:: ASSIGNMENT # 6 ::

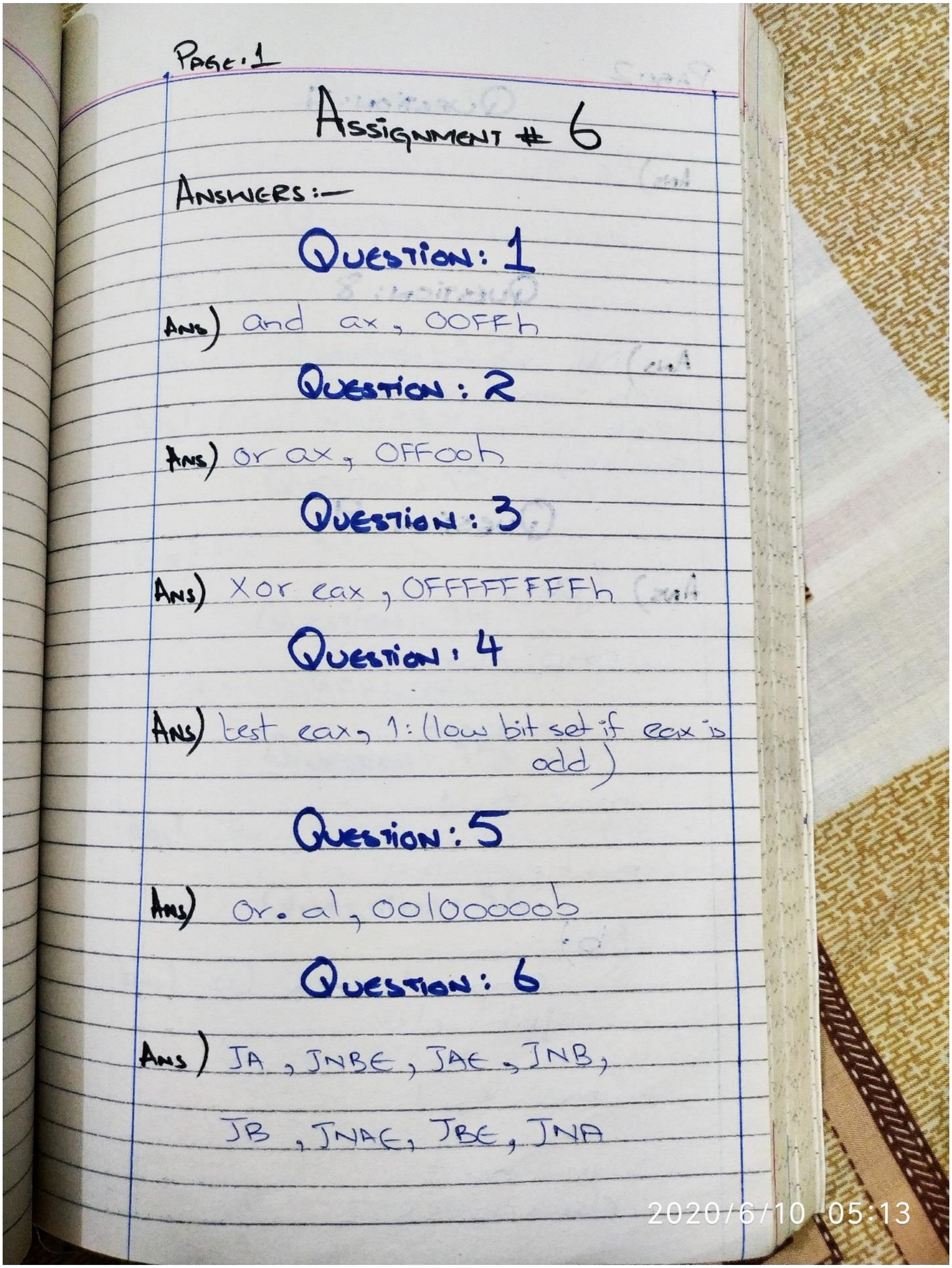
ID: 11533

Name: Ashir Ali Khan

Subject: Microprocessor and Assembly Language

Teacher: Sir Muhammad Amin

Iqra National University



Page.1

Assignment # 6

Answers:—

Question: 1

Ans) and ax, 00FFh

Question: 2

Ans) or ax, 0FF00h

Question: 3

Ans) Xor eax, 0FFFFFFFFh

Question: 4

Ans) test eax, 1: (low bit set if eax is odd)

Question: 5

Ans) or. al, 00100000b

Question: 6

Ans) JA, JNBE, JAE, JNB,

JB, JNAE, JBE, JNA

2020/6/10 05:13

Question: 7

Ans) JG, JNCE, JGE, JNL, JL,

JNGE, JLE, JNG

Question: 8

Ans) No, becaus, the jg is used with signed value and (809h is negative, and 26h is positive.)

Question: 9

Ans) (a:)

```
cmp ebx, eax
jna next
mov X, 1
next:
```

(b:)

```
    cmp edx, ecx
    jnbe else
mov x, 1
jmp next
else: mov X, 2
```

Page : 3

Question: 10

Ans) Bx = 006Bh

Question: 11

Ans) Bx = 092h

Question: 12

Ans) Bx = 064BBh.

Question: 13

Ans) Bx = A857h

Question: 14

EBx = BFAFF69Fh.

Question: 15

Ans) RBx = 0000000050509B64h

Question: 16

Ans) a) 0110101

b) 01001000

c) 0110111

d) 10100011

Question: 17

Ans) a) 85h

b) 34h

c) BFh

d) AEh

Question: 18

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

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

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

Question: 19

Ans) JECXZ

Question: 20

Ans) JA and JNBE jump to the destination if ZF = 0 and CF = 0

PAGE: 5

QUESTION: 21

ANS) ; 7FFh = 32767 d

; 8000h = 32768d

; unsigned Cmp, and

the jump will be taken

QUESTION: 22

ANS) ; 7FFFh = + 32767d

; 8000h = - 32768d

; unsigned Cmp, and

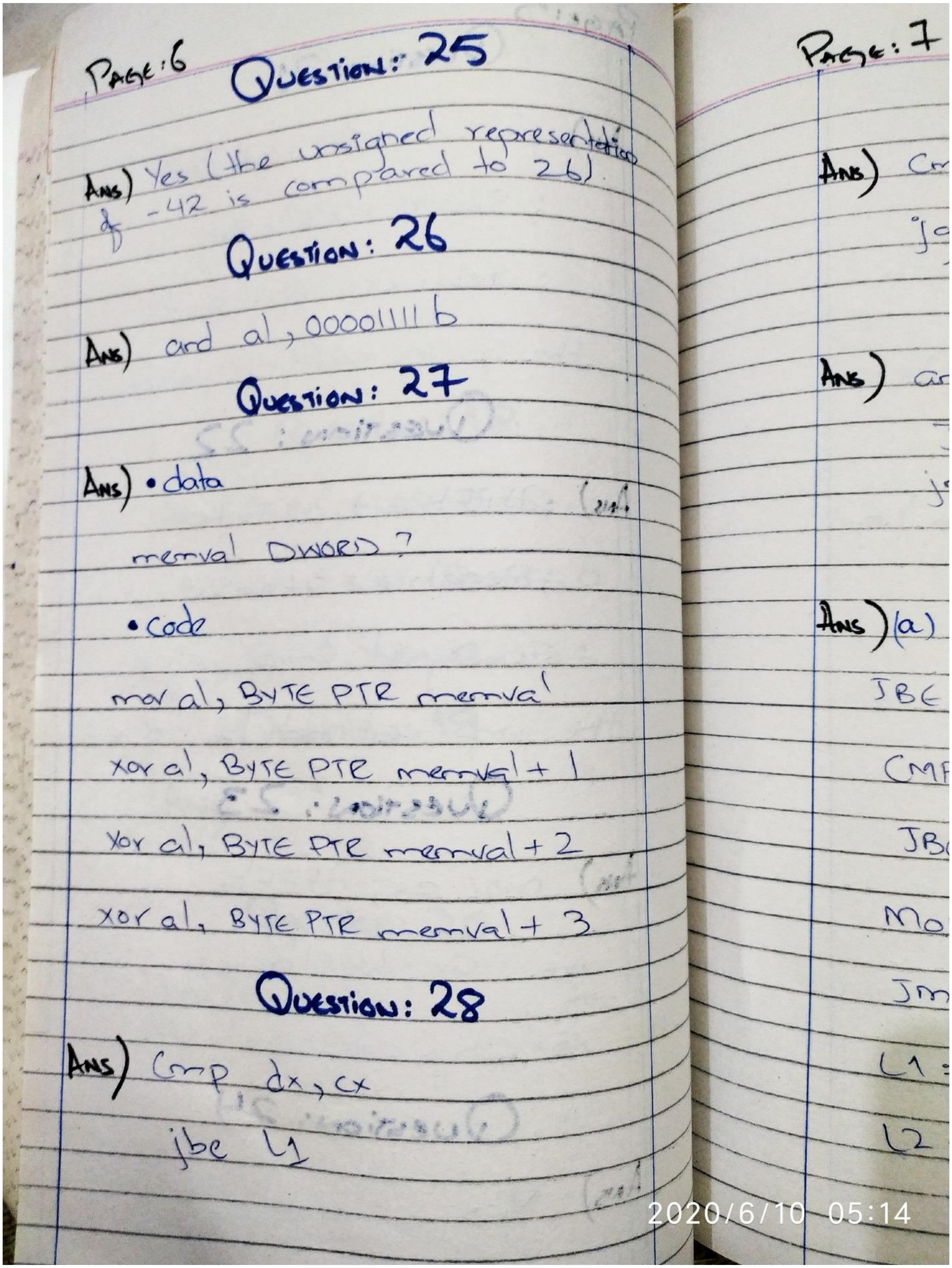
the jump will not be taken

QUESTION: 23

ANS)
```
mov ax, 7FFFh
mov ax, 64clh
xor ah , al
```

QUESTION: 24

ANS) Yes



PAGE:6

QUESTION: 25

Ans) Yes (the unsigned representation of -42 is compared to 26).

QUESTION: 26

Ans) and al, 00001111 b

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: 28

Ans)
```
Cmp dx, cx
jbe L1
```

PAGE: 7

Ans) Cm

jg

Ans) ar

j

j

Ans) (a)

JBE

CMF

JB

Mo

Jm

L1 :

