Name – Mohiuddin Mondal
Roll – 001910501043
System programming
Assignment 2

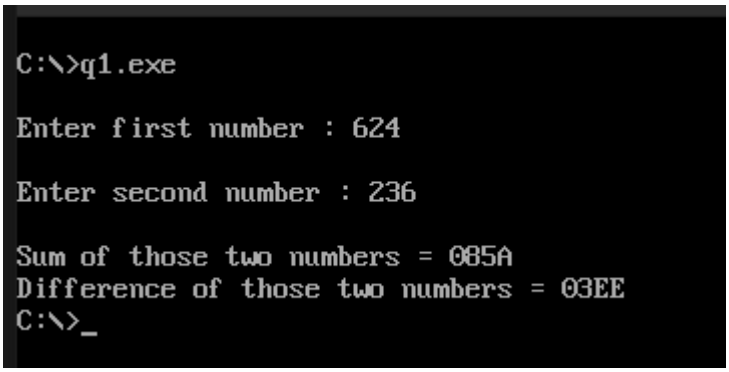
Q1) Write and test a MASM program to add and subtract two 16 bit numbers:

```
.MODEL SMALL
.STACK 100H
.DATA
  inputMessage DB 10,13,'Enter first number : $'
  inputMessage2 DB 10,13,'Enter second number : $'
  outputMessage1 DB 10,13,'Sum of those two numbers = $'
  outputMessage2 DB 10,13,'Difference of those two numbers = $'
  N1 DW ?
  N2 DW ?
  n_line DB 0DH,0AH,"$"

.CODE

  INPUT PROC ;OUTPUT: IN BX

     XOR BX,BX
     MOV CL,4
     MOV AH,1
     INT 21h

      INPUT1:
          CMP AL,0DH
          JE LINE1

          CMP AL,39h
          JG LETTER1

          AND AL,0FH
          JMP SHIFT1

      LETTER1:
          SUB AL,37H

      SHIFT1:
          SHL BX,CL
          OR  BL,AL

          INT 21h
          JMP INPUT1
      LINE1:
          RET
    INPUT ENDP

    OUTPUT PROC
      MOV CL,4
      MOV CH,BH
      SHR CH,CL
      AND CH,0FH
      CMP CH,10
      ADD CH,'0'
      CMP CH,':'
      JL P1
      ADD CH,7

      P1:
        MOV DL,CH
        MOV AH,2
        INT 21H

        MOV CH,BH
        AND CH,0FH
        CMP CH,10
        ADD CH,'0'
        CMP CH,':'
        JL P2
        ADD CH,7

      P2:
        MOV DL,CH
```

```
      MOV AH,2
      INT 21H

      MOV CH,BL
      SHR CH,CL
      AND CH,0FH
      CMP CH,10
      ADD CH,'0'
      CMP CH,':'
      JL P3
      ADD CH,7
   P3:
      MOV DL,CH
      MOV AH,2
      INT 21H

      MOV CH,BL
      AND CH,0FH
      CMP CH,10
      ADD CH,'0'
      CMP CH,':'
      JL P4
      ADD CH,7
   P4:
      MOV DL,CH
      MOV AH,2
      INT 21H

      JMP QUIT
   PC1:
      MOV DL,'1'
      MOV AH,2
      INT 21h
      JMP OUTPUT
   QUIT:
      RET
 OUTPUT ENDP

MAIN PROC
   MOV AX,@DATA
   MOV DS,AX

   LEA DX, inputMessage
   MOV AH,9
   INT 21H

   CALL INPUT
   MOV N1,BX

   LEA DX, inputMessage2
   MOV AH,9
   INT 21H

   CALL INPUT
   MOV N2,BX

   LEA DX, outputMessage1
   MOV AH,9
   INT 21H

   SUM:
       MOV BX,N1
       MOV CX,N2
       ADD BX,CX
       JNC OP
       MOV DL,49
       MOV AH,2
       INT 21H
   OP:
       CALL OUTPUT

   LEA DX, outputMessage2
   MOV AH,9
   INT 21H

   DIFF:
       MOV BX,N1
       MOV CX,N2
       SUB BX,CX
       CALL OUTPUT
```

```
        MOV AH,4CH
        INT 21H

  MAIN ENDP
END MAIN
```

output:

