 100h

.data

a DB "Enter the number: $"

b DB "Factorial of the number $"

.code

start:

MOV AX,@data

MOV DS,AX

MOV DX,OFFSET a

MOV AH,09h

INT 21h

MOV AH,01h

INT 21h

SUB AL,30h

MOV CH,0

MOV CL,AL

MOV AX,1

l:MUL CX

DEC CX

CMP CX,0

JNE l

MOV BX,10

MOV CL,0

m:MOV DX,0

DIV BX

PUSH DX

INC CL

CMP AX,0

JNE m

n:POP DX

ADD DX,30h

MOV AH,02h

INT 21h

DEC CL

CMP CL,0

JNE n

MOV DX,OFFSET b

MOV AH,09h

INT 21h

END start

**Experiment 11**

1. Write a program to print the Fibonacci series up to 233

**Code:**

.data

a DB "Enter the number of terms: $"

.code

start:

MOV AX,@data

MOV DS,AX

MOV DX,OFFSET a

MOV AH,09h

INT 21h

MOV AH,01h

INT 21h

MOV BH,AL

MOV AH,01h

INT 21h

MOV AH,BH

SUB AX,3030h

AAD

MOV BH,AL

MOV DL,32

MOV AH,02h

INT 21h

MOV DL,48

MOV AH,02h

INT 21h

MOV DL,32

MOV AH,02h

INT 21h

MOV DL,49

MOV AH,02h

INT 21h

DEC BH

DEC BH

MOV CX,01

MOV SI,00

l:MOV DI,CX

ADD CX,SI

MOV SI,DI

MOV AX,CX

MOV DI,10

MOV BL,0

m:MOV DX,0

DIV DI

ADD DX,48

PUSH DX

INC BL

CMP AX,0

JNE m

MOV DL,32

MOV AH,02h

INT 21h

p:POP DX

MOV AH,02h

INT 21h

DEC BL

CMP BL,0

JNE p

DEC BH

CMP BH,0

JNE l

END start