Name : Ibadullah

Roll No : 19K-0259

LAB 10 Tasks

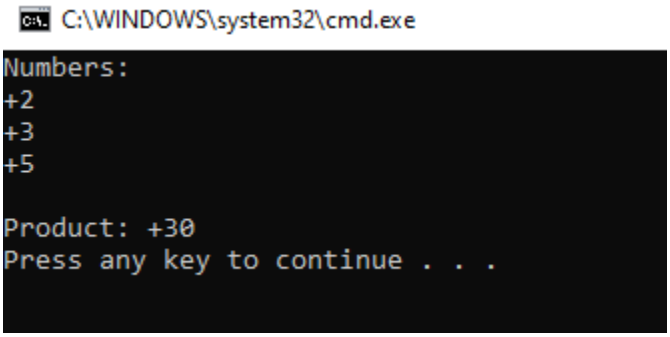
Task 1:

Code:

```
INCLUDE Irvine32.inc
.data
num1 DWORD 2
num2 DWORD 3
num3 DWORD 5
one BYTE "Numbers: ",0
two BYTE "Product: ",0
.code
main PROC
mov eax, 0
mov edx, OFFSET one
call WriteString
mov eax, num1
call crlf
call WriteInt
call crlf
mov eax, num2
call WriteInt
call crlf
mov eax, num3
call WriteInt
mov eax, 0
PUSH num1
PUSH num2
PUSH num3
call threeProd
add esp, 12
call crlf
call crlf
mov edx, OFFSET two
call WriteString
call WriteInt
call crlf
exit
main ENDP
threeProd PROC
	PUSH ebp
	mov ebp, esp
	mov eax, [ebp + 8]
	mov ebx, [ebp + 12]
	mul ebx
	mov ebx, [ebp + 16]
	mul ebx
```

```
	POP ebp
	RET
threeProd ENDP
END main
```

Screenshot:



Task 2:

Code:

```
INCLUDE Irvine32.inc
.data
msg BYTE "Fibonacci Series: ",0
.code
main PROC
mov eax, 0
mov ebx, 0
mov ecx, 7
mov edx, OFFSET msg
call writestring
call crlf
mov edx, 0
PUSH 0
PUSH 1
call Fibonacci
add esp, 8
exit
main ENDP
Fibonacci PROC
	PUSH ebp
	mov ebp, esp
	mov eax, [ebp+12]
	call WriteInt
	call crlf
	mov edx, [ebp+12]
	mov ebx, [ebp+8]
	l1:
		mov eax, edx
		add eax, ebx
		call WriteInt
		call crlf
```

```
            mov ebx, edx
            mov edx, eax
        Loop l1
        pop ebp
        RET
Fibonacci ENDP
END main
```

Screenshot:

Task 3:

Code:

```
INCLUDE Irvine32.inc
.data
array DWORD 383, 4, 210, 65, 10, 20
one BYTE "Array is: ",0
two BYTE "Sorted Array is: ",0
.code
main PROC
mov edx, OFFSET one
call WriteString
mov esi, 0
call crlf
mov ecx, LENGTHOF array
l1:
	mov eax, array[esi*TYPE array]
	call WriteInt
	call crlf
	INC esi
Loop l1
mov ecx, lengthof array
dec ecx
call BubbleSort
call crlf
call crlf
mov edx, OFFSET two
call WriteString
mov esi, 0
call crlf
mov ecx, LENGTHOF array
l2:
	mov eax, array[esi*TYPE array]
	call WriteInt
	call crlf
	INC esi
Loop l2
exit
Main ENDP
BubbleSort PROC
	LOCAL temp : DWORD
	mov esi, 0
	a:
		mov edi, ecx
		mov ecx, 5
		mov esi, 0
	b:
		mov eax, array[esi]
		mov ebx, array[esi+4]
		cmp ebx, eax
		jnc d
		add esi, 4
		Loop b
		jmp e
	d:
		mov temp, eax
```

```
                mov eax, temp
                mov array[esi+4], eax
                mov temp, ebx
                mov ebx, temp
                mov array[esi], ebx
                add esi, 4
        Loop b
        e:
                mov ecx,edi
        loop a
        RET
BubbleSort ENDP
END Main
```

Screenshot: