

Name :- FAYYAZ MUHAMMAD

ID :- 14294

SUBJECT :- MICROPROCESSOR AND ASSEMBLY  
LANGUAGE.

DATE :- 29/06/2020.

Department :- BS(CS). 5<sup>th</sup> Semester

Q No # 01

What will be the value of the destination operand after each of the following instructions execute in sequence?

- $CX = 009Bh$
- $CX = 009Bh$
- $val2 = 1000h$
- $AL = 20h$
- $AX = 200h$
- $EAX = 20000h$

Q No # 02

Write down the values of destination operands and flags after the execution of each instructions.

- $CX = 0$        $ZF = 1$
- $CX = -1$        $SF = 1$
- $AL = 00$        $CF = 1$
- $AL = FF$        $CF = 1$
- $AL = 80h$        $DF = 1$

Q No # 3

What will be the value of EAX after each of the following instructions execute?

Solution:-

mov eax, TYPE myBytes ;(a) EAX = 1

mov eax, LENGTHOF myBytes ;(b) EAX = 4

mov eax, SIZEOF myBytes ;(c) EAX = 4

mov eax, TYPE myWords ;(d) EAX = 2

mov eax, LENGTHOF myWords ;(e) EAX = 4

mov eax, SIZEOF myWords ;(f) EAX = 8

mov eax, SIZEOF myString ;(g) EAX = 5

Q No #4

Write down the value of each destination operand:

VarA LABEL DWORD

VarB BYTE 78h, 56h, 34h, 12h

VarC LABEL BYTE

VarD DWORD 12345678h

• Code

mov bl, BYTE PTR VarD ;(a) BL = 78h

mov eax, DWORD PTR varB ; (b) EAX = 7883412h  
 mov al, val8 ; (c) AL = 78h  
 mov eax, val32 ; (d) EAX = 12345678h

Q No # 5.

What will be the value of the destinations operand after each of the following instructions execute in sequence?

• data

my Bytes	BYTE	10h, 20h, 30h, 40h
my Words	WORD	8Ah, 3Bh, 72h, 44h, 66h
my Doubles	DWORD	1, 2, 3, 4, 5

• Code

```

mov esi, OFFSET my Bytes
mov al, [esi+3] ; (a) AL = 40h
mov esi, OFFSET myWords + 2
mov ax, [esi] ; (b) AX = 0038h
mov edi, 8
mov edx, myDoubles [edi] ; (c) EDX = 3
  
```

Q NO # 06

Write assembly language code for each of the following:

a) Convert the character in AL to upper case.

Solution:

Use the AND instruction to clear bit 5.

```
mov al, 'a'           ; AL = 01100001b
and al, 11011111b    ; AL = 01000001b
```

b) Convert a binary decimal byte into its equivalent ASCII decimal digit.

Solution:

Use the OR instruction to set bits 4 and 5.

```
mov al, b             ; AL = 00000110b
or  al, 00110000b    ; AL = 00110110b
```

c) Jump to label L1 if bits 0, 1, and 3 in AL are all set.

Solution: 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
```

Q No # 7

Write each of the following pseudocode in assembly language and explain:

```
a) if (var1 <= var2)
    var3 = 128;
else
{
    var3 = 110;
    var4 = 90;
}
```

Q NO # (8)

a) Write a sequence of statements that use only PUSH and POP instructions to exchange the values in the EAX and EBX registers.

Ans) PUSH ebx  
 PUSH eax  
 PUSH ebx  
 PUSH eax

b) Write a program with a loop and indirect addressing that copies a string from source to target, reversing the character order in the process. Use the following variables:

Source BYTE "This is the source string", 0  
 target BYTE SIZEOF Source DOP('#')

Ans) Solution:

.386  
 .model flat, Stdcall  
 .Stack 4096  
 Exit procen PROTO, dw Exit code: DWORD

• data

Source Byte "This is the source string", C  
 Target BYTE SIZEOF Source DOP

• Code

```
main PROC
mov esi, 0
mov edi, LENGTHOF Source - 1
mov ecx, SIZEOF Source
```

L1:

```
mov ebx, 0
mov al, Source [esi]
mov target [edi], al
inc esi
dec edi
loop L1
```

```
INVOKE Exit Process, 0
main EOP
END main
```

c) Write a program that displays a string in all possible combinations of foreground and background colours (16x16=256). The colors are numbered from 0 to 15. So you can use a nested loop to generate all possible combinations. Also use a delay of 1s in each foreground color change.



INCLUDE Irvine32.inc

.Data

loopcount DWORD?  
foreground DWORD?  
background DWORD?

.Code

main PROC

mov ecx 16; Outer loop count

L1:

mov loopcount.ecx

mov foreground.ecx

dec foreground; Foreground initial value = 15 by  
1 each time loop repeats

mov ecx, 16; Inner loop count

L2:

mov background.ecx

dec background; Background initial value = 15  
decrements by 1 each time  
loop repeats

mov eax, background; Set EAX = background

shl eax, 4; Shift left, equivalent to multiplying  
EAX by 16

add eax, foreground; Add foreground to EAX

call Settextcolor

mov al, "A"; Set AL to character to be written  
to screen.

loop L2  
call crlf; Advances cursor to beginning of next line.

mov ecx, loopcount  
loop L1

exit

main ENOP

END main