Task#1

Write a program that takes four input integers from the user. Then compare and display a message whether these integers are equal or not.

```
TITLE LAB 8 Q1
INCLUDE Irvine32.inc
.data
var1 BYTE ?
var2 BYTE ?
var3 BYTE ?
var4 BYTE ?
msg1 BYTE "Input integer: ",0
msgequal BYTE "All are equal.",0
msgnotequal BYTE "All are not equal.",0
```

```
.code
main PROC
mov eax, 0
mov edx, OFFSET msg1
call WriteString
call Crlf
call ReadInt
mov [var1], al
mov edx, OFFSET msg1
call WriteString
call Crlf
call ReadInt
mov [var2], al
mov edx, OFFSET msg1
call WriteString
call Crlf
call ReadInt
mov [var3], al
```

```
mov edx, OFFSET msg1
call WriteString
call Crlf
call ReadInt
mov [var4], al

mov al, [var1]
cmp al, [var2]
jne notequal

mov al, [var2]
cmp al, [var3]
jne notequal

mov al, [var3]
cmp al, [var4]
je allequal

allequal:
mov edx, OFFSET msgequal
call WriteString
jmp _exit
```

```
notequal:
mov edx, OFFSET msgnotequal
call WriteString

_exit:
call DumpRegs
exit
main ENDP
END main
```

```
Input integer:
2
Input integer:
2
Input integer:
2
Input integer:
2
All are equal.
  EAX=00000002  EBX=00407000  ECX=000310AA  EDX=00036014
  ESI=000310AA  EDI=000310AA  EBP=006FFC6C  ESP=006FFC60
  EIP=0003370B  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

Task#2

Use cmp and jumps to find the first non-zero value in the given array:

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

```
TITLE LAB 8 Q2
INCLUDE Irvine32.inc
.data
intArr SWORD 0,0,0,150,120,35,-12,66,4,0
msgfound BYTE "First non zero value found: ",0
```

```
.code
main PROC
mov eax, 0
mov esi, OFFSET intArr
mov ecx, LENGTHOF intArr
L1:
mov ax, [esi]
cmp ax, 0
je continueloop
mov edx, OFFSET msgfound
call WriteString
movzx eax, ax
call WriteDec
jmp _exit

continueloop:
add esi, 2
loop L1
```

```
_exit:
call DumpRegs
exit
main ENDP
END main
```

```
First non zero value found: 150
  EAX=00000096  EBX=004AF000  ECX=00000007  EDX=00F06014
  ESI=00F06006  EDI=00F010AA  EBP=0073FBB0  ESP=0073FBA4
  EIP=00F03696  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

Task#3 Implement the following given code in Assembly and Consider var = 5 , edx = var+1 and counter value from array initialized in task#2.

if (var < ecx) AND (ecx >= edx) then
x = 0
else
x = 1

```
TITLE LAB 8 Q3
INCLUDE Irvine32.inc
.data
intArr SWORD 0,0,0,150,120,35,-12,66,4,0
var DWORD ?
msg BYTE "The value of x is: ",0
x BYTE ?
.code
main PROC
mov [var], 5
mov eax, 0
mov esi, OFFSET intArr
mov ecx, LENGTHOF intArr
mov eax, [var]
inc eax
mov edx, eax
```

```
cmp [var], ecx
jb condition2
jmp printx1
```

```
condition2:
cmp ecx, edx
jae printx
jmp printx1

printx:
mov BYTE PTR [x], 0
jmp _exit

printx1:
mov BYTE PTR [x], 1
jmp _exit

_exit:
mov edx, OFFSET msg
call WriteString
mov eax, 0
mov al, [x]
call WriteDec
call DumpRegs
exit
main ENDP
END main
```

```
The value of x is: 0
  EAX=00000000  EBX=00854000  ECX=0000000A  EDX=00D46018
  ESI=00D46000  EDI=00D410AA  EBP=007FFF68  ESP=007FFF5C
  EIP=00D436C1  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

Task#4 Implement the following given code in Assembly and consider var = 0.

```
while ( var <= 10)
        if (var < 5)
                Print “Hello”
        else
                Print “World”
var = var + 1
end while
```

```
TITLE LAB 8 Q4
INCLUDE Irvine32.inc
.data
var DWORD ?
msg1 BYTE "Hello",0
msg2 BYTE "World",0
.code
main PROC
mov eax, 0
mov [var], eax

whileloop:
mov eax, [var]
cmp eax, 10
ja endwhile
cmp eax, 5
jae printworld
mov edx, OFFSET msg1
call WriteString
call Crlf
jmp incvar
printworld:
mov edx, OFFSET msg2
call WriteString
call Crlf
```

```
incvar:
inc DWORD PTR [var]
jmp whileloop

endwhile:
call DumpRegs
exit
main ENDP
END main
```

```
Hello
Hello
Hello
Hello
Hello
World
World
World
World
World
World

  EAX=0000000B  EBX=010BA000  ECX=00DC10AA  EDX=00DC600A
  ESI=00DC10AA  EDI=00DC10AA  EBP=012FF83C  ESP=012FF830
  EIP=00DC36A6  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

