National University of Computer and Emerging Sciences



Assignment 02

COAL

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

Task 1 (a)

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:1A

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

Array1 DW 1000h,2000h,3000h

Array2 DW 7000h,9000h,5000h

Sum DW 3 dup (0)

.code

main PROC

mov esi,OFFSET Array1 ;pointing arrays index

mov edi,OFFSET Array2

mov ecx,LENGTHOF Array1

mov edx,OFFSET Sum ;pointing sum array index

mov eax,0

l2:

add eax,[esi]

add eax,[edi]

mov [edx],eax

mov eax,0

add esi,4

add edi,4

add edx,4

loop l2

call dumpregs

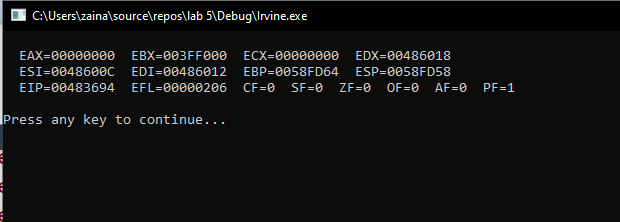
call WaitMsg

exit

main ENDP

END main

# **Snip:**



Task 1 (b)

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:1-b

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

Array1 DW 01000h,02000h,03000h

Array2 DW 07000h,09000h,05000h

Sum DW 3 dup (0)

.code

main PROC

mov ecx,LENGTHOF Sum

mov edx,OFFSET Sum

mov esi,0

mov edi,0

mov ax,0

l1:

add ax,[Array1 + esi]

add ax,[Array2 + edi]

mov [edx],ax

mov ax,0

add esi,4

add edi,4

add edx,4

loop l1

call dumpregs

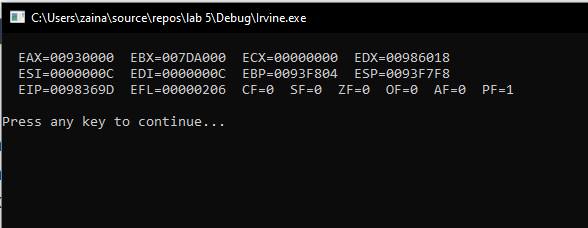
call WaitMsg

exit

main ENDP

END main

# **Snip:**



Task 2

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:2

; Date 11-11-2020

iNCLUDE Irvine32.inc

.data

msg DB "Enter 2 integers : ",0

res DB "result = ",0

.code

main PROC

mov edx,offset msg

call writestring

call ReadInt

mov ebx,eax

call ReadInt

mov edx,OFFSET res

call writestring

.WHILE (ebx != eax) ;

.IF (ebx > eax)

sub ebx,eax

call writeint

.BREAK

.ELSE

sub eax,ebx

call writeint

.BREAK

.ENDIF ;End Loop

.ENDW

call crlf

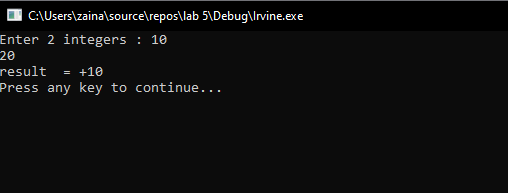
call WaitMsg

exit

main ENDP

END main

# **Snip:**



Task 4

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:4

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

arr DB 1,2,3,4,5,6,7,8,9,10

.code

main PROC

mov ecx,0

.WHILE ecx < 10

mov al,arr[ecx]

mov ah,arr[ecx+1]

mov arr[ecx],ah

mov arr[ecx+1],al

add ecx,2

.ENDW

mov ecx,0

mov eax,0

.WHILE ecx < 10

mov al,arr[ecx]

call WriteDec

inc ecx

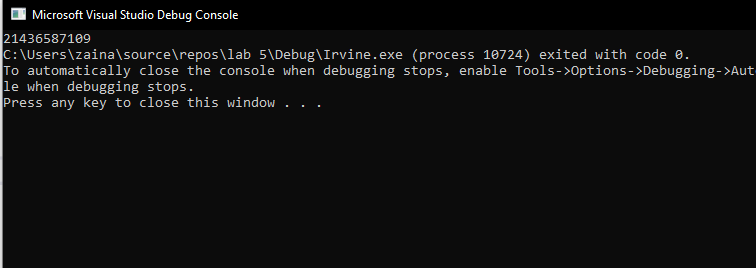
.ENDW

exit

main ENDP

END main

# **Snip:**



Task 5

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:5

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

arr DB 25,30,35,40,45,50

sum DB 0

gap DB ?

