**Practical 5**

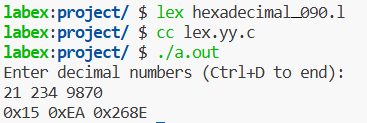
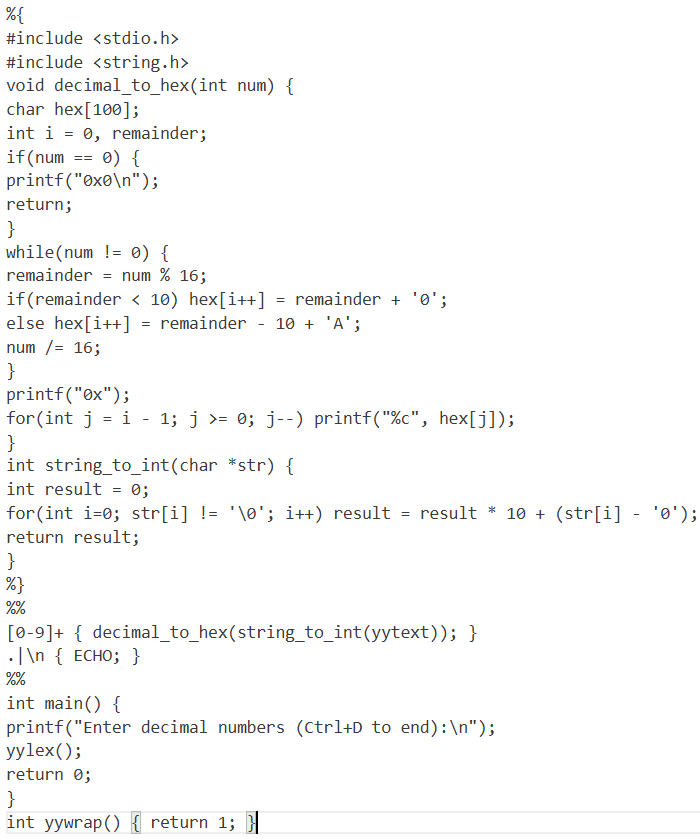
**Aim:** Conversion of decimal to hexadecimal number.

Theory: Decimal (base 10) numbers use digits from 0–9. Hexadecimal (base 16) numbers use digits from 0–9 and letters A–F (where A=10, B=11, ..., F=15). The conversion is done by repeatedly dividing the decimal number by 16 and storing the remainder. The remainders, when read in reverse, give the hexadecimal value.

Program Logic:

* The Lex code uses a regular expression [0-9]+ to match decimal numbers in the input.
* For each match, the string is converted to an integer using a helper function string\_to\_int().
* The integer is passed to decimal\_to\_hex(), which converts and prints the hexadecimal equivalent.
* All other characters are printed unchanged using the fallback rule .|\n.

**Code & Output:**



**Conclusion:** This practical demonstrates the use of Lex in reading numeric patterns and performing a meaningful transformation—converting decimal numbers to hexadecimal format. It also showcases integration with C functions for processing matched patterns, thus bridging pattern matching with computation. This forms a foundational example of how lexical analyzers can be used in real-world compiler tasks such as constant conversion.