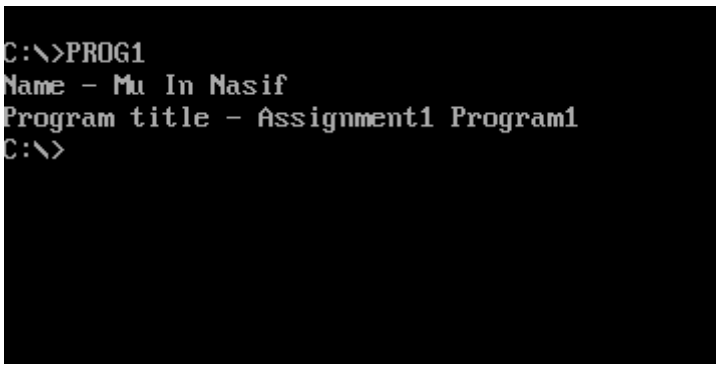
Name: Mu In Nasif
Roll: 001910501036
System Programming
Assignment 1

Q1: Write and test a MASM program to Display your name and program title on the output screen.
Code:

```
.model small
.stack 100h
.data
     x db"Name - Mu In Nasif$"
     y db"Program title - Assignment1 Program1$"
.code
     main proc
          mov ax,@data
          mov ds,ax
          lea dx,x
          mov ah,09h
          int 21h

          mov dl,0ah
          mov ah,02h
          int 21h

          mov dl,0dh
          mov ah,02h
          int 21h

          lea dx,y
          mov ah,09h
          int 21h

          mov ah,4ch
          int 21h

     main endp
end
```

output:



Q2: Write and test a MASM program to convert a letter from uppercase to lowercase.

Code:

```
.model small
.stack 100h
.data
.code
     main proc
          mov ah,01h
          int 21h

          add al,32
          mov bl,al
          mov dl,0ah
          mov ah,02h
          int 21h

          mov dl,0dh
          mov ah,02h
          int 21h

          mov dl,bl
          mov ah,02h
          int 21h

          mov ah,4ch
          int 21h

     main endp
end
```

output:

Q3: Write and test a MASM program to add two Hexadecimal Numbers

Code:

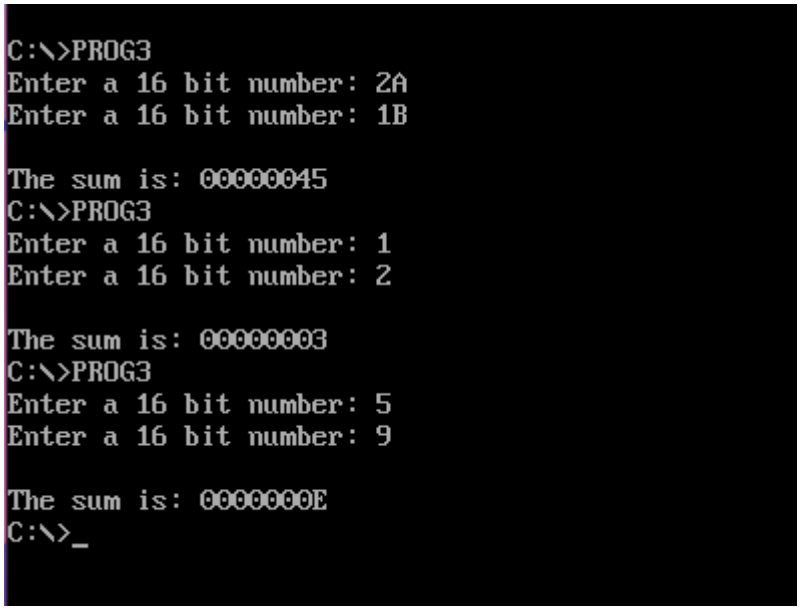
```
.model small
.stack 100h

.data
     inputPrompt DB 13,"Enter a 16 bit number: $"
     num1 DW ?
     num2 DW ?
     num DW ?
     sum DW ?
     sumCarry DW 00h
     sumPrompt DB 10,"The sum is: $"

.code
     mov ax,@data
     mov ds,ax
     call main
     mov ah,4ch
     int 21h

main proc

    ;taking input for first number
    lea dx, inputPrompt
    mov ah,9
    int 21h
    call getNum
    mov num1,ax

    ;taking input for second number
    lea dx, inputPrompt
    mov ah,9
    int 21h
    call getNum
    mov num2,ax

    ;performing addition
    mov ax,num1
    add ax,num2
    jnc noCarry
    inc sumCarry
    noCarry:
    mov sum,ax

    ;output for sum
    lea dx,sumPrompt
```

```
    mov ah,9
    int 21h
    mov ax,sumCarry
    mov num,ax
    call outputNum
    mov ax,sum
    mov num,ax
    call outputNum

    ret
main endp

;function to take 16 bit number input
getNum proc
    push cx
    push dx

    mov dx, 0000
    mov ax, 0000
    mov cl, 4

    getNumber:

    call getChar
    cmp al, 13
    je inputDone
    cmp al, 10
    je inputDone

    shl dx,cl
    sub al, 48
    cmp al, 9
    jle isNumber
    sub al, 7

    isNumber:
    or dl, al

    jmp getNumber

    inputDone:
    mov ax, dx

    pop dx
    pop cx
    ret
getNum endp


;fucntion to take in character
getChar proc
    mov ah,1
    int 21h
```

```
    ret
getChar endp


; program to output a 16 bit number stored in num
outputNum proc
    push cx
    push dx


    mov cl, 4
    mov dx, num
    mov dl,dh
    shr dl,cl
    and dl, 0fh
    cmp dl,0ah
    jl isNumber4
    add dl,7
    isNumber4:
    add dl,48
    mov ah, 2
    int 21h

    mov dx, num
    mov dl,dh
    and dl, 0fh
    cmp dl, 0ah
    jl isNumber3
    add dl, 07h
    isNumber3:
    add dl,48
    mov ah, 2
    int 21h


    mov cl, 4
    mov dx, num
    shr dl,cl
    and dl, 0fh
    cmp dl,0ah
    jl isNumber2
    add dl,7
    isNumber2:
    add dl,48
    mov ah, 2
    int 21h

    mov dx, num
    and dl, 0fh
    cmp dl, 0ah
    jl isNumber1
    add dl, 07h
    isNumber1:
```

```
    add dl,48
    mov ah, 2
    int 21h

    pop dx
    pop cx
    ret
outputNum endp

end
```

output:



Q4: Write and test a MASM program to find the second max and second min from an array.

Code :

```
.model small
.stack 100h

.data
prompt_0  db  'enter the number of array elements :',0dh,0ah,'$'
prompt_1  db  'enter the array elements :',0dh,0ah,'$'
prompt_2  db  'the 2nd maximum is : $'
prompt_3  db  'the 2nd minimum is : $'

array   dw  50 dup(0)

s dw ?
max dw ?
min dw ?

.code
main proc

            mov ax, @data               ; initialize ds
            mov ds, ax

            lea dx, prompt_0            ; load and display the string prompt_0
            mov ah, 9
            int 21h

            mov ah,1                                    ;for taking input
            int 21h

            input1:
            cmp al,0dh                                  ;compare whether the
pressed key is 'enter' or not
            je line1                                    ;if it is equal to
'enter' then stop taking first value

            and al,0fh                                  ;convert it's ascii
value to real value by masking

            shl bx, 1
            shl bx, 1
            shl bx, 1
            shl bx, 1
            or  bl,al                                   ;making 'or' will add
the current value with previous value

            int 21h
            jmp input1

            line1:
            lea dx, prompt_1            ; load and display the string prompt_1
            mov ah, 9
            int 21h

            lea si, array               ; set si=offset address of array
```