L2

2020/6/10 05:14

Question: 29

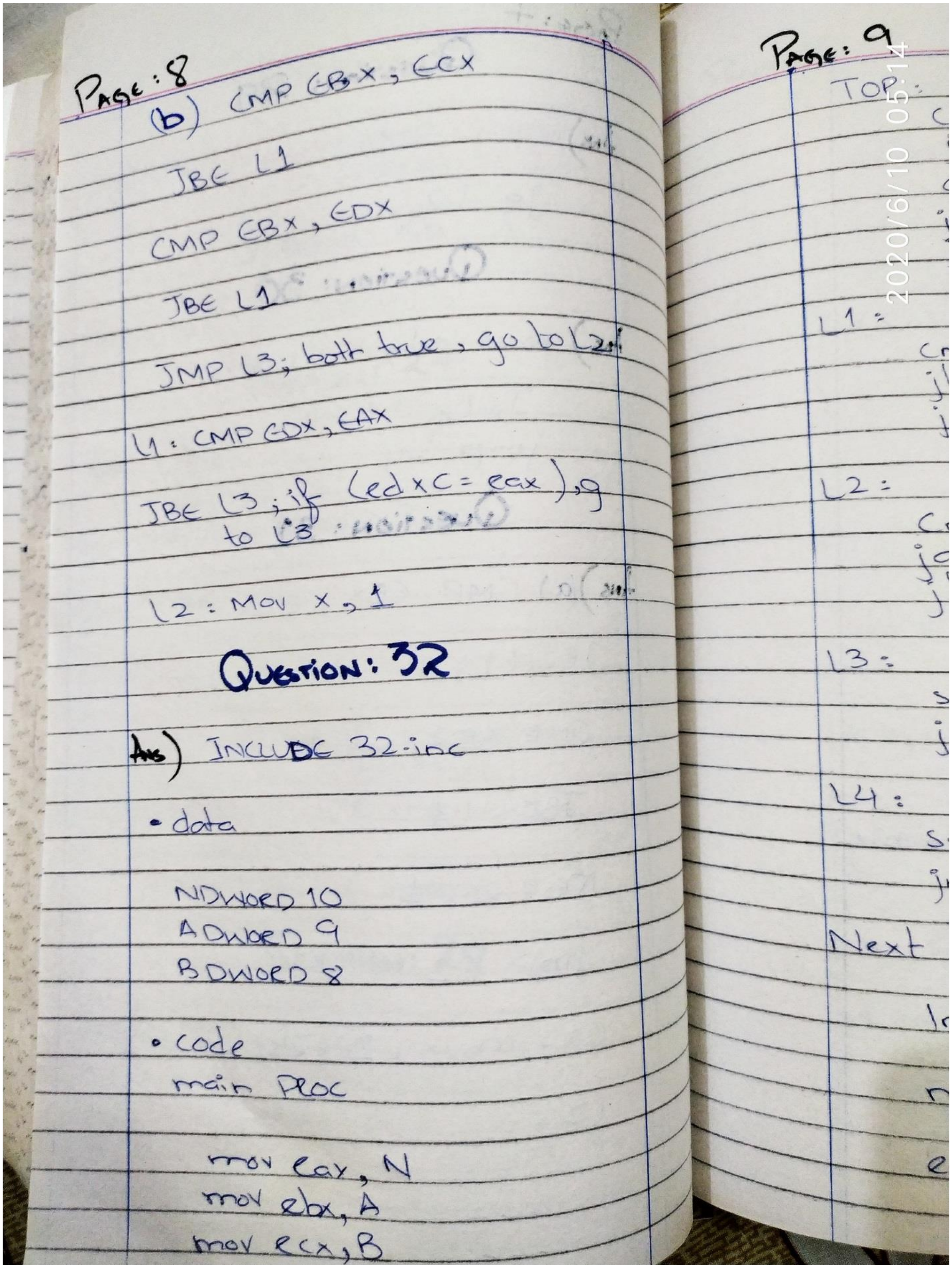
Ans)
```
cmp ax, cx
jg L2
```

Question: 30

Ans)
```
and al, 111111 00b
Jz L3
jmp L4
```

Question: 31

Ans) (a)
```
CMP EBX , ECX
JBE L1 ; if (ebx <= ecx)
CMP EBX, val 1
JBE L1
Mov x , 1
JMP L2
L1 : movx , 2 : else , x = 2
L2 :
```



Page: 8

```
(b) CMP EBX, ECX
JBE L1
CMP 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
```

