Betelhem Shefnie
Assembly Language
Assignment #2

Answer the following questions.

Question 1. Describe the relation between a basic assembly code and memory segmentation.
- The register values are stored in a memory space in assembly coding. The segmented memory divides the system into three independent segments which are referenced by the pointer.

Question 2. If we compile the following piece of C++ code, Explain how the registers will act?

```
String str = ”Welcome”
Std::cout<< str;
```

- Length of the string str is stored in a register(edx), the string is stored in a register(ecx) and register(ebx) calls the function std::cout.

Question 3. When an instruction requires two operands what do the first operand and the second operand represents? What does the following addressing mode represent?

- The first operand is generally the destination which contains data in a register or memory location and the second operand is the source. The address mode represents the destination operand.

Question 4. What is the syntax of the MOV instruction? What are the five forms a MOV instruction may have?
- The syntax of the MOV instruction is

```
MOV destination, source.
```

- The five form of the MOV instruction

```
MOV register, register
MOV register, immediate
MOV memory, immediate
MOV register, memory
MOV memory, register
```

Question 5. Find the name of the variable and variable-length from the following code.
https://rextester.com/l/nasm_online_compiler

- Variable length is 13, string is Hello World

Question 6. Why the two codes differ from each other.
Code 1
Code 2
In the first code, line 14, we don't need to do move edx to get it's actual value, 'helloLen' is constant. The second code, in the same line, it's saying mov edx, 'helloLen' to get its actual value.

Question 7. What are the change you have to make in the code Code_Multiple-Initialization to get the following output.
https://rextester.com/GGS72465

Question 8. Describe each line of the following code
https://rextester.com/l/nasm_online_compiler

Line	**Explanation**
3	Syntax to declare data section which are used to declare initialized data
4	"hello" is string of byte which contains 'Hello world!'
5	determines length of the string by subtracting the location of the start of the spring from location after the string
7	declaring text section that is used to keep the actual code.
8	make an identifier accessible to the linker
10	defines the entry point of the program
11	letting the system know to show output
12	register ebx calls the function for the ouput
13	register ecx holds the string pointer
14	register edx holds the string length
16	calls kernel to show output
18	calls for exit
19	line exit with return of 0, program run successfully
20	line is interrupted with system call