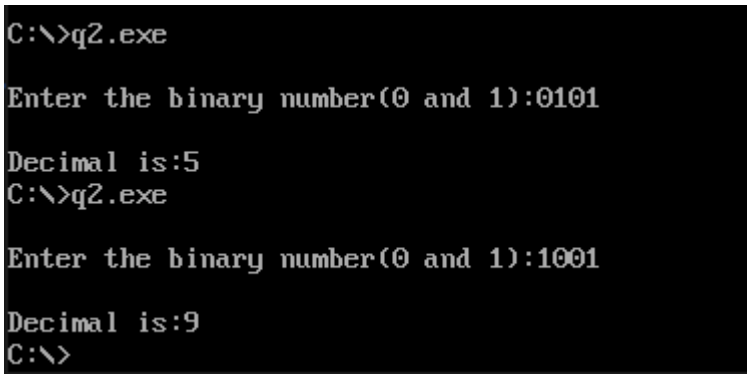


Q2) Write and test a MASM program to convert Binary digit to Decimal and vice versa.
Code:

```
.model small
printmsg macro msg
        lea dx,msg
        mov ah,09h
        int 21h
endm
.data
        inputMsg1 db 0ah,0dh,"Enter the binary number(0 and 1):$"
        inputMsg2 db 0ah,0dh,"Decimal is:$"
        inputMsg3 db 0ah,0dh,"Invalid Number$"
        spc db 0ah,0dh,"$"
        num dw 0
        cnt dw 0
.code
        mov ax,@data
        mov ds,ax
        printmsg inputMsg1
        xor bx,bx
     l2:mov cx,0
        mov ah,01h
        int 21h
        cmp al,0dh
        je l1
        cmp al,30h
        jb l3
        cmp al,31h
        ja l3
        sub al,30h
        and ax,00ffh
        mov cx,ax
        shl bx,1
        add bx,cx
        jmp l2
      l1:  printmsg inputMsg2
           mov num,bx
           call print
           jmp e1
        l3:printmsg inputMsg3
          e1: mov ah,4ch
           int 21h
print proc near
        push ax
        push bx
        push cx
        push dx
        mov ax,num
        mov bx,10
     lu1: mov dx,00
          div bx
          push dx
          inc cnt
          cmp ax,00
          jne lu1
     lu2: cmp cnt,00
```

```
          je ext
          pop dx
          add dl,30h
          dec cnt
          mov ah,02h
          int 21h
          jmp lu2
      ext: pop dx
           pop cx
           pop bx
           pop ax
           ret
        print endp
        end
```

output:



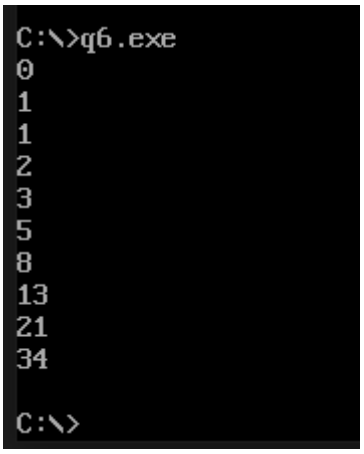
Q6) Write and test a MASM program to Print Fibonacci series up to 10 terms..