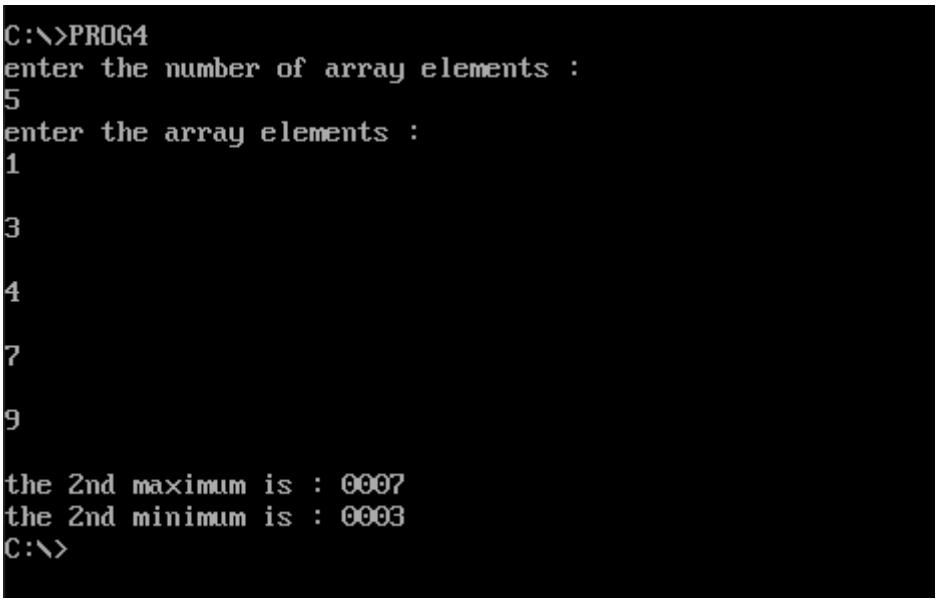
```
            mov s,bx
            mov cx, bx                      ; set cx=bx


            @read_array:                    ; loop label

            mov ah,1                                    ;for taking input
            int 21h

            xor dx,dx

            input2:
            cmp al,0dh                                  ;compare whether the
pressed key is 'enter' or not
            je line2                                   ;if it is equal to
'enter' then stop taking first value

            and al,0fh                                 ;convert it's ascii value
to real value by masking

            shl dx,1
            shl dx,1
            shl dx,1
            shl dx,1
            or  dl,al                                   ;making 'or' will add
the current value with previous value

            int 21h
            jmp input2

            line2:
            mov [si], dx                ; set [si]=ax
            add si, 2                   ; set si=si+2

            mov dl, 0ah                 ; line feed
            mov ah, 2                   ; set output function
            int 21h                     ; print a character

            loop @read_array              ; jump to label @read_array while cx!
=0
            ; array input done

            lea si,array
            mov ax,bx
            dec ax
            xor bx,bx
            xor cx,cx
            mov bx,word ptr[si]     ;store the maximum
            mov cx,word ptr[si]     ;store the 2nd
            add si, 2

            ; loop to find max and 2nd max
            arrayloop2:

            cmp word ptr[si],bx
            jl max2
            mov cx,bx
            mov bx,word ptr[si]

            max2:
            cmp word ptr[si],cx
            jl incre
            cmp word ptr[si],bx
            je incre
```

```
mov cx,word ptr[si]

incre:
add si, 2
dec ax

jnz arrayloop2

; now bx has max cx has 2nd max
mov max,bx
; displaying the prompt
lea dx,prompt_2
mov ah,09h
int 21h

; display contents of cx
mov bx,cx

mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh

add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
and dh,0fh
cmp dh,10
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dl, 0ah                    ; line feed
mov ah, 2                      ; set output function
int 21h                        ; print a character

lea si,array
mov ax,s
dec ax
mov bx,max
```

```
; loop to find min and 2nd min
arrayloop3:

cmp word ptr[si],bx
jg min2
mov cx,bx
mov bx,word ptr[si]

min2:
cmp word ptr[si],cx
jg incre2
cmp word ptr[si],bx
je incre2
mov cx,word ptr[si]

incre2:
add si, 2
dec ax

jnz arrayloop3

; now bx has min cx has 2nd min

; displaying the prompt
lea dx,prompt_3
mov ah,09h
int 21h

; display contents of cx
mov bx,cx

mov dh,bh
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh

add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bh
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
shr dh, 1
shr dh, 1
shr dh, 1
shr dh, 1
and dh,0fh
add dh,'0'
mov dl,dh
mov ah,2
int 21h

mov dh,bl
```

```
            and dh,0fh
            cmp dh,10
            add dh,'0'
            mov dl,dh
            mov ah,2
            int 21h


            exit:
            mov ah, 4ch                                ;return control to dos
            int 21h

main endp
end main
```

output:



Q5: Write and test a MASM program to display a terminating message.

