Name:- Ubaid ali

subject:- Assembly language

Id No:- 6840

Instructur:- Sir amin

Date:- 29-6-2020

--------*---*

Q 8 Writ a complete Program using the Loop instruction with indisect addresing that copies a string from source to toget-

source BYTE "This is the source string", 0

target BYTE SIZEOF source DUP ('#')

insert the following

statement immediately before and after the loop to display the hexadecimal contents of the source.

ANS

Part (a)

Q1 write a sequence of statments

ANS Push ebx

2 that use only push and pop

ANS Push eax

3 instruction te exchange the

ANS Pop ebx

4 wait in the EAX and

Page 3

EBX registers

Ans Pop eax

Part "B"

Before the Loop:

```
mov esi, OFFSET source
                ; offest of variable
mov ebx, 1      ; byte formate
mov ecx, SIZEOF source ; counter
call DumpMem ; require the internal
                values of esi, ebx
                ecx
```

After the Loop

```
mov esi, OFFSET target ; offest of
                variable
mov ebx, 1 ;  byte format
move ecx sizeOF target ; couted
call DumpMem ; seruise the
                inital values of
```

Hints:

Copy a string backwards. Dumpmem usage: cal 6th if your program word correctly - it will display the following sequence of hexadecimal bytes.

DUMP of offset 00404000

54 68 69 73 20 69 73 20 74 68 65 20
73 6F 75 72
63 65 20 73 74 72 69 6E 67
This is the source string
DUMP of offset 0040401A

67 6E 69 72 74 73 20 65 63 72 75 6F 73 20
65 68
74 20 73 69 20 73 69 68 54
gnirts ecuos eht si siht

Part "C"

```
INCLUDE Irvine32.inc

.data
    count   DWORD ?

.code
    main    PROC
mov   eax, 0 + (0 * 16)
mov   ecx, 16

L1:
    mov   count, ecx
    Push  eax
    mov   ecx, 16

L2:
    call settext color
    Push  eax
    mov   al, 'H'
```

6

```
call    write char
pop     eax
inc     eax

loop L2

        call crlf
        pop  eax
        add  eax, 16
        mov  ecx, count

call crlf
call waiting

exit
        main ENDP

END main
```

7

Q # 3

value of EAX of the followings.

ANS
```
mov eax, TYPE myBytes ; a . 1
mov eax, LENGTHOF myBytes ; b . 4
mov eax, LENGTHOF myBytes ; c . 4
mov eax, TYPE myword ; d . 2
mov eax, LENGTHOF myword ; e . 4
mov eax, SIZEOF myword ; f . 8
mov eax, SIZEOF mystring ; g 5
```

Q # 6

write assembly language code of followings

Ans **a :-**

```
mov al, 'a' ; al = 01100001b
```

or and
al, 00110000b ; al = 00110110b

(b)

mov al, b ; al = 00000110b

or
al, 00110000b ; al = 00110110b

(c)

and al, 00000111b ; clear unwte bits
cmp al, 00000111b ; check remaing bits
je L1 ; all set? Jump to L1

Q # 5)

Read what will be the value of the destination operand each of the

9

following instructions execute in squence?

- data

my Byte BYTE 6th, 2th
my BYTE wood 8Ath 3Bh
- - - - - - - -

- code - - - - - - - -

ANSWER :-

AL : 40h

AX : AX 0038h

EDX : 3

Q # 17

What will be the value of the destination operand after each of the following instruction

- data

```
val1   WORD 1000h
val2   WORD 2000h
array B  BYTE 10th, 20th
  30th  40th  50th
```

- code

```
MOVZX CX, bl ; (a) CX = ?
MOVSX CX, bl ; (b) CX = ?
mov   ax, val1
mov   eax (array) ; (f) EAX = ?
```

Answer:-

```
MOVZX CX, bl    CX = 009Bh
MOVSX CX, bl    CX = 009Bh
xchg  val2, ax  Val2 = 1000h
mov al, [arrayB+1] ; (d) AL = 20h
mov ax[array w+2] ; (e) Ax = 200h
mov eax, [array D+4] ; (f) EAX =
                        20000h
```

11

Q # 2

write down the values of destination operands and flags after the execution of each instruction.

Code:

```
mov cx, 1
Sub cx, 1   ; (a) cx=? ZF=?
mov cx, 0
Sub cx, 1   ; (b) cx=? SF=?
mov al, 0FFh
add al, 1   ; (c) AL=? CF=?
mov al, 0
Sub al, 1   ; (d) AL=? CF=?
mov al 7Fh
add al, 1   ; (e) AL=? OF=?
```

(12)

Answer:-

```
Mov cx, 1
Sub cx, 1 ; (a) cx=0 ZF=1

Mov cx, 0
Sub cx, 1 ; (b) cx=-1 SF=1

mov al, 0ffh
add al, 1 ; (c) AL=00 CF=1

mov al, 0
Sub al, 1 ; (d) AL= FF CF=1

mov al, 7Fh
add al, 1 ; (e) AL=80 OF=1

mov al, -128
neg al ; (f) CF=1 OF=1
```

Q # 7

write each of the following Pseudocode in assembly language and explain:

a) if (val1 <= val2)
    val3 = 128;
  else
  {
    val3 = 110;
    val4 = 90;
  }

b) if (val1 > ecx) or (ecx > ebx)
    x = 30
  else
    x = 40

c) while (eax < ebx)
    eax = eax + 1;

Answer :-

a)

```
mov eax , var 1
cmp eax , var 2

    jle L 1
mov var3 , 10
mov var4 , 80
 Jmp L2

L1 :-
    mov var 8 12 B
```

c)

```
top : cmp eax, ebx ; check up condition
Jae menut ; false ? exit loop
Jnc eax ; body of loop
```

ans

```
jnp top ; repeat the
          loop
next:
```

b/

```
cmp val1, ecx
jna L1

cmp ecx, edx
jna L1

MOV x, 30
jmp next

L1: mov x, 40

next:
```

16

Q # 6

write down the value of each destination operand:

```
.data
Val32 LABEL DWORD
varB BYTE 78h, 86h, 34h, 12th
Val8 LABEL BYTE
varD DWORD 12348878h

.code
mov bl, BYTE PTR varD
                ; (a) BL = ?
mov eax, DWORD ; (b) EAX = ?
mov al varB ; (c) AL = ?
mov eax, Val32 ; (d) EAX = ?
```

Answer :-

```
val32 LABEL DWORD
valB BYTE 78h, 56h, 34h, 12h
val8 LABEL BYTE
valD DWORD 12345678h

.code
mov bl, BYTE PTR valD ; (a) BL= 78h
mov eax, DWORD PTR valB ; (b) EAX =785634 12h
mov eax, val8 ; (c) AL=78h
mov eax, val32 ; (d) Eax = 12345678h
```