Code:

```
.model small
.stack 100h
.data
x db 0ah, 0dh, "$"
.code
main proc
mov ax, @data
mov ds, ax
mov al, 0
mov bl, 1

call displayNumber
mov al, bl
call displayNumber
mov al, 0
mov ch, 02h
l1:
mov cl, bl
add bl, al
mov al, bl
call displayNumber
mov al, cl
inc ch
cmp ch, 10
jne l1

mov ah, 4ch
int 21h
main endp

displayNumber proc
push bx
mov bl, 10
mov bh, 00h
l2: mov ah, 00h
div bl
push ax
inc bh
cmp al, 0
jne l2
l3: pop dx
mov dl, dh
mov dh, 0
add dl, 48
mov ah, 02h
int 21h
dec bh
cmp bh, 0
```

```
jne l3
lea dx, x
mov ah, 09h
int 21h
pop bx
ret

displayNumber endp
end
```

Output:



Q10)Write and test a MASM program to print prime numbers between 1 to 100.

Code:

```
.model small
.stack 100
.data
    res db 3 dup(0)
    msg db "Primes(1-100):",13,10,"$"
.code
main proc
mov ax,@data
mov ds,ax

lea dx,msg
mov ah,9
int 21H

mov dl,1
mov cx,25
l1:
       mov bl, 02
       add dl, 01h

       cmp dl, 02h
       je print
       cmp dl, 03h
       je print
       cmp dl, 04h
       jge Logic

    logic:
          mov ah, 00
          mov al, dl
          div bl
          cmp ah, 00
      je L1
          add bl, 01h
          cmp bl, al
          jle Logic
      jmp print

    print:
       mov al, dl
       mov ah,00
       call output
       loop l1
```

```
    exit:
       mov ah, 4ch
       int 21h
    ret
main endp

output proc                  ;data in ax
    push ax
    push bx
    push cx
    push dx
    MOV CX,0
    MOV BX,10
    MOV SI,OFFSET RES
LOOP1: MOV DX,0
       DIV BX
       ADD DL,30H
       PUSH DX
       INC CX
       CMP AX,9
       JG LOOP1

       ADD AL,30H
       MOV [SI],AL

LOOP2: POP AX
       INC SI
       MOV [SI],AL
       LOOP LOOP2

    MOV DL,res[0]
    MOV AH,2
    INT 21h

    MOV DL,res[1]
    INT 21h

    MOV DL,32
    INT 21H

    pop dx
    pop cx
    pop bx
    pop ax
    ret
output endp
end main
```

```
C:\>q10.exe
Primes(1-100):
02 03 05 07 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
C:\>_
```

Q12) Write and test a MASM program to rename a file.

Code:

```
.model small
.stack 64
.data
      msg1 db 0AH,0DH,'Enter old file name: $'
      msg2 db 0AH,0DH,'Enter new file name: $'
```

```
       old db 80 dup('$')
       new db 80 dup('$')
       sucmsg db 'has been renamed to $'
        failmsg db 'not found. ERROR!!!$'

.code
print macro msg
       push ax
       push dx
       mov ah, 09h
       lea dx, msg
       int 21h
       pop dx
       pop ax
endm

main proc
        mov ax,@data
        mov ds,ax
        mov es,ax

       print msg1
       lea SI, old
       call readstring

       print msg2
       lea SI, new
       call readstring

       mov ax,@data
       mov ds,ax
       mov es,ax
        lea dx,old  ;ds:dx points to the ASCIIZ string old,0
        lea di,new  ;es:di points to the ASCIIZ string new,0
        mov ah,56h  ;DOS function 56h is used for renaming
        int 21h
        jc error    ;if there is an error carry flag is set
        print old
       print sucmsg
       print new
        jmp exit
error:
       print old
        print failmsg

exit:
        mov ah,4ch
        int 21h
main endp

readstring proc near
read:
       mov ah, 01h
       int 21h
       cmp al, 13
       je done
       mov [SI],al
       inc SI
       jmp read

done:
        mov al, 0
        mov [SI],al
       ret
readstring endp

end main
```

```
C:\>q12.exe

Enter old file name: q2.obj

Enter new file name: q22.obj
q2.obj has been renamed to q22.obj
C:\>
```