ASSIGNMENT 2

Name: Kulsoom Khurshid

Reg#: Sp20-BCS-044

Course: Microprocessor and Assembly Language

1. **Write a program in assembly language that sorts a given integer array using bubble sort algorithm.**

Code

```
INCLUDE Irvine32.inc

.data
array DWORD 8, 7, 3, 13, 1,10
j DWORD 0
sizeArr DWORD LENGTHOF array - 1
i DWORD 0

.code
main PROC
mov ecx,sizeArr
L1:
    PUSH ecx
    mov ecx,sizeArr
    sub ecx,i
    L2:
        mov ebx,j
        mov eax,array[ebx]
        cmp eax,array[ebx+4]
        jg display
        jmp done
    display:
        mov edx,array[ebx+4]
        mov array[ebx],edx
        mov array[ebx+4],eax
    done:
        add j,4
        loop L2
    POP ecx
    mov j,0
    inc i
loop L1
mov esi, OFFSET array
mov ecx, LENGTHOF array
mov ebx, TYPE array
call DumpMem
exit
main ENDP
END main
```

Output

```
Microsoft Visual Studio Debug Console

Dump of offset 00DB6000
------------------------------
00000001  00000003  00000007  00000008  0000000A  0000000D

C:\Users\sa\source\repos\AssemblyProject\Debug\AssemblyProject.
To automatically close the console when debugging stops, enable
le when debugging stops.
Press any key to close this window . . .
```

2. **Given an integer array, write a program that finds the smallest and the largest integers among them. The program must place the smallest number in EAX register and the largest number in EBX register.**

Code	Output

```
INCLUDE Irvine32.inc

.data
arr DWORD 8, 7, 3, 13,1,10
sizeArr DWORD LENGTHOF arr - 1

.code
main PROC
mov ecx,sizeArr
mov esi,0
mov eax,arr[esi]
mov ebx, arr[esi]
L1:
    cmp arr[esi],ebx
    jg func1
    jmp done
func1:
    mov ebx,arr[esi]
done:
    cmp arr[esi], eax
    jl func2
    jmp ans
func2 :
    mov eax, arr[esi]
ans:
    add esi,4
loop L1
call DumpRegs
exit
main ENDP
END main
```

Microsoft Visual Studio Debug Console

```
EAX=00000001  EBX=0000000D  ECX=00000000  EDX=007710AA
ESI=00000014  EDI=007710AA  EBP=00B5F8A4  ESP=00B5F898
EIP=007736A1  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1

C:\Users\sa\source\repos\AssemblyProject\Debug\AssemblyProject.exe (proce
To automatically close the console when debugging stops, enable Tools->Op
le when debugging stops.
Press any key to close this window . . .
```

3. **Write a program that finds out whether an unsigned integer in EAX register is even or odd. If the number is even, place 'Y' in DL otherwise 'N' in DL. (Hint: Least significant bit of an even number is always 0 while that of odd number is 1)**

Code

```
include irvine32.inc

.data
msg byte "Plese enter a no : ",0,13h,10h
msg1 byte "Output: Y",0,13h,10h
msg2 byte "Output: N",0,13h,10h
num word ?
result word ?

.code
main PROC
call clrscr
call crlf
mov edx,offset msg
call writestring
call readint
mov num,ax
mov bx,2
mov ax,num
mov dx,0
div bx
mov result,dx
cmp result,0
JE E
JNE O
E:
    call crlf
    mov edx,offset msg1
    call writestring
    jmp stop
O:
    call crlf
    mov edx,offset msg2
    call writestring

stop:
    exit
main ENDP
end MAIN
```

Output

Microsoft Visual Studio Debug Console

```
Plese enter a no : 6

Output: Y
C:\Users\sa\source\repos\AssemblyProject\Debug\AssemblyProj
To automatically close the console when debugging stops, en
le when debugging stops.
Press any key to close this window . . .
```

4. **Banks use a Personal Identification Number (PIN) to uniquely identify each customer. Let us assume that our bank has a specified range of acceptable values for each digit in its customers' 5-digit PINs. The table shown below contains the acceptable ranges, where digits are numbered from left to right in the PIN. Then we can see that the PIN 52413 is valid. But the PIN 43534 is invalid because the first digit is out of range. Similarly, 64535 is invalid because of its last digit.**
Write a program in assembly language that validates the given PIN number. If any digit is found to be outside its valid range, place the digit's position (between 1 and 5) in the EAX register. If the entire PIN is valid, place 0 in EAX.

Code

```
INCLUDE Irvine32.inc
 ;KEY = 239 ; any value between 1-255
 pin_length = 5 ; maximum buffer size

.data
    endMsg BYTE "Press any key to continue or 1 to exit... "
    pin BYTE 5 dup(?)


.code
main PROC

doAgain:
        mov esi, OFFSET pin
        call getPin
        call Crlf
        mov esi, OFFSET pin
        call validatePin
        call Crlf
        mov edx, OFFSET endMsg
        call WriteString
        call Crlf
        call readChar
        cmp al, '1'
        je ending
        jmp doAgain
ending:


exit
main ENDP
```

```
.data
validStr BYTE "PIN is valid.",0
notValidStr BYTE "PIN is not valid.",0
llimit BYTE '5','2','4','1','3'
hlimit BYTE '9','5','8','4','6'

.code
mov ecx, pin_length
mov edi, 0
l1:

mov bl, [esi]
cmp bl, llimit[edi]
jl  notValid
cmp bl, hlimit[edi]
jg notValid
inc edi
inc esi

loop l1
mov edx, OFFSET validStr
call WriteString
jmp final

notValid:
mov edx, OFFSET notValidStr
call WriteString

final:
ret



ret
validatePin ENDP
```

Output

Microsoft Visual Studio Debug Console

```
Enter PIN: 56789
PIN is not valid.
Press any key to continue or 1 to exit...
Enter PIN: 52413
PIN is valid.
Press any key to continue or 1 to exit...

C:\Users\sa\source\repos\AssemblyProject\Debug\AssemblyProject.exe (proces
To automatically close the console when debugging stops, enable Tools->Opt
le when debugging stops.
Press any key to close this window . . .
```

```
.data
enterPin BYTE "Enter PIN: ",0
numbersOnlyStr BYTE "Numbers only! Try again... ",0

.code
again:
mov edx, OFFSET enterPin
call WriteString

mov ecx, pin_length
mov edi,esi

l1:
call ReadChar
call WriteChar
cmp al, '0'
jl numbersOnly
cmp al, '9'
jg numbersOnly

mov [edi], al
inc edi
loop l1
jmp final

numbersOnly:
call Crlf
mov edx, OFFSET numbersOnlyStr
call WriteString
call Crlf
jmp again


final:

ret
getPin ENDP
```