**CSE341: Microprocessors**

**Lab Assignment 01**

**Name:** Umme Abira Azmary

**ID:** 20101539

**Section:** 10

**Array:**

**Answer to the question no - 1:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter name length: $"

str2 dw "Enter your name: $"

str3 dw "Your name is: $"

size1 db ?

arr1 db 25 dup(?)

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

;take input name length

mov ah, 1

int 21h

sub al, 30h

mov size1, al

mov ch, 0

mov cl, size1

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str2

int 21h

;loop

mov si, 0

loop\_start:

mov ah, 1

int 21h

mov arr1[si], al

add si, 1

loop loop\_start

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str3

int 21h

;print loop

mov ch, 0

mov cl, size1

mov ah,2

mov si,0

start:

mov dl,arr1[si]

int 21h

add si,1

loop start

;exit to DO

MAIN ENDP

END MAIN

**Answer to the question no - 2:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter values of the Array: $"

str2 dw "Values in Ascending order: $"

size1 db ?

size2 db ?

arr1 db 5 dup(?)

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

;loop

mov ch, 0

mov cl, 5

mov si, 0

enter\_value:

mov ah, 1

int 21h

mov arr1[si], al

add si, 1

loop enter\_value

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str2

int 21h

;ascending using sorting

mov ch, 0

mov cl, 5

func1:

mov dx, 0

func2:

mov si, dx

mov bh, arr1[si]

add si, 1h

mov bl, arr1[si]

mov size1, bh

mov size2, bl

cmp bh, bl

jg swap

jle no\_swap

swap:

mov arr1[si], bh

sub si, 1h

mov arr1[si], bl

no\_swap:

inc dx

cmp dx, 4

jle func2

loop func1

;output

mov si, 0

sorting\_vals:

mov dl, arr1[si]

mov ah, 2

int 21h

inc si

cmp si, 6

jl sorting\_vals

;exit to DOS

MAIN ENDP

END MAIN

**Answer to the question no - 3:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter three values: $"

str2 dw "The highest value is: $"

size1 db ?

size2 db ?

arr1 db 3 dup(?)

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

mov ch, 0

mov cl, 3

;loop

mov si, 0

enter\_value:

mov ah, 1

int 21h

mov arr1[si], al

add si, 1

loop enter\_value

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str2

int 21h

;ascending using sorting

mov ch, 0

mov cl, 3

func1:

mov dx, 0

func2:

mov si, dx

mov bh, arr1[si]

add si, 1h

mov bl, arr1[si]

mov size1, bh

mov size2, bl

cmp bh, bl

jg swap

jle no\_swap

swap:

mov arr1[si], bh

sub si, 1h

mov arr1[si], bl

no\_swap:

inc dx

cmp dx, 2

jle func2

loop func1

;output

mov si, 3

sorting\_vals:

mov dl, arr1[si]

mov ah, 2

int 21h

;exit to DOS

MAIN ENDP

END MAIN

**STACK:**

**Answer to the question no - 4:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter a number: $"

str2 dw "Factors are: $"

size1 db ?

size2 db ?

size3 db ?

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

;input

mov ah, 1

int 21h

sub al, 30h

mov size1, al

;loop

mov ch, 0

mov cl, size1

factoring:

mov ah, 0

mov al, size1

div cl

mov size2, al

mov size3, ah

mov dl, 0

cmp ah, dl

je stack\_val

jne no\_val

stack\_val:

mov ah, 0

mov al, cl

push ax

no\_val:

loop factoring

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str2

int 21h

stack\_pop:

cmp sp, 100h

je exit

pop dx

add dx, 30h

mov ah, 2

int 21h

jmp stack\_pop

;exit to DOS

exit:

MAIN ENDP

END MAIN

**Answer to the question no - 5:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter a string and stop is using a dot(.): $"

str2 dw "Reverse of the string: $"

size1 db ?

