Part - 1

1. LEA:
LEA is an instruction that load 'offset' variable while adjusting the address between 16 and 32 bits as necessary. "LEA (16 bit register), (32-bit address)" load the lower 16 bit of the address into the register and "LEA (32-bit register), (16-bit address)" load the 32-bit register with the address zero extended to 32-bit.

offset: offset is an assembler directive in x86 assembly language. It actually means address and is a way of handling the over loading of the 'mov' instruction.

2. Data Segment: · generally points where · variables are defined. Data segment is the starting point of the data segment in a program and data is the name given to this segment.

· Data: Where we define variable In this section we define different variable like, string.

3. ① ASSUME DS: DATA ES: CODE

In this Assembly Language programing there are different registers present for different purpose so we have to assume · DATA is the name given to Data · segment and CODE is the name given to Code Segment register (SS, ES are used in the same way as CS, DS)

Q. To access memory, we can use these four registers: BX, SI, DI, BP. Combining these registers inside [] symbol, we can get different memory locations.

[BX + SI] [BX + DI] [BP + SI] [BP + DI]	[SI] [DI] d16 [BX]	[BX + SI + d8] [BX + DI + d8] [BP + SI + d8] [BP + DI + d8]
[SI + d8] [DI + d8] [BP + d8] [BX + d8]	[BX + SI + d16] [BX + DI + d16] [BP + SI + d16] [BP + DI + d16]	[SI + d16] [DI + d16] [BP + d16] [BX + d16]