National University of Computer and Emerging Sciences



Assignment 03

COAL

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No. | 19F-0228 |
| INSTRUCTOR | Mr. Abdul Qadir Bilal |
| Semester | Fall 2020 |

Task 1

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 1

;Name :Dry Run

INCLUDE Irvine32.inc

.data

var DW 9988h, 7051h, 1134h, 4567h, 8815h, 7711h, 5511h

.code

main PROC

mov eax, white + (lightred \* 16)

call SetTextColor

mov ecx,6

mov esi,offset var

mov eax,6+9

L1:

add eax,[esi]

inc esi

call writedec

call crlf

loop L1

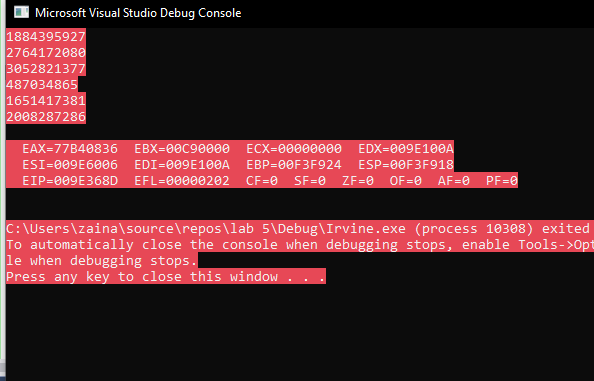
call dumpregs

exit

main ENDP

END main

# **Snip:**



Task 2

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 2

;Name :Grade Calculation

INCLUDE Irvine32.inc

.data

input\_grades byte "Input grades offset 20 Students ",0

for\_counter\_1 byte "Number of A grades = ",0

for\_counter\_2 byte "Number of B grades = ",0

for\_counter\_3 byte "Number of C grades = ",0

for\_counter\_4 byte "Number of D grades = ",0

for\_counter\_5 byte "Number of F grades = ",0

Result DW 20 dup(0)

counter\_A dword 0

counter\_B dword 0

counter\_C dword 0

counter\_D dword 0

counter\_H dword 0

A\_grade = 'A'

B\_grade = 'B'

C\_grade = 'C'

D\_grade = 'D'

F\_grade = 'F'

space byte " , ",0

.code

main PROC

mov eax, white + (lightred \* 16)

call SetTextColor

call grade\_reading

CALL CRLF

call grade\_calculation

mov eax,counter\_A

mov edx, OFFSET for\_Counter\_1

call writestring

call writeDec

call crlf

mov eax,counter\_B

mov edx, OFFSET for\_Counter\_2

call writestring

call writeDec

call crlf

mov eax,counter\_C

mov edx, OFFSET for\_Counter\_3

call writestring

call writeDec

call crlf

mov eax,counter\_D

mov edx, OFFSET for\_Counter\_4

call writestring

call writeDec

call crlf

mov eax,counter\_H

mov edx, OFFSET for\_Counter\_5

call writestring

call writeDec

call crlf

call WaitMsg

exit

main ENDP

grade\_reading proc

mov ecx,LENGTHOF Result

mov esi,OFFSET Result

mov edx, OFFSET input\_grades

call writestring

call crlf

.WHILE (ecx !=0) ;20 dfa

call readchar ;only capital letter

mov dx,ax

call writechar

mov edx ,offset space

call writestring

mov [esi],eax ;mov krdea array ma

inc esi

dec ecx

.ENDW

RET

grade\_reading ENDP

;=============================================================

;Grade Counter

;increment and decrement in variables

;to get total number of grades

;============================================================

grade\_calculation PROC

mov ecx,LENGTHOF Result

mov esi,OFFSET Result

.WHILE ecx !=0

mov bl,[esi]

.IF bl == A\_grade

add counter\_A,1

.ELSEIF bl == B\_grade

add counter\_B,1

.ELSEIF bl == C\_grade

add counter\_C,1

.ELSEIF bl == D\_grade

add counter\_D,1

.ELSEIF bl == F\_grade

add counter\_H,1

.ENDIF

inc esi

dec ecx

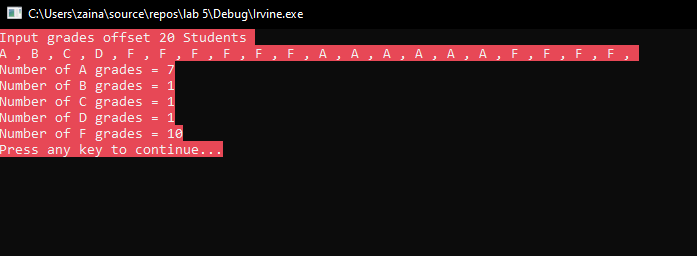
.ENDW

RET

grade\_calculation ENDP

END main

# **Snip:**



Task 3

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 3 part 1

;Name :swapping using registers

INCLUDE Irvine32.inc

.data

val1 byte "Input 1st Value ",0

val2 byte "Input 2nd Value ",0

msg1 byte "Before Swapping ",0

msg2 byte "After Swapping ",0

space byte " ",0

.code

main proc

mov eax, white + (lightred \* 16)

