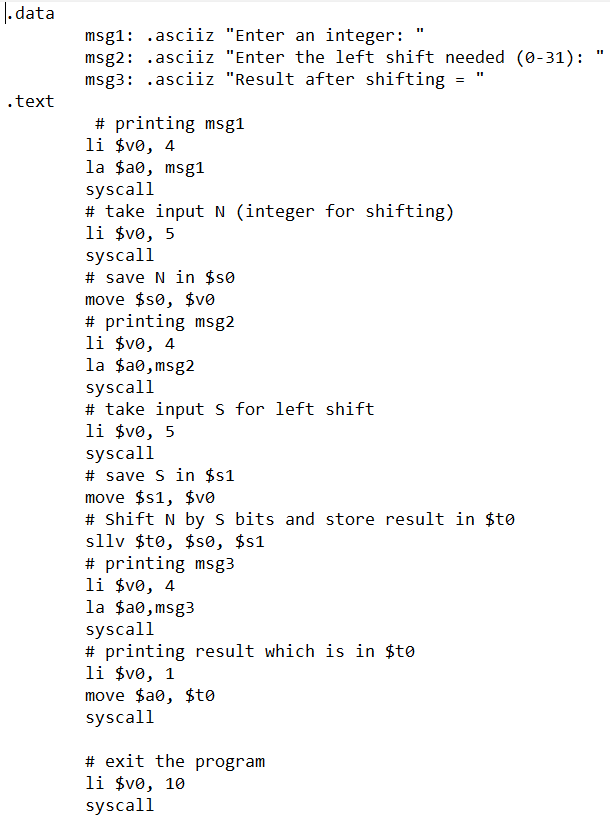
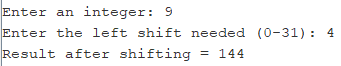
COA LAB -4

1. Write a program in MIPS, that take an integer from the user, perform SLL and Print the output.

Code

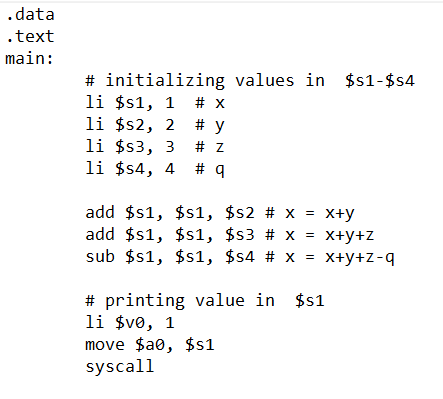


Result:



1. Translate the following Cstatement into MIPS assembly code. Assume that x, y, z, q are stored in registers $s1-$s4. You may use the other registers to hold intermediate results.C stt: x = x + y + z -q;

Code



Result

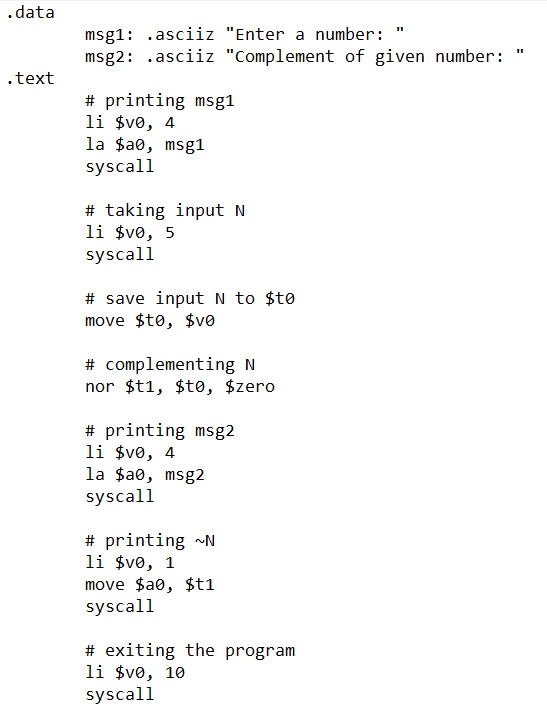


1. How can we implement logical not?Write down the statement and show the result.

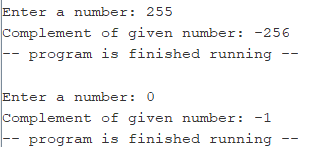
Ans. The NOT operation is done by using the NOR operation with $0 as one of the operands:

