Quiz 1 - COAL Lab - Section C

(Lab 1 to Lab 5)

Question 1:

You are given this array:

```
myArray1 DWORD 0FFFA5FFFh, 153A5AA5h, 8F0C665Bh
```

Access each element of array using Indirect Operant method and store value at first index to EAX, second value to EBX and third value to ECX. With the help of PTR operator, access the last word from ECX and store it in CX registers. Multiply EAX with EBX and take screenshot of the Registers window. And dump register values on console using Irvine32.inc built-in procedure.

Question 2:

Insert a binary number to AL and insert another binary value to BL. Add AL to BL in such a way that Auxiliary Cary (AC) flag sets. Then take two hexadecimal numbers in ECX and EDX and then multiply them to produce non-zero value in EDX register. Also take screenshot of the CPU registers and flags register at the end of program.

Question 3:

Write a code to print the integer value 3 on console 10 times using LOOP instruction. After completion of LOOP, write an infinite loop to show integer value 9 on console using JMP instruction without using code label.

Question 4:

Store a value 01110111011100011110001110011101b to a variable “myVariable” and then extract each byte from this number one by one using the LABEL operator and print that byte using “WriteInt” procedure.

Question 5:

You are given a string:

```
myString1 BYTE "COAL LAB SECTION C",0
```

Write a loop to access each element of the array using Indexed-operand method and multiply the ASCII value of each element with integer value 3 and add the answer to EAX register. As well as replace the array element value with 0 at the end of each iteration of loop.

After the loop ends, get the AL value from EAX and store it to EBX. Get AH value and store it to ECX. Add EBX and ECX and store the sum into EDX.

And print the whole string on console using Irvine32.inc built-in procedure.

Question 6:

With the help of Newton's second law, find the force exerted by a train moving with the acceleration of 30 ms^2 and having mass of 50000 kg. Store the value of mass in EBX and acceleration in ECX and store the answer in EAX register.

Then solve the following equation with the value of force you calculated:

$$ECX + (EAX * -EBX)$$

Also perform debugging and take screenshot of the registers window at the end of the program.

Question 7:

You are given a program:

```
	Include Irvine32.inc

	const1 = 4
	const2 = 9

	.data
	var1 BYTE const1
	var2 WORD 9

	.code
	main proc
		movzx eax,var1
		mov ebx,const2
		mul ebx
		movzx ecx,var2
		add ebx,ecx
		call WriteInt
		invoke ExitProcess,0
	main endp
	end main
```

Debug this program and take screenshot of the complete byte code (encoded program) shown in the Disassembly window. Change the program to multiply “const1”, “const1” and “var2” and store the answer in EAX and print it in console.

Question 8:

Make corrections in the following code:

```
data
	var1 BYTE F5h
	var2 WORD -35
	var3 WORD -515
.code
main()
	mov eax,var1
	inc eax
	mov ebx,var2
	dec ebx
	mov ecx,var3
	mul ebx
	dec ecx
	mul ecx
	call WriteInt

	INVOKE ExitProcess,0
main ENDP
END main
```

After resolving all the syntax and logical errors the program should produce this answer: +4569696

Question 9:

Store 0FA60h to EDX and store 0FF5Ch to ECX registers. Take backup of flags register in a variable names “flagsBackup” then exchange values of DX and CX without using any 3rd variable or register. Multiply DX to CX and print the answer in console. At the end of program restore the flags register values from “flagsBackup” variable.

Question 10:

You are given this array:

```
myString1 BYTE "HELLO WORLD, COAL LAB QUIZ 1",0
```

Write a program to calculate sum of all the ASCII values of this array elements. And store the sum in a DWORD variable “sum1”. Then exchange the data of the above array with the following array:

```
myString2 BYTE "This is a demo array for you",0
```

After exchanging, again calculate sum of all the elements of “myString1” and store the answer in a DWORD variable “sum2”.

Move a character value “A” to 1st index of array “myString1” and move the character value of “Z” to the last index of array “myString2”. Print both strings on console.

Question 11:

Find the length (number of elements) and size (bytes occupied) of the following array using “Current Location Counter” directive:

```
intArray1 DWORD 10,20,30,70,80,90,100,11,12
```

Question 12:

You are given this data:

```
var1 BYTE 56
var2 WORD 78
     BYTE 59
var3 LABEL BYTE
var4 WORD 6ACFh
```

Perform following activities on this data:

- Get the value 59 using the “var1” and store in EAX register.
- Get the value 59 using BYTE PTR on “var4” and store it in EBX register.
- Using the “var3” label, access the 2nd byte (6A) from the var4.