**Coal lab 8**

**Q1:**

**Text

Description automatically generated**

INCLUDE Irvine32.inc

.data

p1 byte " equal found",0

p2 byte "equal not found",0

sign byte " = ",0

intg dword ?

l dword ?

x dword ?

y dword ?

z dword ?

w dword ?

.code

main PROC

mov w,0

mov esi, offset intg

mov z,esi

mov ecx,4

input:

call readint

mov [esi],eax

add esi,4

loop input

mov esi,z

mov ecx,4

mov edi, offset intg

call crlf

mov z,4

compare:

cmp w, 4

je fin

mov eax,[edi]

mov ecx,z

cmp z,0

je fin

compare2:

add esi,4

mov ebx,[esi]

mov x,eax

cmp eax, ebx

je cond

call crlf

mov edx,offset p2

call writestring

loop compare2

add edi,4

sub z,1

loop compare

cond:

add edi,4

add w,1

cmp w,1

jg equal

jmp compare2

loop cond

equal:

call crlf

mov eax,w

call writeint

mov edx,offset p1

call writestring

mov eax,x

call writeint

add esi,4

jmp compare

cmp w, 4

je fin

loop equal

fin:

exit

loop fin

main ENDP

END main

**Q2:**

**Graphical user interface, text

Description automatically generated**

INCLUDE Irvine32.inc

.data

p1 byte " non zero : ",0

arr SWORD 0, 0, 0, 150, 120, 35, -12, 66, 4, 0

.code

main PROC

mov ecx, lengthof arr

mov esi,offset arr

l1:

mov eax,[esi]

mov edx,0

mov ebx,10

div ebx

cmp edx, 0

jne l2

add esi, 2

loop l1

l2:

mov edx,offset p1

call writestring

mov ax, [esi]

movsx eax,ax

call writeint

call crlf

exit

loop l2

main ENDP

END main

**Q3:**

**A picture containing graphical user interface

Description automatically generated**

INCLUDE Irvine32.inc

.data

arr SWORD 0, 0, 0, 150, 120, 35, -12, 66, 4, 0

var dword ?

x dword ?

y dword ?

.code

main PROC

mov var,5

mov ecx,lengthof arr

mov ebx,ecx

mov y, -1

l1:

cmp y,ecx

jge fin

add y,1

add var,1

mov edx,var

cmp ecx,0

je fin

cmp var,ecx

jl l2

l2:

cmp ecx,edx

jge l3

mov x,1

call else1

l3:

mov x,0

mov eax,x

call writedec

jmp l1

loop l3

loop l1

else1:

mov eax,x

call writedec

jmp l1

loop else1

fin:

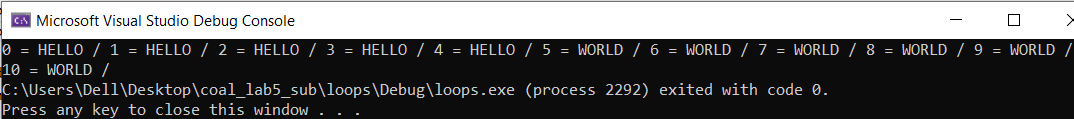
exit

loop fin

main ENDP

END main

**Q4:**

****

INCLUDE Irvine32.inc

.data

p1 byte " = HELLO / ",0

p2 byte " = WORLD / ",0

var dword ?

.code

main PROC

mov var,-1

l1:

add var,1 ;

cmp var , 10

jg fin

mov eax,var

call writedec

cmp var ,5

jl hello

cmp var ,5

jge world

loop l1

hello:

mov edx, offset p1

call writestring

jmp l1

world:

mov edx, offset p2

call writestring

jmp l1

fin:

exit

loop fin

main ENDP

END main

**Q5:**

**Text

Description automatically generated**

INCLUDE Irvine32.inc

.data

p1 byte " enter the number = ",0

p2 byte " found ",0

p3 byte " not found ",0

arr WORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

p4 byte " ,",0

x dword ?

y dword ?

