CS 321 Lab
Model Answers

First Lab :

Q1: Define a character string named CO_NAME containing "Internet Services" as a constant ?

ANS: CO_NAME EQU ' Internet Services'

Q2: Define the following numeric values in data items named ITEM1 through ITEM4, respectively:

a) A 1-byte item containing the hex equivalent to decimal 71?

ANS: Item1 DB 47H

b) A 2-byte item containing an undefined value?

ANS: Item2 DB ?

c) A 3-byte item containing the binary equivalent to decimal 47?

ANS: Item3 DB 00101111b

d) A 4 item containing the consecutive word values 6,9,14,18,23,29?

ANS: Item4 DB 6,9,14,18,23,29

Q3:Use an EQU directive to redefine the value 16H as ATTRIBUTE

ANS: ATTRIBUTE EQU 16H

Second Lab

Q:Write assembly program that define two variables(BYTE1 and BYTE2) of type byte and uninitialized, then write a transfer instruction that assign value ('A') in BYTE1 and value(45H) in BYTE2 , than try to swap the values

SOLUTION :

```
org 100h

.Data
;declare variables with uninitialized value

BYTE1   DB   ?
BYTE2    DB  ?

.CODE

MOV BYTE1, 'A'

MOV BYTE2 ,  45H

;swap values between BYTE1 and BYTE2

MOV  BL,BYTE1
XCHG BL,BYTE2
MOV  BYTE1 ,BL

;return to DOS
ret
```

Third Lab (practice on arithmetic operation):

Q: Write assembly program that:

1. **define variable (BYTE1 and SUM) of type byte and uninitialized.**
2. **display ? for the user to enter a number from the keyboard .(USER SHOULD ENTER A NUMBER LESS THAN 7)**
3. **assign value that enter from the user in BYTE1 .**
4. **add '3' to BYTE1 and put the result in SUM.**
5. **display the result.**

SOLUTION :

```
org 100h
.Data

;declare variables with uninitialized value

BYTE1   DB   ?
 SUM    DB  ?

 .CODE
; read number and store it in BYTE1
MOV  AH,1
INT 21H
MOV BYTE1 , AL

; BYTE1= BYTE1+3
ADD BYTE1 , 3

;move addition from BYTE1 to Sum
MOV BH , BYTE1
MOV SUM,  BH
```

```
;New Line
MOV AH,2
MOV DL , 0AH
INT 21H
MOV DL , 0DH
INT 21H

;display the SUM
MOV AH,2
MOV DL , SUM
INT 21H


;return to DOS
ret
```

Forth Lab (practice on if statement)

Q: Write assembly program that ask User to enter his/her age and determine if he/she adult or child (child (from 0 and 5) adult (from 6 to 9))

Sample execution:

```
Output 1:
    please enter your age: 6   <-- Inputs
    you are adult
```

```
Output 2:
    please enter your age: 3   <-- Inputs
    you are child
```

Solution:

```
org 100h
.Data

MSG1   DB   ' PLEASE ENTER YOUR AGE ?  $'
MSG2   DB    0AH , 0DH ,'YOU ARE CHILD $'
MSG3   DB    0AH , 0DH ,'YOU ARE ADULT $'
MSG4   DB    0AH , 0DH ,'Illegal Input $'

.CODE
  MOV AX, @DATA
  MOV DS, AX              ; initialize DS

    LEA DX,MSG1         ; display the string MSG1
    MOV AH,9
    INT 21H

;read number
    MOV AH,1
    INT 21H
;to convert the digit from ASCII value to real value
    SUB AL , 30H

 ;check if the value <=9 then go to adult
  CMP AL , 9
  JLE ADULT

  LEA DX,MSG4        ;display the string MSG4
 MOV AH,9
 INT 21H
JMP EXIT
```

```
ADULT:
check if the value <=5 go to CHILD
  CMP AL , 5
  JLE CHILD

 LEA DX,MSG3    ;display MSG3
MOV AH,9
INT 21H

JMP EXIT


CHILD:
LEA DX,MSG2    ;display MSG2
MOV AH,9
INT 21H
EXIT :
;return to DOS
    ret
```

Fifth Lab (practice on if statement)

Q: By using For Loop Write a program that Ask user to enter a character and then display it 15 times on the next line .

Sample execution:

```
Please Enter a letter: p
p
P
..
..
P (15 times)
```

Input

Solution:

```
org 100h
.Data
mSG1  DB ' PLEASE ENTER A ONE CHARCTER ?  $'

.CODE
     MOV AX, @DATA    ; initialize DS
     MOV DS, AX

  ;display message
     LEA DX,MSG1
     MOV AH,9
     INT 21H

;read character
     mov AH,1
     INT 21H
     MOV BL,AL

     XOR CX,CX
     MOV CX , 15   ; set counter = 15

;New Line
    MOV AH,2
     MOV DL,0AH
     INT 21H
     MOV DL,0DH
    INT 21H
```

```
;for loop
FOR:
;display character
MOV AH,2
MOV DL,BL
 INT 21H

 ;New Line
MOV DL,0AH
 INT 21H
 MOV DL,0DH
 INT 21H

 LOOP FOR
;return to DOS
 RET
```

sixth Lab (practice on logic operations)

Q : write a program to multiply AX by BX and put the product in CX by repeated addition.