Code :

```
.MODEL SMALL
.STACK 100H

.DATA
    TERM_PROMPT DB '::::::::::TERMINATE MESSAGE::::::::::$'

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

        LEA DX, TERM_PROMPT
        MOV AH,9
        INT 21H

        MOV AH,4CH
        INT 21H
    MAIN ENDP
END MAIN
```

output:

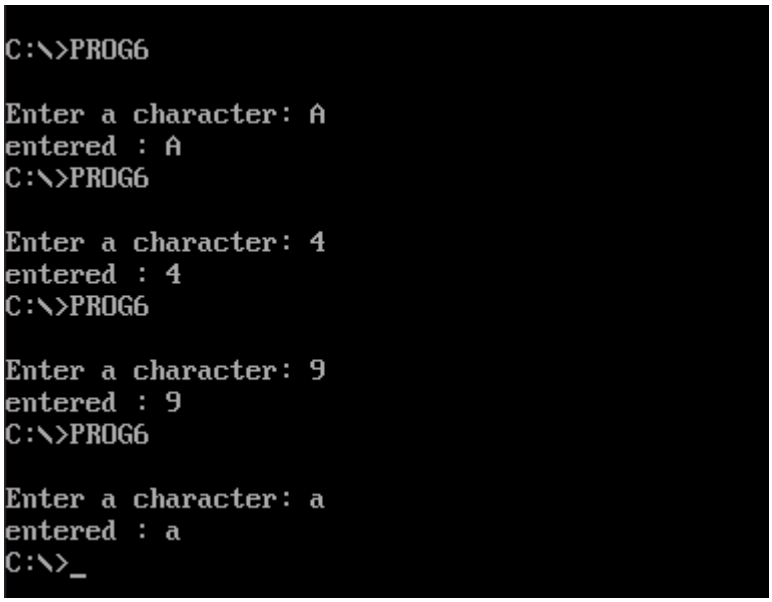
Q6: Write and test a MASM program to Take a character from the keyboard and print it.

Code:

```
.model small
.stack 100H

.data
    msg1 db 10,13,"Enter a character: $"
    msg2 db 10,13,"entered : $"
```

```
.code
    main proc

        mov ax,@data
        mov ds,ax

        ;display input prompt
        lea dx, msg1
        mov ah,09h
        int 21h

        ;accept a character
        mov ah, 01h
        int 21h

        ;al has the character
        ;display prompt
        lea dx, msg2
        mov ah,09h
        int 21h

        ;display the character
        mov dl,al
        mov ah,02h
        int 21h

        mov ah, 4ch
        int 21h

    main endp
end main
```

output:



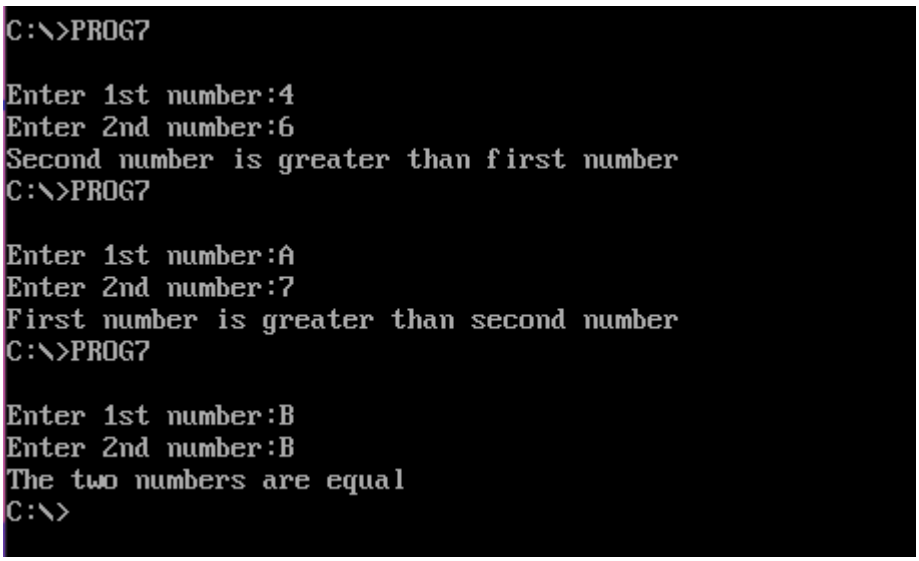
Q7: Write and test a MASM program to validate second numbers is less than the first.