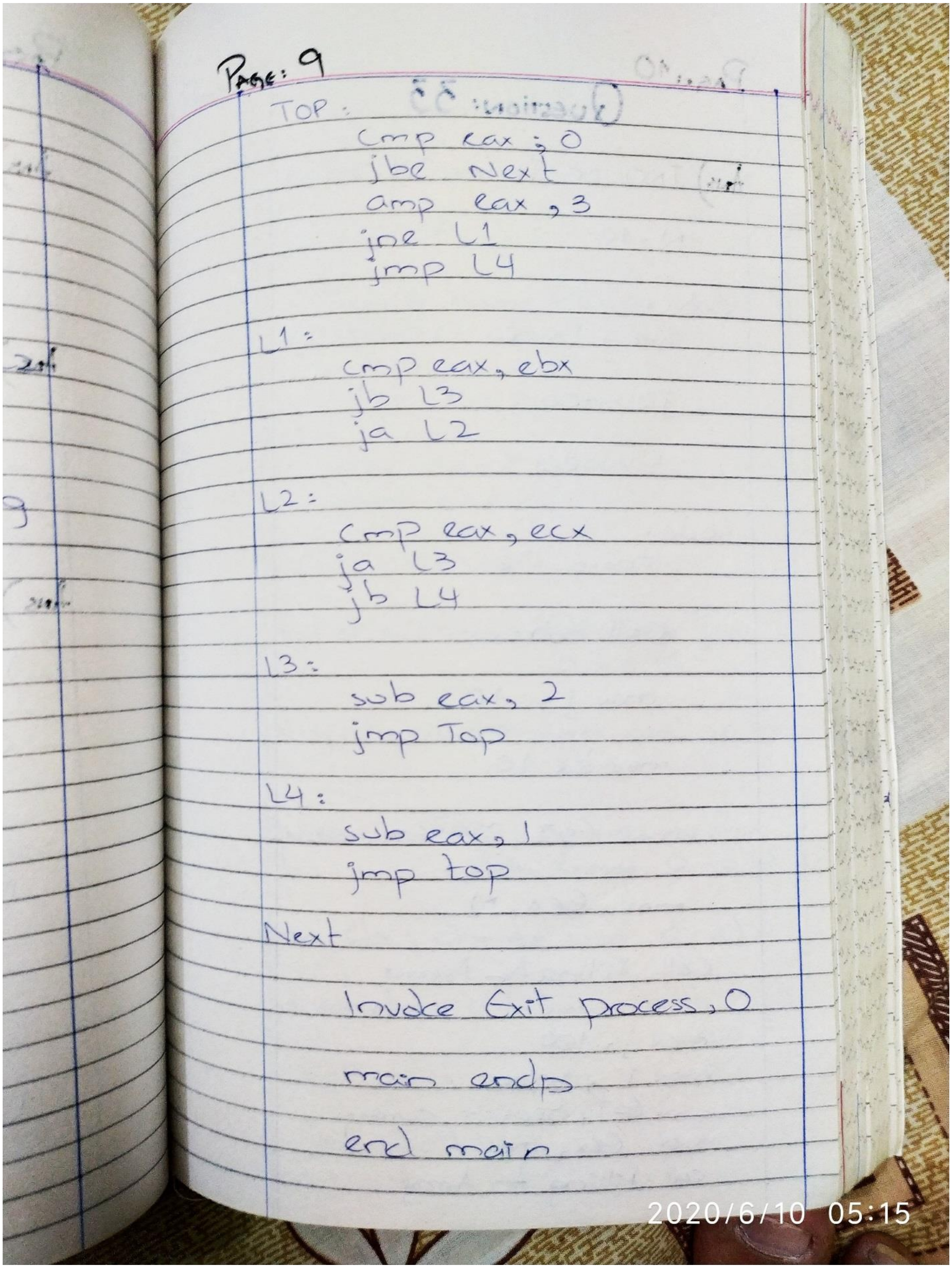
Question: 32

Ans) INCLUDE 32.inc

```
.data

NDWORD 10
ADWORD 9
BDWORD 8

.code
main PROC

   mov eax, N
   mov ebx, A
   mov ecx, B
```

Page: 9

```
TOP:

L1:

L2:

L3:

L4:

Next
```



Page: 9

```
TOP :
        cmp eax , 0
        jbe Next
        amp eax , 3
        jne L1
        jmp L4

L1 :
       cmp eax, ebx
       jb L3
       ja L2

L2 :
       cmp eax, ecx
       ja L3
       jb L4

L3 :
       sub eax, 2
       jmp Top

L4 :
       sub eax, 1
       jmp top

Next

       Invoke Exit process, 0

       main endp

       end main
```

2020/6/10 05:15

Page: 10 QUESTION: 33

```
Ans) INCLUDE Irvine 32.inc

     N=10

data:-
     array DWORD N DUP(?)

     jDWORD ?

     KWORD ?

Code.
       main PROC

       Call clrscr

       mov j, -10

       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
```

Page: 11

```
   Cal

exit

main

Fillin

   F
   P
L1:
   m
   m

dec

sub e

Xchg,e

Call Ran

neg e

sub eax,

   Call

   Call
```

2020/6/10 05:15

```
    call wait msg

exit

main ENDP

Filling An Array PRoc

    Pushecx
    Push esi
L1:
    mov eax, j
    mov ebx, k

dec ebx

sub ebx, ecx ; create range from
                     0 to N

Xchg, ebx, eax ; randoms work 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 Crlf

Call Wait msg
```

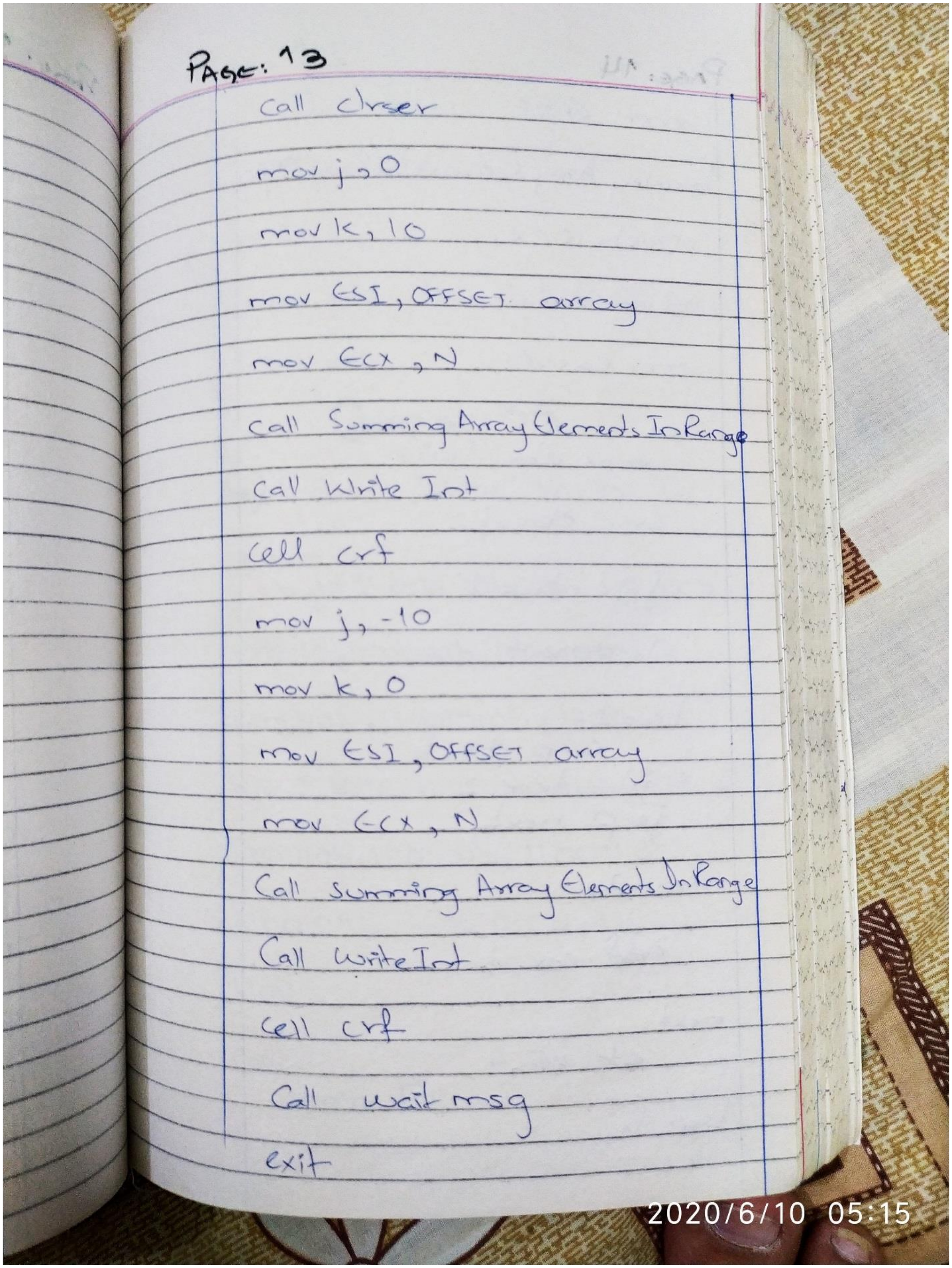
2020/6/10 05:15

```
mov [esi], eax
add esi, 4
loop L1
 pop esi
  pop ecx
ret
 FillingAnArray ENDP
END main
```

QUESTION: 34

Ans)

```
INCLUDE Irvine 32.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
```



Page: 13

```
call clrser
mov j, 0
mov k, 10
mov ESI, OFFSET array
mov ECX , N
call Summing Array Elements InRange
call Write Int
cell crf
mov j, -10
mov k, 0
mov ESI, OFFSET array
mov ECX , N
Call summing Array Elements In Range
Call WriteInt
cell crf
Call wait msg
exit
```

2020/6/10 05:15

```
main ENDP
Summing Array ElementsInRange PROC
  push ecx,
  push exi
  mov eax,0

l1:
    mov ebx, [esi]
    cmp ebx,j
    jge true 1
    Jmp next

  true 1:
         cmp ebx,k
     jle true 2
     jmp next
  true 2:

     add eax,ebx

next
     add esi,4

Loop l1
```

re

Sum

(

Ans)

•

10

11

by

10

11

11

Co

2020/6/10 05:15

```
    Pop esi
    Pop ecx
ret
Summing Array Elements In Range ENDP
END main.
```

Question: 35

Ans) Include Irvine 32.inc

```
.data
    byte 1 BYTE 1111110b, 1101110b,
1000111b, 11001100b, 11001010b,
11001010b, 11001010b, 1100.
byte 2 BYTE 1111110b, 1101111b,
10001110b, 1110100b, 11001100b,
11001011b, 11001010b, 11001010b,
1100
Code:
    main PROC
```

2020/6/10 05:15

```
mov esi, OFFSET byte 1

mov ecx, SIZEOF byte 2

call PE Check

Call Write Int

mov esi, OFFSET byte 2
mov ecx, SIZEOF byte 2
 call PFcheck
 call writeInt

exit
     main ENDP
pfcheck PROC

 ; eax PF=1 TRUE PF=0 False

 ;esi, ecx

   push esi
   push ecx

  sub ecx, 1
 mov al,0
 xor al,0

  mov al, [esi]

L1;
     inc esi
     xor al, [esi]
```

2020/6/10 05:15

```
mov bl, [esi]
Loop L1

jP LPE1
 mov eax, 0
 jmp LEND
 LPF 1:

mov eax, 1
LEND:
pop ecx.
 pop esi

rec

PF check ENDP

 END main.
```