.code

main PROC

mov ecx,0

mov eax,0

mov ebx,lengthof arr

sub ebx,1

.WHILE ecx < ebx

mov al,arr[ecx]

mov ah,arr[ecx+1]

.IF al > ah

mov gap,al

sub gap,ah

.ELSEIF ah > al

mov gap,ah

sub gap,al

.ELSE

mov gap,0

.ENDIF

mov al,gap

add sum,al

inc ecx

.ENDW

mov eax,0

mov al,sum

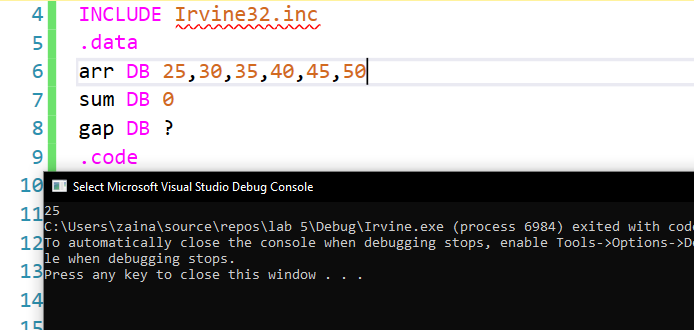
call WriteDec

exit

main ENDP

END main

# **Snip:**



Task 6

# **Source Code**

*; Author:Muhammad Zain*

; Program Name:6

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

arr1 byte 1,2,3,4,5,6

arr2 word SIZEOF arr1 dup (0)

.code

main PROC

mov ecx,LENGTHOF arr1

mov eax,0

mov edi,offset arr2

mov esi,offset arr1

call crlf

L1:

mov eax,[esi]

mov [edi],eax

add esi,TYPE arr1

add edi,TYPE arr2

LOOP L1

mov ecx,LENGTHOF arr2

mov esi,offset arr2

mov eax,0

L2:

mov eax,[esi]

call writehex

call crlf

add esi,TYPE arr2

LOOP L2

call dumpregs

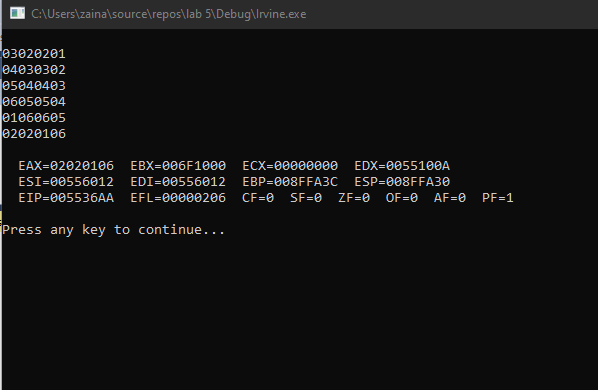
call WaitMsg

exit

main ENDP

END main

# Snip



Task 7

# **Source Code**

*; Author:Muhammad Zain*

; Program Name:7

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

str1 db "Reverse Order ",0

arr1 DWORD 10,20,30,40,50

.code

main PROC

mov ecx, LENGTHOF arr1

mov esi, OFFSET arr1

L1:

mov eax, [esi]

call WriteDEC

call Crlf

add esi, TYPE arr1

LOOP L1

mov esi, OFFSET arr1

mov edi, SIZEOF arr1

add edi, OFFSET arr1

sub edi, TYPE arr1

mov ecx, LENGTHOF arr1

shr ecx, 1

L2:

mov eax, [esi]

mov ebx, [edi]

mov [edi],eax

mov [esi],ebx

add esi, TYPE arr1

sub edi, TYPE arr1

LOOP L2

mov edx,offset str1

call writestring

call crlf

mov ecx, LENGTHOF arr1

mov esi, OFFSET arr1

L3:

MOV eax, [esi]

call WriteDEC

call Crlf

add esi, TYPE arr1

LOOP L3

call dumpregs

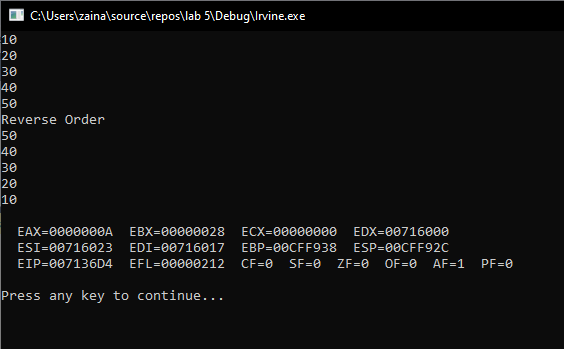
call waitmsg

exit

main ENDP

END main

# **Snip**



Task 8

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:8

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

array1 DB 55,66,77,88,99

msg2 byte "Values after swaping",0

.code

main PROC USES eax ebx ecx edx

mov ecx,Sizeof array1

sub ecx,type array1

mov ebx,0

mov edx,offset msg2

call writestring

call crlf

.WHILE ecx >= ebx

movzx eax, array1[ebx]

movzx edx,array1[ecx]

mov array1[ebx],dl

mov array1[ecx],al

sub ecx,TYPE array1

add ebx,type array1

.ENDW

mov ecx,Sizeof array1

mov ebx,0

.WHILE ebx < ecx

movzx eax,array1[ebx]

call WriteDec

call crlf

add ebx,type array1

.ENDW

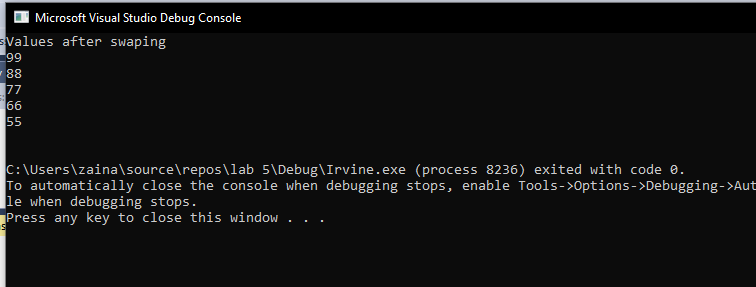
call crlf

exit

main ENDP

END main

# **Snip:**



Task 9

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:9

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

arr DD 55255478h, 6578h, 7845h, 0FFCAh, 45FCh, "hy", "Ass", 9875h

.code

main PROC

MOV ax, word PTR arr ; ax= 05478h

call dumpregs

MOV al, TYPE arr ; al= 04h

call dumpregs

MOV ax, LENGTHOF arr ; ax= 08h

call dumpregs

MOV ax, SIZEOF arr ; ax= 032h

call dumpregs

MOV eax, OFFSET arr + 6 ; eax= "Ass" address

call dumpregs

MOV eax, OFFSET arr ; eax= address of 6 bytes ( arr [0])

call dumpregs

Exit

main ENDP

END main

Task 10

# **Source Code:**

*; Author:Muhammad Zain*

; Program Name:10

; Date 11-11-2020

INCLUDE Irvine32.inc

.data

val16 LABEL WORD

val32 DW 012345678h

arrayD DW 010000h,020000h,030000h,040000h

Bytes DB 010h,020h,030h,040h

Words DW 08Ah,03Bh,072h,044h,066h

Doubles DW 1,2,3,4,5

Pointer DW myDoubles

.code

mov ax, val16 ; AX = 05678h

mov dx, [val16+2] ; DX = 01234h

mov bx,0A69Bh

mov al,6Bh

movzx cx, bl ; CX = 009Bh

movsx cx, al ; CX = 006Bh

mov eax, [arrayD+4] ; EAX = 00020000h

mov ax,7FF0h

add al,10h ; CF=1 SF=0 ZF=1 OF=0

call DumpRegs

add ah,1 ; CF=0 SF=1 ZF=0 OF=1

call DumpRegs

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

mov esi, offset myBytes

mov al, [esi+3] ; AL = 040h

movzx eax, DW ptr myWords ; EAX = 0000008Ah

mov esi, myPointer

mov ax, [esi+6] ; AX = 0004h

movsx cx, al ; CX = 0004h

exit

main ENDP

END main