Code :

```
.model small
.stack 100h
.data
     mes1 db 0ah,0dh,"Enter 1st number:$"
     mes2 db 0ah,0dh,"Enter 2nd number:$"
     mes3 db 0ah,0dh,"First number is greater than second number$"
     mes4 db 0ah,0dh,"Second number is greater than first number$"
     mes5 db 0ah,0dh,"The two numbers are equal$"
.code
     main proc
          mov ax,@data
          mov ds,ax
          lea dx,mes1
          mov ah,09h
          int 21h

          mov ah,01h
          int 21h

          mov bl,al
          lea dx,mes2
          mov ah,09h
          int 21h

          mov ah,01h
          int 21h

          cmp bl,al
          jg cond1
          jl cond2
          lea dx,mes5
          mov ah,09h
          int 21h

          mov ah,4ch
          int 21h

     cond1:
          lea dx,mes3
          mov ah,09h
          int 21h

          mov ah,4ch
          int 21h
     cond2:
```

```
          lea dx,mes4
          mov ah,09h
          int 21h

          mov ah,4ch
          int 21h
main endp
end
```

output:



Q8: Write andtest a MASM program to find maximum and minimum from an array.

Code :

```
.model small
.stack 100h

.data
prompt_0  db  'enter the number of array elements :',0dh,0ah,'$'
prompt_1  db  'enter the array elements :',0dh,0ah,'$'
prompt_2  db  'the maximum is : $'
prompt_3  db  'the minimum is : $'

array   dw  50 dup(0)

s dw ?

.code
main proc

          mov ax, @data                ; initialize ds
          mov ds, ax
```

```
          lea dx, prompt_0             ; load and display the
string prompt_0
          mov ah, 9
          int 21h

          mov ah,1                                          ;for taking
input
          int 21h

          input1:
          cmp al,0dh                                        ;compare
whether the pressed key is 'enter' or not
          je line1                                          ;if it is
equal to 'enter' then stop taking first value

          and al,0fh                                        ;convert it's
ascii value to real value by masking

          shl bx, 1
          shl bx, 1
          shl bx, 1
          shl bx, 1
          or  bl,al                                         ;making 'or'
will add the current value with previous value

          int 21h
          jmp input1

          line1:
          lea dx, prompt_1             ; load and display the
string prompt_1
          mov ah, 9
          int 21h

          lea si, array                ; set si=offset address of
array

          mov cx, bx                     ; set cx=bx


          @read_array:                   ; loop label

          mov ah,1                                          ;for taking
input
          int 21h

          xor dx,dx

          input2:
          cmp al,0dh                                        ;compare
whether the pressed key is 'enter' or not
          je line2                                         ;if it is
equal to 'enter' then stop taking first value
```

```
          and al,0fh                                        ;convert it's
ascii value to real value by masking

          shl dx,1
          shl dx,1
          shl dx,1
          shl dx,1
          or  dl,al                                          ;making 'or'
will add the current value with previous value

          int 21h
          jmp input2

          line2:
          mov [si], dx                ; set [si]=ax
          add si, 2                   ; set si=si+2

          mov dl, 0ah                 ; line feed
          mov ah, 2                   ; set output function
          int 21h                     ; print a character

          loop @read_array              ; jump to label
@read_array while cx!=0
          ; array input done

          lea si,array
          mov ax,bx
          dec ax
          xor bx,bx
          xor cx,cx
          mov bx,word ptr[si] ;store the maximum
          mov cx,word ptr[si] ;store the minimum
          add si, 2

          ; loop to find max and min
          arrayloop2:

          cmp word ptr[si],bx
          jg maximum

          cmp word ptr[si],cx
          jl minimum

          jmp incre
          maximum:
          mov bx,word ptr[si]
          jmp incre

          minimum:
          mov cx,word ptr[si]

          incre:
```

```
            add si, 2
            dec ax

            jnz arrayloop2

            ; displaying the prompt
            lea dx,prompt_2
            mov ah,09h
            int 21h

            ; display contents of bx
            output:                                     ;level for
printing their sum

            mov dh,bh
            shr dh, 1
            shr dh, 1
            shr dh, 1
            shr dh, 1
            and dh,0fh

            add dh,'0'
            mov dl,dh
            mov ah,2
            int 21h

            mov dh,bh
            and dh,0fh
            add dh,'0'
            mov dl,dh
            mov ah,2
            int 21h

            mov dh,bl
            shr dh, 1
            shr dh, 1
            shr dh, 1
            shr dh, 1
            and dh,0fh
            add dh,'0'
            mov dl,dh
            mov ah,2
            int 21h

            mov dh,bl
            and dh,0fh
            cmp dh,10
            add dh,'0'
            mov dl,dh
            mov ah,2
            int 21h

            mov dl, 0ah                    ; line feed
```

```
        mov ah, 2                       ; set output function
        int 21h                         ; print a character

        ; displaying the prompt
        lea dx,prompt_3
        mov ah,09h
        int 21h

        ; display contents of cx
        mov bx,cx

        mov dh,bh
        shr dh, 1
        shr dh, 1
        shr dh, 1
        shr dh, 1
        and dh,0fh

        add dh,'0'
        mov dl,dh
        mov ah,2
        int 21h

        mov dh,bh
        and dh,0fh
        add dh,'0'
        mov dl,dh
        mov ah,2
        int 21h

        mov dh,bl
        shr dh, 1
        shr dh, 1
        shr dh, 1
        shr dh, 1
        and dh,0fh
        add dh,'0'
        mov dl,dh
        mov ah,2
        int 21h

        mov dh,bl
        and dh,0fh
        cmp dh,10
        add dh,'0'
        mov dl,dh
        mov ah,2
        int 21h


        exit:
        mov ah, 4ch                                 ;return control
to dos
```

```
          int 21h

main endp
end main
```

