

NAME SAEEDA NAZ

I.D 14332

Semester 5<sup>th</sup> semester

Department BS (CS)

DATE 29 Mon / Jun / 2020

Subject Assembly Language

Question No 5

Ans :-

Al : 40h

Ax : Ax 003Bh

EDX : 3

Question No (8)

Part No A :-

Ans ~~to~~ A sequence of Statement are  
push ebx ; Assume EBX = x  
and EAX = y, here the content  
of EBX (i.e x) is pushed  
eax, which is assumed to  
be y. pop ebx ; y from stack  
is assigned to EBX, therefore  
EBX = y

pop eax ; x from stack is  
assigned to EAX therefore  
EAX = x



Page 2

push ebx

push eax

pop ebx

pop eax

~~Part~~ No B :-

Ans :- and al, 1111100b  
jz L3

imp L4

OR X

## Part B :-

## Solution :-

- 386

- model Flat . Std call

- Stack 4096

Exit Process PROTO, dw Exit code:  
DWORD

- data

Source BYTE " This is the Source  
String" , 0

target 0 BYTE SIZE OF Source Dup(x)

- code

main PROC

mov esi, 0

mov edi, LENGTHOF Source - 1

mov ecx, SIZE OF Source

L1 :-

mov eax, 0

mov al, Source [ esi ]

mov target [ edi ], al

inc esi

dec edi

loop L1



INVOKE Exit Process, 0

main ENDP

END main

OR

Before

~~After~~ the loop :-

```
mov esi, OFFSET source ; offset of variable
mov ebx, 1 ; byte format
mov ecx, SIZEOF source
```

```
call DumpMem ; require the initial
of esi, ebx, ecx.
```

After the loop :-

```
mov esi, OFFSET target ; offset of variable
mov ebx, 1 ; byte format
mov ecx, SIZEOF target ; counter
call DumpMem ; require the initial
value of esi,
ebx, ecx.
```

Part (c)

Ans :-

```
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 SetTextColor  
push eax  
mov al, 'H'  
call WriteChar  
pop eax  
inc eax
```

```
loop L2
```



Page 6

```
call crif  
pop  eax
```

```
add  eax, 16  
mov  ecx, count
```

Loop L1

```
call crif  
call wait Msg  
exit.
```

main ENDP

END main.

Question No (3)

- A) EAX = 1
- B) EAX = 4
- C) EAX = 4
- D) EAX = 2
- E) EAX = 4
- F) EAX = 8
- G) EAX = 5

## Question No (6)

part No C :-

Ans Solution: clear all bits except bit 0, 1 and 3. Then compare the result with 00001011 binary.

and al, 00001011b ; clear unwanted bits

cmp al, 00001011b ; check remaining bits

je L1 ; all set? jump to L1.

part No A :-

Ans :-

## • DATA

name\_prompt db "please type your name." 0  
out\_msg db "your name" in (capital: "0,

## • UDATA

in\_name resb 3)

## • CODE

## • STARTUP

putstr name\_prompt ; request  
character string



Page 8

```
GetStr in-name, 31 ; read input character  
String.
```

```
PutStr out-msg  
mov EBX, in-name ; EBX = pointer to  
in-name
```

Process-char :

```
mov AL, [EBX] ; move the char to AL  
cmp AL, 0 ; if it is the NULL  
character.
```

```
je done ; conversion done
```

```
cmp AL, 'a' ; if (char < 'a')
```

```
jll not_lower_case ; not a lowercase  
letter
```

```
cmp AL, 'z' ; if (char > 'z')
```

```
jg not_lower_case ; not a lowercase  
letter
```

lower case

```
add AL, 'A' - 'a' ; convert to uppercase
```

not\_lower\_case :

```
putch AL ; write the character  
inc EBX ; EBX points to the  
next char.
```

done :

Page 9

rwln

• Exit

(OR)

mov al, 'a' ; al = 01100001b  
cmd al, 11011111b ; al = 0100001b

part No (b) :-

Ans :- mov al, 6 ; al = 00000110b  
or al, 00110000b ; al = 00110110b

Question No (2)

Ans :- ~~Movzx cx, bl cx = 009Bh~~

~~Movsx cx~~

Ans Movzx cx, bl cx = 009Bh  
Movsx cx, bl cx = 009Bh

Xchg val2, ax val2 = 1000h  
mov val, [arrayB+1] ; (d) AL = 20h

mov ax, [arrayW+2] ; (e) AX = 200h



Page 10

mov eax, [array D+4]; (F) EAX = 20000h.

### Question No (2)

Ans :-

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, 7Fh

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

mov al, -128

neg al; (F) CF = 1 OF = 1.

### Question No (7)

Ans :-

Part (a)

mov eax, var1

cmp eax, var2

Page 11

jge L1

mov var 3, 110

mov var 4, 90

jmp L2

L1 : mov var 3, 128

Part No C :-

Ans top : cmp eax, ebx ; check loop condition

jge next ; false, exit loop

jnc eax ; body of loop

jmp top ; repeat the loop

next :

Part No b :-

Ans :-

cmp val 1, ecx

jna L1

cmp ecx, edx



Page 12

jna LI

mov X, 30

jmp next

LI : mov x, 40

next

### Question No (4)

• data

Ans Val 32 LABEL DWORD  
Val B BYTE 78h, 56h, 34h, 12h  
Val 8 LABEL BYTE  
Var D DWORD 12345678h

• code

mov ~~eax~~ bl, BYTE PTR varD ;  
(a) BL = 78h

mov eax, DWORD PTR varB ;  
(b) EAX = 78563412h

mov al, val 8 ; (c) AL = 78h

mov eax, val 32 ; (d) EAX = 12345678h