call SetTextColor

mov edx ,offset val1

call writestring

call readdec

mov ebx ,eax

mov edx,offset val2

call writestring

call readdec

call swap\_using\_reg

exit

main ENDP

;===========================================

;===========================================

;Printing message of before swapping

;Moving values in registers to save for swapping purpose

;prinitng them

;using xhange exchanged values reg

;printing swapped messages

;===========================================

swap\_using\_reg proc

mov edx ,offset msg1

call writestring

mov ecx,eax

mov eax,ebx

call writedec

mov edx ,offset space

call writestring

mov eax,ecx

call writedec

call crlf

XCHG eax,ebx

mov edx ,offset msg2

call writestring

mov ecx,eax

mov eax,ebx

call writedec

mov edx ,offset space

call writestring

mov eax,ecx

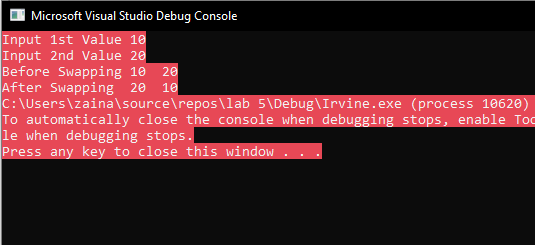
call writedec

ret

swap\_using\_reg endp

END MAIN

# **Snip:**



Task 5

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 5

;Name :xor ,2 use registers

INCLUDE Irvine32.inc

.data

msg1 db "Enter Integer ", 0

var dd 0

.code

main PROC

mov eax, white + (lightred \* 16)

call SetTextColor

mov edx,offset msg1

call writestring

call readint

mov var,eax

mov ebx,-1

add eax,ebx

call crlf

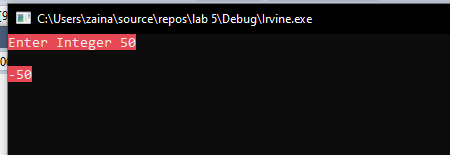
xor eax, ebx

call writeint

main ENDP

END main

# **Snip:**



Task 6

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 6

;Name :simple Dry Run

INCLUDE Irvine32.inc

.data

.code

main PROC

mov al,0D4h

ror al,3 ;--------------------> AL = FE9A

mov al,0D4h

rol al,7 ;--------------------> AL =FF6A

stc

mov al,0D4h

rcl al,1 ;--------------------> AL =FCA9

stc

mov al,0D4h

rcr al,3 ;--------------------> AL =F93A

exit

main ENDP

END main

Task 8

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 8

;Name :Tell the correct order

.MODEL SMALL

.STACK 100H

.DATA

a DB 10h

.CODE

MAIN PROC

MOV BL, 10 ;-------------->1

MOV AL, 20h ;-------------->2

ADD AL, A ;-------------->3

JNZ LBL1 ;-------------->4

JP LBL2 ;-------------->ZF:0

JMP EXIT\_LBL

LBL1: ;-------------->5

XOR AL, 63;-------------->PF=1

JZ LBL1;-------------->7

JP LBL3;-------------->8 PF=1

JNP LBL4

JMP EXIT\_LBL

LBL2:

AND BL, 0

OR AL, 48

CMP AL, 30h

JP LBL5

JG EXIT LBL

JS LBL5

JZ LBL1

LBL3:;-------------->9

MOV DL, 0;-------------->10

JZ LBL1;-------------->11

OR DL, 18;-------------->12

JZ LBL1;-------------->13

JS LBL5;-------------->14

LBL4: ;-------------->SF:1

MOV DL, 1

AND DL, 0

JZ LBL2

JMP EXIT LBL

LBL5:;-------------->15

AND DL, 0;-------------->16 ZF:1

JNZ LBL1;-------------->17 ZF:1

JZ EXIT LBL;-------------->18

EXIT LBL:;-------------->19

exit ;-------------->20

MAIN ENDP

# END MAIN

Task 9

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 8

;Name :Tell the correct order

.MODEL SMALL

.STACK 100H

.DATA

a DB 10h

.CODE

MAIN PROC

MOV BL, 10 ;-------------->1

MOV AL, 20h ;-------------->2

ADD AL, A ;-------------->3

JNZ LBL1 ;-------------->4

JP LBL2

JMP EXIT\_LBL

LBL1:

XOR AL, 63;-------------->5

JZ LBL1;-------------->6

JP LBL3;-------------->7

JNP LBL4

JMP EXIT\_LBL

LBL2:

AND BL, 0

OR AL, 48

CMP AL, 30h

JP LBL5

JG EXIT LBL

JS LBL5

JZ LBL1

LBL3:

MOV DL, 0;-------------->8

JZ LBL1;-------------->9

OR DL, 18;-------------->10

JZ LBL1;-------------->11

JS LBL5;-------------->12

LBL4:

MOV DL, 1

AND DL, 0

JZ LBL2

JMP EXIT LBL

LBL5:

AND DL, 0;-------------->13

JNZ LBL1;-------------->14

JZ EXIT LBL;-------------->15

EXIT LBL:

exit

MAIN ENDP

# END MAIN

Task 10

# **Source Code:**

*;Author :Muhammad Zain*

;Task :Task 10

;Name :number found

INCLUDE Irvine32.inc

.data

print BYTE " Found = ", 0

Array WORD 1234h, 0ACE2h, 7AB0h, 5678h

Check = 7AB0h

.code

main PROC

mov eax, white + (lightred \* 16)

call SetTextColor

mov ecx, LENGTHOF Array

mov esi, OFFSET Array

.while ecx != 0

mov edx, [esi]

.if dx == Check

mov ebx, Check

mov edx,offset print

call writestring

.break

.endif

add esi, 2

dec ecx

.endw

mov eax, ebx

call crlf

call writeHex

main ENDP

END main

# **Snip:**