.code

main PROC

mov edx ,offset p1

call writestring

call readint

mov x,eax

mov ecx, lengthof arr

mov esi, offset arr

mov y,esi

display:

mov edx ,offset p4

call writestring

mov ax,[esi]

movzx eax,ax

call writedec

add esi,2

loop display

mov eax,x

mov esi,y

mov ecx, lengthof arr

check:

mov bx,[esi]

movzx ebx,bx

cmp ax,bx

je found

add esi,2

loop check

jmp notfound

found:

call crlf

mov edx , offset p2

call writestring

exit

loop found

notfound:

mov edx , offset p3

call writestring

call crlf

exit

loop notfound

main ENDP

END main

**Q6:**

**Text

Description automatically generated**

INCLUDE Irvine32.inc

.data

arr WORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

p1 byte ",",0

i DWORD 1

temp DWORD ?

.code

main PROC

mov ecx, lengthof arr

l1:

mov temp, ecx

mov eax, 0

mov ebx, 0

sub ecx, i

mov esi, 0

mov edi, 1

l2:

mov ax, arr[esi\*TYPE arr]

mov bx, arr[edi\*TYPE arr]

cmp ax, bx

JG exchange

JLE cont

exchange:

mov ax, arr[esi\*TYPE arr]

mov bx, arr[edi\*TYPE arr]

xchg ax, arr[edi\*TYPE arr]

xchg bx, arr[esi\*TYPE arr]

cont:

inc esi

inc edi

cmp ecx, 0

JLE outsideinner

loop l2

outsideinner:

mov ecx, temp

inc i

loop l1

mov edi, 0

mov ecx, LENGTHOF arr

mov esi, 0

l3:

mov edx,offset p1

call writestring

mov ax, arr[esi\*TYPE arr]

inc esi

call writedec

loop l3

exit

main ENDP

END main

**Q7:**

**Text

Description automatically generated**

INCLUDE Irvine32.inc

.data

msg byte "enter a number : ",0

mon byte " monday",0

tue byte " tuesday",0

wed byte " wednesday",0

thurs byte " thursady",0

fri byte " friday",0

sat byte " saturday",0

sun byte " sunday",0

p2 byte "enter valid number",0

.code

main PROC

call crlf

num:

call crlf

mov edx, offset p2

call writestring

call readdec

cmp eax,1

jl fin

cmp eax,1

je mo

cmp eax,2

je tu

cmp eax,3

je we

cmp eax,4

je th

cmp eax,5

je fr

cmp eax,6

je sa

cmp eax,7

je su

cmp eax,7

jge fin

loop num

mo:

mov edx, offset mon

call writestring

call crlf

jmp num

tu:

mov edx, offset tue

call writestring

jmp num

we:

mov edx, offset wed

call writestring

jmp num

th:

mov edx, offset thurs

call writestring

jmp num

fr:

mov edx, offset fri

call writestring

call crlf

jmp num

sa:

mov edx, offset sat

call writestring

jmp num

su:

mov edx, offset sun

call writestring

jmp num

fin:

call crlf

mov edx, offset p2

call writestring

call crlf

exit

loop fin

main ENDP

END main

**Q8:**

**Graphical user interface, text

Description automatically generated**

INCLUDE Irvine32.inc

.data

p1 byte "Enter a character : ",0

p2 byte "alphabet ",0

p3 byte "not an aplphabet",0

letter byte 65,0

.code

main PROC

mov edx, offset p1

call writestring

call readchar

call writechar

call crlf

mov ecx ,255

mov edx,offset letter

l2:

cmp al,letter

je l1

inc letter

loop l2

call crlf

mov edx, offset p3

call writestring

call crlf

exit

l1:

call crlf

mov edx, offset p2

call writestring

call crlf

exit

loop l1

main ENDP

END main