

NAME Rafi ULLAH

ID 14283

Department Bs(cs)

5th Semester:

SUBJECT Microprocessor  
AND Assembly LANGUAGE.

Submitted To Sir AMIN

Assignment No:

6:

QUESTION # 1

ANS: and, ax, 00FFh

QUESTION # 2

ANS: 08 ax, 00F00h

QUESTION # 3

ANS: XOR eax, 0FFFFFFFh

QUESTION # 4

ANS: test eax, 1 : (low bit set if

eax is odd

QUESTION # 5

ANS: 08, al, 00100000b

QUESTION # 6

ANS: JA, JNBE, JAE, JNB, JB, JNAE,  
JBE, JNA.

QUESTION # 7:

ANS: JG, JNLE, JGE, JNL, JL, JNGE, JLE  
JNG.

Page # 2

Assignment: 6

QUESTION # 8

ANS: No, Because the `jb` is used with signed value and `8109h` is negative, and `26h` is positive)

QUESTION # 10

ANS: `BX = 006Bh`

QUESTION # 11

ANS: `BX = 092h`

QUESTION # 12

ANS `BX = 064BBh`

QUESTION # 13

ANS `BX = A857h`

QUESTION # 14

ANS: `EBX = BFAFF69Fh`

QUESTION # 15

ANS: `RBX = 0000000050509B64h`

QUESTION # 16

ANS: (a) 01101101

(b) 01001000

(c) 01101111 (d) 10100011

QUESTION # 17

ANS: (a) 85h

(b) 34h

(c) BFh

(d) AEh

QUESTION # 18

(a) CF=0, ZF=0, SF=0

(b) CF=0, ZF=0, SF=0

(c) CF=1, ZF=0, SF=1

QUESTION # 19

ANS J ECXZ

QUESTION # ~~20~~ 24

ANS: YES

QUESTION # 25

ANS YES (the unsigned representation of -42 is compared to 26)

QUESTION # 28

cmp dx, cx

jbc l1

QUESTION # 29

ANS: cmp cx, cx

jglz

QUESTION # 30

and eel, 11111000

JZ L3

jmp L4

QUESTION # 31

(a) CMP EBX, ECX.

JBE L1; if (ebx <= ecx)

CM P EBX, val1

JBE L1

MOV X, 1

JMP L2

L1: MOV X, 2; else, X = 2

(b) CMP EBX, ECX

JBE EBX, EDX

JBE L1

JMP L3; both true, go to L2

L1: CMP EDX, EAX

JBE L3; if (edx <= eax), go to L3

L2: MOV X, 1

Page # 5

Assignment No: 6

QUESTION # 21

Ans: ; 7FFh = 32767d

; 8000h = 32768d

; unsigned cmp, and  
the jump will be taken.

QUESTION # 22

Ans: ; 7FFFh = +32767d

; 8000h = -32768d

; unsigned cmp, and  
the jump will not be taken

QUESTION # 23

Ans: MOV AX, 7FFFh

MOV AX, 6401h

XOR, AH, AL

QUESTION # 26

Ans: AND AL, 00001111b

## QUESTION #27

ANS: • data

memval DWORD?

• code

mov al, BYTE PTR memval

xor al, BYTE PTR memval+1

xor al, BYTE PTR memval+2

xor al, BYTE PTR memval+3

## QUESTION #34

ANS: include Irvine32.inc

N = 10

• data

array SDWORD N DUP(-10, -8, -6, -4,  
-2, -1, 1, 3, 5, 7)

j DWORD?

k DWORD?

• code

main PROC

CALL closes

mov j, 0

mov k, 10

mov ESI, OFFSET array

mov ECX, N

Page # 7

Assignment No: 6

call summing Array Elements In Range  
call write Int

call eof

mov j, -10

mov k, 0

mov ESI, OFFSET array

mov ECX, N

call summing Array Elements In Range  
call write Int

call eof

call waitmsg

exit

main ENDP

summing Array Elements In Range PROC

push ecx

push eax

mov ecx, 0

l1:

mov ebx, [esi]



Perge#8

Assignment: 6

cmp ebx, j

jge true1

jmp next

true1:

cmp ebx, k

jle true2

jmp next

true2:

add eax, ebx

next:

add esi, 4

loop 11

pop esi

pop ecx

ret

Summing Array Elements In Range ENDP

END main

~~ANS~~ QUESTION # 35

ANS: Include Irvine32.inc

.data

byte1 BYTE 1111110b, 1101110b,

1000111b, 11001100b, 11001010b,

11001010b, 11001010b, 1100.

byte2 BYTE 11111110b, 11011111b,

10001110b, 11101001b, 11001100b,

11001011b, 11001010b, 11001010b,

1100

.code

main PROC

mov esi, OFFSET byte1

mov ecx, SIZEOF byte2

call PEcheck

call writeINT

mov esi, OFFSET byte2

mov ecx, SIZEOF byte2

call PFcheck

call writeINT

exit

main ENDP

PFcheck PROC

; ecx PF=1 TRUE PF=0 false

; esi, PCX

push esi

push ecx

sub ecx, 1

mov al, 0

xor al, 0

mov al, [esi]

L1;

inc esi

xor al, [esi]

mov bd [esi]

loop L1

jp LPE 1

mov LPE 1

Page # 11

Assignment: 6

```
mov ecx, 0
```

```
jmp LEND
```

```
LPF 1:
```

```
mov ecx, 1
```

```
LEND:
```

```
pop ecx
```

```
pop esi
```

```
ret
```

```
PF check ENDP
```

```
END main
```

(QUESTION # 32)

Ans: INCLUDE 32.inc

• data

N DWORD 10

A DWORD 9

B DWORD 8

• code

main PROC

Page #12

Assignment: 6

mov eax, N

mov ebx, A

mov ecx, B

Top:

cmp eax, 0

jbe Next

cmp eax, 3

jne L1

jmp L4

L1:

cmp eax, ebx

jbl3

ja L2

L2:

cmp eax, ecx

ja L3

jbl4

Top:

L3:

Sub eax, 2

Jmp Top

L4:

Sub eax, 1

Jmp top

Next

Invoke Exit process, 0

main endp

end main

## QUESTION # 33

Ans: INCLUDE Irvine32.inc

N = 10

data

array DWORD N DUP(?)

j DWORD ?

k DWORD ?

code

main proc

call edxscr

mov j, -10

Page # 14

Assignment: 6

mov k, 10

mov esi, OFFSET array

mov ecx, N

call Filling An Array

mov j, 100

mov k, 1000

mov esi, OFFSET array

mov ecx, N

call Filling An Array

call waitmsg

exit

main ENDP

Filling An Array PROC

push ecx

push esi

L1:

mov ecx, j

mov ebx, k

dec ebx

Range #15

Assignment No: 6

• sub ebx, eax; create range from  
0 to N

xchg, ebx, eax; randoms works with  
eax

call Random Range; generate random  
with range 0 to N

neg ebx; change sign of ebx

sub eax, ebx; sub from eax to

define range

call exit

call waitmsg

mov [esi], eax

add esi, 4

loop L1

pop esi

pop ecx

ret

Filling An Array ENDP

END main