Q 1: Write an assembly language program using MASM syntax for the given statement. Using Stack PUSH and POP

Part a: INPUT NAME: Input a name as string terminated by $ and save it in memory (Use function 1 of INT

21H, loop to implement the operation, and register indirect addressing mode to address the memory).

Part b: CASE CONVERSION: Convert the saved string’s case (capital ⇄ small) and save the string with case

conversion. (Use logical operation with loop, use based addressing mode to address the memory

locations).

Part c: VOWELS: Use part a, calculate the number of vowels in the string. (Use conditional jumps)

Part d: CONSONANTS: Use part a, calculate the number of consonants in the string. (Use conditional jumps)

Part e: BINARY CONVERSION: Use part a, convert the saved ASCII values of whole string to binary values

and save the binary characters in the memory. (Use shift or rotate operations, use indexed addressing mode)

Part f: HEXADECIMAL CONVERSION: Use part a, convert the saved ASCII values of whole string to hexadecimal values and save the hexadecimal characters in the memory. (Use multiple shifts/rotateoperations along with loop)

Part g: 1’s BITS: Use part e, find the numbers of ones’ bits in the whole string. (Use indexed addressing mode to address the memory).

Part h: 0’s BITS: Use part e, find the number of zeros’ bits in the whole string. (Use based addressing mode to address the memory)

Part i: REVERSE THE STRING: Use part a, reverse the string and save it in thememory. (Use based and indexed addressing mode)

Part j: WITHOUT VOWELS: Use part a & c, remove the vowels from the string and save it in the memory.

Part k: WITHOUT CONSONANTS: Use part a & d, remove the consonants from the string and save it in the memory.

Part l: PRINTING: Print all the strings in the memory separated by new line. (Using function 9 of INT21H).