Task#5

Write a program for sequential search. Take an input from the user and find if it occurs in the following array:

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

```
TITLE LAB 8 Q5
INCLUDE Irvine32.inc
.data
msg BYTE "Enter the value to search: ",0
arr WORD 10,4,7,14,299,156,3,19,29,300,20
tosearch WORD ?
foundmsg BYTE "Value found.",0
notfoundmsg BYTE "Value not found.",0
.code
main PROC
mov eax, 0
mov edx, OFFSET msg
call WriteString
call ReadInt
mov [tosearch], ax
mov bx, [tosearch]
mov esi, OFFSET arr
mov ecx, LENGTHOF arr
mov eax, 0
```

```
L1:
mov ax, [esi]
cmp ax, bx
je found
add esi, 2
loop L1
jmp notfound
found:
mov edx, OFFSET foundmsg
call WriteString
jmp _exit

notfound:
mov edx, OFFSET notfoundmsg
call WriteString
jmp _exit

_exit:
exit
main ENDP
END main
```

```
Enter the value to search: 4
Value found.
```

Task#7

Write a program to print weekday based on given number.

```
TITLE LAB 8 Q7
INCLUDE Irvine32.inc
.data
msg1 BYTE "MONDAY",0
msg2 BYTE "TUESDAY",0
msg3 BYTE "WEDNESDAY",0
msg4 BYTE "THURSDAY",0
msg5 BYTE "FRIDAY",0
msg6 BYTE "SATURDAY",0
msg7 BYTE "SUNDAY",0
msg8 BYTE "INVALID",0
number BYTE 6
.code
main PROC
mov al, [number]
```

```
cmp al, 1
jl invalid
jne tues
mov edx, OFFSET msg1
call WriteString
jmp _exit

tues:
cmp al, 2
jne wed
mov edx, OFFSET msg2
call WriteString
jmp _exit
```

```
wed:
cmp al, 3
jne thurs
mov edx, OFFSET msg3
call WriteString
jmp _exit

thurs:
cmp al, 4
jne fri
mov edx, OFFSET msg4
call WriteString
jmp _exit

fri:
cmp al,5
jne sat
mov edx, OFFSET msg5
call WriteString
jmp _exit

sat:
cmp al, 6
jne sun
mov edx, OFFSET msg6
call WriteString
jmp _exit
```

```
sun:
cmp al, 7
jne invalid
mov edx, OFFSET msg7
call WriteString
jmp _exit
```

```
invalid:
mov edx, OFFSET msg8
call WriteString
call Crlf

_exit:
exit
main ENDP
END main
```

```
SATURDAY
```

Task#8

Write a program to check whether a character is alphabet or not.

```
TITLE LAB 8 Q8
INCLUDE Irvine32.inc
.data
msg BYTE "Enter a character: ",0
msgAlpha BYTE "The character is an alphabet.", 0
msgNotAlpha BYTE "The character is NOT an alphabet.", 0
char BYTE ?
.code
main PROC
mov eax, 0
mov edx, OFFSET msg
call WriteString
call ReadChar
call WriteChar
mov [char], al
call Crlf
mov al,[char]
```

```
cmp al, 'A'
jl lowercasecheck
cmp al, 'Z'
jle isalpha

lowercasecheck:
cmp al, 'a'
jl notalpha
cmp al, 'z'
jg notalpha
```

```
isalpha:
mov edx, OFFSET msgAlpha
call WriteString
jmp _exit

notalpha:
mov edx, OFFSET msgNotAlpha
call WriteString
_exit:
exit
main ENDP
END main
```

```
Enter a character: a
The character is an alphabet.
```

Task#6

Write a program for bubble sort on this array.

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

```
TITLE Sorting an Array
INCLUDE Irvine32.inc
.data
arr WORD 1, 4, 2, 3, 6, 5, 10, 8
swap BYTE 0

.code
main PROC
mov eax, 0
    mov ecx, LENGTHOF arr

outerloop:
    mov [swap], 0
    mov esi, OFFSET arr
innerloop:
    mov ax, [esi]
    mov dx, [esi + 2]
    cmp ax, dx
    jbe noswap    ; Skip if in order

    ;swap
    mov WORD PTR [esi], dx
    mov WORD PTR [esi + 2], ax
    mov [swap], 1   ; Indicate a swap occurred
```

```
noswap:
    add esi, 2
    cmp esi, OFFSET arr + (LENGTHOF arr - 1) * 2
    jl innerloop

    cmp [swap], 1
    je outerloop
    mov esi, OFFSET arr
    mov ecx, LENGTHOF arr
    mov eax, 0
printloop:
    mov ax, [esi]
    call WriteDec
    call Crlf
    add esi, 2
    loop printloop
    exit
main ENDP
END main
```

```
1
2
3
4
5
6
8
10
```