

Name: Noor Saleem

ID: 6811

Subject: Assembly Language

Department: BS(CS)

Submitted To:

SIR Muhammad Amin

Date: 26-09-2020

ON02

write down the value of the carry, sign, zero and overflow flags after each instruction executed.

mov ax, 7FF0h

add cx, 10h ; (a) CF=1 SF=0 ZF=1 OF=0

add dx, 1 ; (b) CF=0 SF=1 ZF=0 OF=1

add ax, 2 ; (c) CF=0 SF=1 ZF=0 OF=0

mov dx, 1

sub dx, 2 ; (d) CF=1 SF=0 ZF=1 OF=0

QND3:

Use The Following data definition for the coming question.

(i) What will be the value of EAX after each of the following instruction execute

(Ans)

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

mov eax, LENGTHOF listB; (b) EAX = 6

mov eax, SIZEOF listB; (c) EAX = 6

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

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

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

mov eax, SIZEOF string1; (g) EAX = 17

QNO4 Use the following
Ans data definition
for coming question.

ListB BYTE 10h, 20h, 30h, 40h

ListW WORD 8Ah, 3Bh, 72h, 44h, 66h

ListD DWORD 1, 2, 3, 4, 5

Pointer1 DWORD ListD

What will be the value
of destination operand
after each of the ---

mov esi, OFFSET ListB

mov al, [esi] ; (a) AL = 10h

mov al, [esi+3] ; (b) AL = 40h

mov esi, OFFSET ListW+2

mov ax, [esi] ; (c) AX = 003Bh

mov edi, 8

mov edx, [ListD+edi] ; (d) EDX = 3

mov edx, @ListD[edi] ; (e) EDX = 3

mov ebx, Pointer1

mov eax, [ebx+4] ; (f) EAX = 2

Q NOS:

(i)

```
if (var1 <= var2)
```

```
    var3 = 15
```

```
else
```

```
{
```

```
    var3 = 10;
```

```
    var4 = 30;
```

```
}
```

Assembly Language code.

```
move eax, var1
```

```
cmp eax, var2
```

```
Jle L1
```

```
mov var3, 10
```

```
mov var4, 30
```

Jmp L2:

```
L1: mov var3, 10
```

```
L2: mov var4, 30
```

(ii) if (vall > ecx) AND (ecx > edx)
Then

A = 12

else

B = 6

((Ans))

cmp vall, ecx

Jna else

cmp ecx, edx

Jna else

mov A, 12

Jmp next

else: mov B, 6

next:

(iii) While (ebx < eax)

ebx = ebx + 1

Ans

cmp ~~ebx~~, ebx, eax

jae next

inc ebx

Jmp top

next:

(QNO6)

(i) What will be the final value of eax in this example:

```
mov eax, 0
```

```
mov ecx, 10; out loop counter.
```

```
L1: mov eax, 3.
```

```
mov ecx, 5 inner loop count
```

```
L2: add eax, 5
```

Loop L2; repeat inner loop

Loop L1; repeat outer loop

(Ans) The loop is infinite loop so value of eax not possible.

QNO6: (ii)

- model flat, stdcall
- stack 4096

Exit process PROTO,

du Exit code: DWORD

• code

main PROC

mov eax, 3h

mov ebx, 8h

mov ecx, 1h

mov edx, 8h

```
add eax, ebx
add ecx, edx
sub eax, ecx
INVOKE ExitProcess, 0
main ENDP
END main.
```

QNO 6 (iv)

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 ecx

Loop L2

call crlf

pop eax

add ecx, 16

mov ecx, count

Loop L1

call crlf

call WaitMsg

exit

main ENDP

END main

QNO 6

(iii)

- model flat, stdcall
- stack 4096

Exit process PROTO, duExit code:

DWORD

- data

array WORD 0, 2, 5, 9, 10

newArray DWORD LENGTH OF

array DUP (?)

- code

~~array~~ main PROC

mov ecx, LENGTH OF array

mov esi, OFFSET array

mov edi, OFFSET new Array

L1:

MOV EAX, 0

MOV AX, [ESI]

MOV [EDI], EAX

Add esi, TYPE array

Add edi, TYPE new Array

I O O D I T

INVOKE Exit process 0

mdid ENID P

END mdid