nor $t0, $t1, $0 # $t0 🡨 bitwise NOT of $t1

Code



Result



1. Given that: $t2 = 0xabcd1234 and $t3 = 16. Calculate the following.

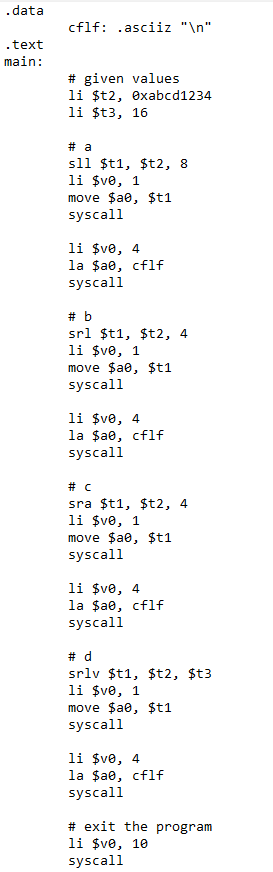
a.sll $t1, $t2, 8

b.srl $t1, $t2, 4

c.sra $t1, $t2, 4

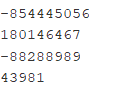
d.srlv $t1, $t2, $t3

Code



Result

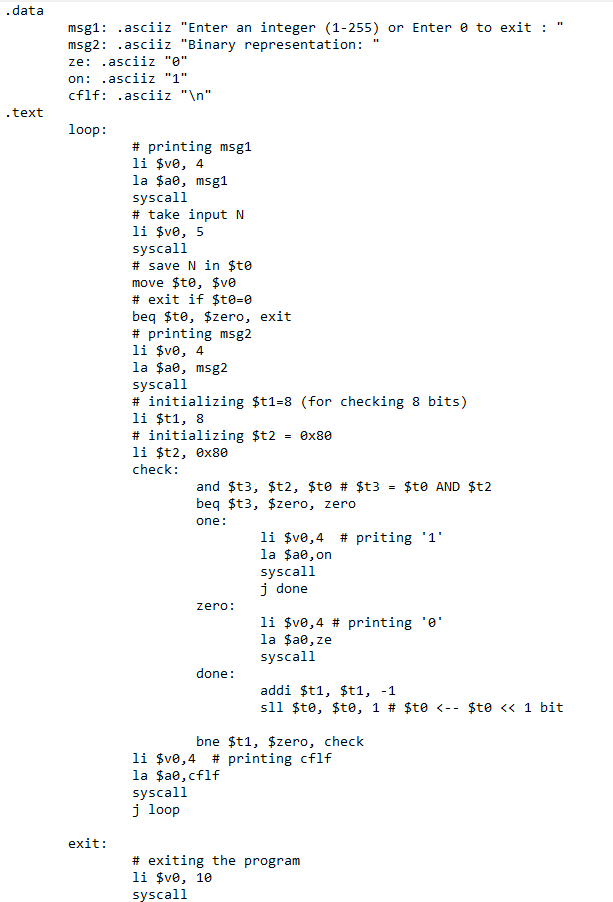
Output of each part of the question is printed in separate lines.



1. Write a program that reads a sequence of decimal numbers on the standard input (one per line) andprints the 8-bit binary representation of each number (one per line), until a 0 is read on the input. You can assume that the numbers entered will be positive and within the range of 8-bit binary numbers (0 to 255).Hint : -you simply need to examine the appropriate bits usingonly logical and shift operations. Shift and AND operations are all you need to generate the binary representation of an integer.

(Please view the code in the next page.)

Code



Result