output:

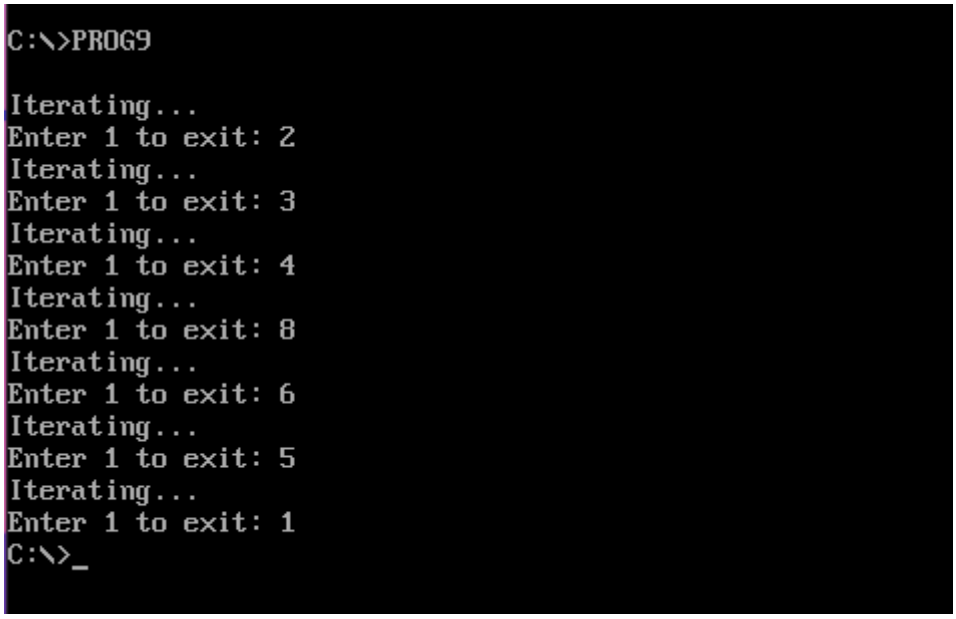
Q9: Write and test a MASM program to loopuntil the user decides to quit.

Code :

```
.model small
.stack 100h
.data
     mes1 db 0dh, 0ah, "Enter 1 to exit: ", "$"
     mes2 db 0dh, 0ah, "Iterating...", "$"
.code
     main proc
          mov ax,@data
          mov ds,ax
     loop1:
          lea dx,mes2
          mov ah,09h
          int 21h
```

```
          lea dx,mes1
          mov ah,09h
          int 21h

          mov ah,01h
          int 21h

          cmp al,'1'
          jne loop1
          mov ah,4ch
          int 21h

     main endp
end
```

output:



Q10: Write and test a MASM program to print all the characters from A-Z.

Code :

```
.model small
.stack 100h
.data
.code
     main proc
          mov cx,26
          mov dl,'A'
     looplabel:
          mov ah,02h
          int 21h

          inc dl
          loop looplabel
          mov ah,4ch
          int 21h

     main endp
end
```

output: