

Grand Assignment Goal

22K-6006

Question : 01

L2 :

Include iostream.h

```

        add esi,4
        add edi,4
        loop L1
        set
        .DATA
array1 SDWORD 1,2,3,4,5
array2 SDWORD 1,0,3,0,5
long int EQU ($ - array1)/4
main PROC

```

.

```

array1 SDWORD 1,2,3,4,5
array2 SDWORD 1,0,3,0,5
long int EQU ($ - array1)/4
main PROC

```

```

        mov esi,offset array1
        mov edi,offset array2
        mov ecx,length
        invoke contmpatches
        esi,edi,ecx
        call writeint
        exit
        .CODE
main PROC
Contmpatches PROC Uses esi,edi,ecx;
ptr1 PTR SDWORD
ptr2 PTR SDWORD
size1 SDWORD
size2 SDWORD

```

```

        mov esi,ptrarray1
        mov edi,ptrarray2
        mov ecx,size1
        xor eax,eax

```

```

L1 :
        mov edx,[esi]
        mov edi,[edi]
        cmp edx,[edi]
        jne L2
        inc eax

```

Question : OR

Include "main32.inc"

main PROC

- Deref

array 1 DWORD 10 DUP(0)
array 2 DWORD 10 DUP(0)

- CODE

Extended-Sub Proc Uses es:edi,ecx
esi, edi, ecx.

pArray1 PTR dword
pArray2 PTR dword
size's dword

exit

main endp

end main

mov esi, pArray1
mov edi, pArray2
mov ecx, size's
xor eax, eax

L1:

mov edx, [esi]
mov edx, [edi]
add esi, 4
add edi, 4

loop L1

ret

Extended-Sub EndP

mov esi, offset array1
mov edi, offset array2

Invoke Extended-Sub,
esi, edi, ecx.

Question : 03

findable
main Proc

Date: _____

Dafia
array dword 100 DUP(0)
f... diised. ?

Page 2

new offset array
new ack, bright
call reading

• CODE
findValue proc uses ecx

mon val, cor
Brooke digital, espe, val
call awaiting

array : The Dword
size : dword
val : dword

exit
main endof
end main

DIAGRAM

MOV OSIO, PTR[SI]
MOV AX, SIZE
MOV BX, VAL

1

```
cmp [esi],eax  
je nomd  
add esi,4  
loop  
jmp notnomd.
```

1

found:
MOV CX=1
SET

Not found

Nov 2000

Question: 4

Date: _____

L1:

```

        cmp [esi], 3
        jne L2
        inc ebx ; ebx ++
array1 DWORD 1,2,3,3,3,4,5
array2 DWORD 1,2,3,4,5
length1 EQU ($ - array1)/4
length2 EQU ($ - array2)/4

L2:
    xor ebx, ebx

```

L3:

```

    add esi, 4
    loop L1
    jmp NOT FOUND

```

Invoke findthree, esi, ecx

Found:

```

    mov esi, offset array2
    mov eax, length of array2
    set
    invoke findthree, esi, ecx
    NOT FOUND:
    mov eax, 0
    ret
    main endp
    end main

```

Findthree Proc Uses esi, ecx

```

    PfArray PTR DWORD
    Size: DWORD

```

```

    mov esp, offset esi
    mov ebx, size, ecx
    xor eax, eax
    xor ebx, ebx

```

Question : 05

Question : 06

Include Irvine32.inc
.DATA

dividend DWORD 0DHABH
divisor DWORD 0AH

String BYTE "ABC",0
char BYTE "X"

- CODE
divide Proc Uses eax,
divide = Proc DWORd
divisor = DWORd

.CODE

Str TrimLead PROC USES esi
 mov eax, dividend

cmp eax, 5h

jne DONE

idiv divisor

MOV esi, pString

Invoke divide, eax, divisor

L1:

DONE: ret

cmp Byte PTR [esi], ch
 Divide ENDP

jne EDone
 inc esi

jmp L1

mov eax, dividend

mov eax, divisor

invoke divide, eax, ecx

exit

DONE:

mov eax, esi

ret

Str TrimLead endP

main endP

main PROC
 mov eax, offset String

movzx ecx, char

invoke StrTrimLead, eax, ecx,

call writeString

exit

end main

Question : 7

1) Reg, Mem 0000 0011 1000 0000
 $D = 1 \quad N = 1$

0	3	8	0
---	---	---	---

$M_{0d} = 10 \rightarrow 0010 = 000$
 $mmmm = 000$

2) Reg 0000 0011 1000 0000
 $N = 1 \quad M_{0d} = 11$

1	1	1	1
---	---	---	---

$1111 \quad 1111 \quad 1000 \quad 0100$

3) Reg, M + disp (06)
 $D = 1 \quad N = 1$

$M_{0d} = 01 \rightarrow 0001 = 000$
 $mmmm = 110$

1	0	0	1
---	---	---	---

$1000 \quad 1001$

[FFC8]

1 000 1001

8 9

1001 1100
 a c

$\rightarrow = 011$

$mmmm = 100$

1001 1100
 a c

[899C 004]

1000 0110

8 B

0100 0110

4 6

[8B46]

4) $D = 1 \quad N = 1$

001010 00101011

$M_{0D} = 00 \quad 2 \quad B$

000001 00001110

m = 110

0 E

[2B0E]

Q8

1) B9 00 12

1100 1001

0000 0000

[MOV ex, 1200]

2) 8C 85 DC 01

8C = 1000 1100

85 = 1000 0101

3) 8B 87 56 78

8B = 1000 1011

87 = 1000 0111

[MOV Ax, [BX] + 7856]

4) RR 1D

RR = 0010 1000

1D = 0001 1101

[Sub [DI], BL]

Question : Q9

Date: _____

1) $A = 5 \times A - 7$

• DATA

A word ?

• CODE

mov ax, A

imul ax, 5

sub ax, 7

mov ax, A

sub ax, B

add B, 10

imul ax, B

mov B, ax

mov A, ax

call writeDec

call crlf

exit

main endp

end main

movzx eax, B

call writeDec

exit

main endp

end main

2) $B = (A - B) \times (B - 10)$

• DATA

A word ? B word ?

Bres word ?

• CODE

mov edx, offset A

call writeString

call readInt

mov A, cx

call readInt

mov B, eax

• CODE

call readInt

mov A, eax

mov ax, A

imul ax, 9

sub ax, 6

mov A, ax

call writeDec

call crlf

main endp

end main

4) $A^2 + B^2 + C^2 \rightarrow$ Set CF, CLR CF

Clear CF:

clc

A dword? B dword?

done:

- CODE

main PROC

movzx eax, CF

call writeDec

call clc

mov edx, offset A
call writeString
call ReadInt
mov A, eax

call writeDec
end main

call ReadInt
mov B, eax
call ReadInt
mov C, eax

mov eax, A
imul eax, A
add eax, B
imul eax, B
add eax, C
imul eax, C

jne SetCF
jnc alrcf

Set CF:

stc

jmp done

QUESTION 10 :-

- DATA

N DWORD ?

N DWORD ?

C DWORD ?

R DWORD ?

mov R, eax

jmp euclideanLoop.

gcdfound:

mov eax, N

call writedec

call crlf

- CODE

call readint

mov N, eax

call readint

mov R, eax

main endp

end main.

mov Q, 0

mov R, 0

; Euclidean Algorithm

euclidean loop:

cmp R, 0

je gcdfound

mov eax, M

mov ebx, N

div ebx

mov Q, eax

mov R, edx

mov eax, N

mov ebx, R

mov N, ebx

Question : 11

Date: _____

DARIA

L3:

score dword 5 DUP(67, 45, 98, 33) mov eax, average[ecx*4]
average dword 4 DUP(0) call writeint

qpmStudent dword 5 call ori
qpmExam EDV 5 dec ecx

CODE

jne L3
exit

calculate Avg Proc

main endp

mov eax, 0

mov edi, 0

L2:

add eax, score[edi*4 + ecx + 4 * qpmStudent]

inc edi
cmp edi
jle L2

idiv qpmStudent

mov average[ecx*4], eax

dec ecx

jge L1
ret

CalculateAvg endp

main PROC

call CalculateAverage

mov eax, 2 qpmExam

dec c ecx

Specification : LR

Date:

DATA

doc1 Byte 1,2,3,4,5

doc2 Byte 6,2,3,4,7

length Byte (4 - doc1)

matchCont Byte 0

plagiarismdet Byte 0

DONE :

mov eax, plagiarismdet

call writeInt

exit

main endp

main endp

CODE

mov esi, offset doc1

mov edi, offset doc2

mov ecx, length

xor eax, eax

L1:

mov al, [esi]

cmp al, [edi]

jne L2

inc matchCont

cmp contMatch, 3

jne L3

inc plagiarism, 1

jmp done

L2:

mov matchCont, 0

L3:

inc esi

inc edi

inc ecx

loop L1