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

;loop

mov si, 0

inp\_stack:

mov ah, 1

int 21h

mov size1, al

mov ah, 0h

cmp ax, 2Eh ;used . to stop taking input

je new\_line

je loop\_exit

;stack

push ax

inc si

jmp inp\_stack

;new line

new\_line:

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov ah, 9

lea dx, str2

int 21h

loop\_exit:

cmp sp, 100h

je exit

pop dx

mov ah, 2

int 21h

jmp loop\_exit

;exit to DOS

exit:

MAIN ENDP

END MAIN

**Answer to the question no - 6:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter a sequence of number and stop using a dot(.): $"

str2 dw "It is a unique number. $"

str3 dw "It is not a unique number. $"

size1 db ?

count db ?

zero db ?

arr1 dw 40 dup(?)

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

mov si, 0

inp\_stack:

mov ah, 1

int 21h

mov size1, al

mov ah, 0h

cmp ax, 2Eh ;used '.' symbol to stop taking input

je new\_line

je loop\_exit

;stack

push ax

inc si

jmp inp\_stack

new\_line:

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

mov cx, 0

loop\_exit:

cmp sp, 100h

je exit

pop dx

cmp cx, 0

mov si, 0

je uniq

uniq\_nums:

mov bx, arr1[si]

cmp dl, bl

je not\_uniq

jne uniq

not\_uniq:

mov ah, 9

lea dx, str3

int 21h

jmp exit2

uniq:

cmp si, cx

je inc\_val

jne inc\_val2

inc\_val2:

inc si

jmp uniq\_nums

inc\_val:

mov arr1[si], dx

inc cx

jmp loop\_exit

exit:

mov ah, 9

lea dx, str2

int 21h

;exit to DOS

exit2:

MOV AX,4C00H

INT 21H

MAIN ENDP

END MAIN

**Answer to the question no - 7:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter a number: $"

str2 dw "Prime Numbers are: $"

size1 db ?

var1 db ?

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

mov ah, 1

int 21h

sub al, 30h

cmp al,0

je notinsert

mov size1, al

;loop

mov ch, 0

mov cl, size1

loop1:

mov var1, cl

cmp cl, 1h

jle not\_prime

mov bl, 2

loop2:

mov ah, 0

mov al, var1

div bl

cmp ah, 0

je not\_prime

inc bl

cmp bl, var1

je push\_stack

jne loop2

push\_stack:

mov bh, 0

push bx

not\_prime:

loop loop1

mov al, size1

cmp al, 1h

jne twoinsert

je notinsert

twoinsert:

mov ax, 2h

push ax

notinsert:

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

;printing

mov ah, 9

lea dx, str2

int 21h

loop\_exit:

cmp sp, 100h

je exit

pop dx

add dx, 30h

mov ah, 2

int 21h

jmp loop\_exit

;exit to DOS

exit:

MAIN ENDP

END MAIN

**Answer to the question no - 8:**

.MODEL SMALL

.STACK 100H

.DATA

str1 dw "Enter a sequence of text and stop using a dot(.): $"

str2 dw "Output: $"

size1 db ?

var1 db ?

arr1 dw 80 dup(?)

store\_si dw ?

; declare variables here

.CODE

MAIN PROC

; initialize DS

MOV AX,@DATA

MOV DS,AX

; enter your code here

mov ah, 9

lea dx, str1

int 21h

;loop

mov si, 0

inp\_stack:

mov ah, 1

int 21h

mov size1, al

mov ah, 0h

cmp al, 2Eh ;used . to stop taking input

je inp

cmp al, 20h

je pop\_func

push ax

jmp inp\_stack

pop\_func:

cmp sp, 100h

je space

pop dx

mov arr1[si], dx

inc si

jmp pop\_func

space:

mov arr1[si], 20h

inc si

jmp inp\_stack

inp:

cmp sp, 100h

je new\_line

pop dx

mov arr1[si], dx

inc si

jmp inp

new\_line:

;new line

mov ah, 2

mov dl, 0AH

int 21h

mov dl, 0DH

int 21h

;printing

mov ah, 9

lea dx, str2

int 21h

mov store\_si, si

;output

mov si, 0

out\_vals:

mov dh, 0

mov dx, arr1[si]

mov ah, 2

int 21h

inc si

cmp si, store\_si

jl out\_vals

;exit to DOS

exit:

MAIN ENDP

END MAIN