

IQRA NATIONAL UNIVERSITY

PESHAWAR

SUBMITTED BY SHARIQ

ID

13698

SUBJECT

ASSEMBLY

LAUNGATUGE

### Question: 1

Answer:

(a)

```
mov bx, 0A69Bh
movzx eax, bx; EAX = 0000A69Bh
movzx edx, bl; EDX = 0000009Bh
movzx cx, bl; CX = 009Bh
```

(b)

```
mov bx, 0A69Bh
movsx eax, bx; EAX = FFFFA69Bh
movsx edx, bl; EDX = FFFFFFF9Bh
movsx cx, bl; CX = FF9Bh
```

(d)

```
mov al, [arrayB+1]; AL = 20h
```

(e)

```
mov ax, [arrayW+2]; AX = 200h
```

(f)

```
mov eax, [arrayD+4]; EAX = 20000h
```

### Question: 3

Answer

```
mov eax, TYPE myBytes; a. EAX 1
mov eax, LENGTHOF myBytes; EAX 4
mov eax, SIZEOF myBytes; EAX 4
mov eax, TYPE my words; EAX 2
mov eax, LENGTHOF my words; EAX 4
mov eax, SIZEOF my words; EAX 8
mov eax, SIZEOF my string; EAX 5
```

### Question: 5

Answer

(a)

```
mov al, [esi + 3]; AL = 40h
```

(b)

```
mov esi, OFFSET my words + 2
mov ax, [esi]; AX = 003Bh
```



(c)  
move edi, 8  
move edx, myDoubles[edi]; EDX=3

Question: 6

Answer

(a)

MODEL SMALL  
STACK 100H

DATA

PROMPT-1 DB 'Enter the lower case letter:'  
PROMPT-2 DB 'The upper case letter is: \$!'

CODE

MAIN PROC

MOV AX, @DATA

; initialize DS

MOV DS, AX

LEADX, PROMPT-1

; load and print

PROMPT-1

MOV AH, 9

INT 21H

; read a letter

MOV BL, AL

; Save the letter in BL

MOV AH, 2

MOV DL, 0DH

; return carriage

INT 21H

MOV DL, 0AH

; line feed

INT 21H

LEADX, PROMPT-2

; load and print

PROMPT-2

MOV AH, 9

INT 21H

SUB BL, 20H  
to upper case letter

; convert a lower case  
letter

MOV AH, 2

; print the upper case  
letter

MOV DL, BL

INT 21H

MOV AH, 4CH

; return control to Dos

INT 21H

MAIN ENDP

END MAIN

(c)

Clear all bits except bits 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

Question 7.

(a)

Answer

mov eax, var1

cmp eax, var2

jle L1

mov var3, 110

mov var4, 90

jmp L2

L1: mov var3, 128

L2:

(b)

Answer

cmp val1, ecx

ja L1

cmp ecx, edx

ja L1

move x, 40

L1: move x, 30

next.

(c)

Answer

- while

cmp eax, ebx; check loop condition  
cmp eax, ebx;

check loop condition

jae - endwhile; false? exit loop

inc eax; body of loop

jmp - while; repeat the loop

endwhile



Q4:- Write down the value of each destination operand.

Ans:

• Data

Val 32 Label DWORD

Var B Byte 78h, 56h, 34h, 12h

Val 8 Label Byte

Var D Dword 12345678h

• code

mov bl, BYTE PTR var D; BL = 78h

mov eax, DWORD PTR var B; EAX =

mov al, val 8; AL =

mov eax, val 32; EAX = 02h

Q8 :-

Answers:-

(a).

Push ebx

Push eax

Pop ebx

pop eax

(b)

• 386

• model flat stdcall

• Stack 4096

ExitProcess PROTO, dwExitCode: DWORD

• data

• source BYTE "This is the source string", 0

target Byte SIZEOF source DUP('#')

• Code

main PROC

mov esi, 0

mov edi LENGTHOF source - 1

```
mov ecx, SIZEOF SOURCE
```

```
main PROC
```

```
mov esi, 0
```

```
mov edi, LENGTHOF source - 1
```

```
mov ecx, SIZEOF SOURCE.
```

```
L1:
```

```
mov eax, 0
```

```
mov al, source [esi]
```

```
mov target [edi], al
```

```
inc esi
```

```
dec edi
```

```
loop L1
```

```
Invoke Exitprocess, 0
```

```
main ENDP
```

```
END main.
```



① :- INCLUDE Irvine32.inc

• data

count DWORD ?

• Code

main PROC

mov eax, 0 + (0x16)

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

call crif

pop eax

add eax, 16

mov ecx, count

Loop L1

call crif

Call waitmsg

exit

main ENDP

END main.