Sample execution:

```
Please Enter Two digits to product:  23
Result : 6
```

Input

Solution:

```
org 100h
.Data

MSG1    DB          ' PLEASE ENTER TWO DIGIT ?  $'
MSG2    DB          0AH , 0DH , '  PRODUCT $'
PRODUCT  DB   ?


.CODE
  MOV AX, @DATA     ; initialize DS
  MOV DS, AX

  ;display message
    LEA DX,MSG1
    MOV AH,9
    INT 21H

  ; read first digit
   MOV AH,1
  INT 21H
  MOV BL,AL

     ; read first digit
   MOV AH,1
   INT 21H
;to convert the digit from ASCII value to real value
   SUB BL,30H
   SUB AL,30H
```

```
WHILE_:
;IF AL < 0 go to ENDWHILE_
CMP AL ,0
JE ENDWHILE_

; PRODUCT = PRODUCT+ BL
ADD  PRODUCT,BL

;decrement  Al
DEC AL

JMP WHILE_

ENDWHILE_:

   ;display message
    LEA DX,MSG2
    MOV AH,9
    INT 21H

;to convert the digit from  real value to ASCII value
    ADD PRODUCT , 30H
;display result
    MOV AH,2
    MOV DL,PRODUCT
    INT 21H

    RET
```

seventh Lab (practice on logic operations)

Q : Write a program that reads a binary number, stores it in BX and display it's one's complement:

Sample execution:

```
Enter the binary number (max 16-bit)     :1100111000101010
The given binary number one's complement is:0011000111010101
```

Input

Solution:

```
 org 100h

.DATA
  PROMPT_1  DB  'Enter the binary number (max 16-bit) : $'
  PROMPT_2  DB  0DH,0AH,' The given binary number ones complement is:$'

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

    LEA DX, PROMPT_1        ; load and display PROMPT_1
    MOV AH, 9
    INT 21H

    XOR BX, BX              ; clear BX
    MOV CX, 16              ; initialize loop counter
    MOV AH, 1               ; set input function

    INPUT:                  ; jump label
      INT 21H               ; read a digit
      CMP AL, 0DH           ; compare digit with carriage return
      JE  END_               ; jump to label END if carriage return
      AND AL, 0FH           ; convert ascii to decimal code
      SHL BX, 1             ; rotate BX to left by 1 bit

OR BL, AL             ; set the LSB of BX with input
    LOOP INPUT              ; jump to label INPUT

    END_:                    ;  jump label
NOT BX                                ;1's complement of BX
    LEA DX, PROMPT_2         ; load and display PROMPT_2
    MOV AH, 9
    INT 21H

    MOV CX, 16               ; initialize loop counter
    MOV AH, 2                ; set output function
```

```
OUTPUT:                 ; jump label
  SHL BX, 1             ; shift left BX by 1 bit

  JNC ZERO              ; jump to label ZERO if CF=0
    MOV DL, 31H         ; set DL=1
    JMP  DISPLAY        ; jump to label DISPLAY

  ZERO:                 ; jump label
    MOV DL, 30H         ; set DL=0

  DISPLAY:              ; jump label
    INT 21H             ; display digit
LOOP OUTPUT             ; jump to label OUTPUT

ret
```

eighth Lab :

Q : Write a program to convert the high level language assignment statement to assembly code:

$$((A * B) + 4) / 2$$

Sample execution:

```
PLEASE ENTER THE THE VALUE OF A :2 3
THE RESULT OF ((A*B+4)/2) IS : 5
```

Input

Solution:

```
org 100h


.DATA
MSG DB 'PLEASE ENTER THE THE VALUE OF A and B : $'
MSG2 DB 0DH,0AH,' THE RESULT OF ((A*B+4)/2) IS :$'
b db ?
a  db ?

.CODE
MOV AX, @DATA       ; initialize DS
MOV DS, AX
MOV AH,9            ; display the string MSG
LEA DX, MSG
INT 21h

;input a number and move it to variable  A

     mov ah,1
     int 21h
     mov A,al

;input a number and move it to variable  B

      mov ah,1
     int 21h
     mov b,al


      ;to convert the digit from ASCII value to real value

      sub a,30h
      sub b,30h

      mov al,A

         mul B      ; = Al*B
         add al,4   ; = Al + 4
```

```
          mov cl,2    ; Al /2
         div cl

 ;to convert the digit from ASCII value to real value
        add al,30h

;display result

         mov ah,2
         mov dl,al
         int 21h

      ret
```

Final Lab :

Q: Write a program that Sum the Values of odd index of array A .

Suppose that Array A contains :

A : 2,3,5,8,1,4

Sample execution:

```
THE SUM OF ARRAY ELEMENTS: 15
```

Solution:

```
org 100H
.DATA
MSG DB 'THE SUM OF ARRAY ELEMENTS: $'
A DB 2,3,5,8,1,4

.CODE
MOV AX, @DATA ; initialize DS
MOV DS, AX

XOR SI,SI     ;clear SI
XOR AX,AX     ;AX holds sum
Mov SI,1      ;to point in  index one
MOV CX,3      ; no. of even indexes in A

TOP:
ADD AL,A[SI]    ;Sum= Sum+element
ADD SI,2       ; to go to even index
LOOP TOP        ;loop until done

PUSH AX        ;save Sum (save the value in AX)



MOV AH,9 ; display the string MSG
LEA DX, MSG
INT 21h

POP AX         ;retrive number
CALL OUTDEC

; exit to DOS
ret
INCLUDE C:\ASM\PGM9_1.ASM ;include outdec
```