

# Atria Institute Of Technology

## System software Lab (18CSL66)

S No	Name of the program	Youtube link
1	1a- Write a LEX program to recognize valid <i>arithmetic expression</i> . Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately	<a href="https://youtu.be/WsLsWQ0-YP8">https://youtu.be/WsLsWQ0-YP8</a>
2	1b- Write YACC program to evaluate <i>arithmetic expression</i> involving operators: +, -, *, and /	<a href="https://youtu.be/SHxQOuWytvQ">https://youtu.be/SHxQOuWytvQ</a>
3	2- Develop, Implement and Execute a program using YACC tool to recognize all strings ending with <i>b</i> preceded by <i>n a</i> 's using the grammar $a_n b$ (note: input <i>n</i> value )	<a href="https://youtu.be/ORIm39hrJX8">https://youtu.be/ORIm39hrJX8</a>
4	3- Design, develop and implement YACC/C program to construct <i>Predictive / LL(1) Parsing Table</i> for the grammar rules: $A \rightarrow \square aBa$ , $B \rightarrow bB / e$ . Use this table to parse the sentence: <i>abba</i> \$	<a href="https://youtu.be/r5iX7EBzJbs">https://youtu.be/r5iX7EBzJbs</a>
5	4- Design, develop and implement YACC/C program to demonstrate <i>Shift Reduce Parsing</i> technique for the grammar rules: $E \rightarrow E+T / T$ , $T \rightarrow T * F / F$ , $F \rightarrow (E) / id$ and parse the sentence: <i>id + id * id</i>	<a href="https://youtu.be/LtY7kCiDdp8">https://youtu.be/LtY7kCiDdp8</a>
6	5- Design, develop and implement a C/Java program to generate the machine code using <i>Triples</i> for the statement $A = -B * (C + D)$ whose intermediate code in three-address form: $T1 = -B$ $T2 = C + D$ $T3 = T1 + T2$ $A = T3$	<a href="https://youtu.be/8gonsMAzjP0">https://youtu.be/8gonsMAzjP0</a>
7	6a- Write a LEX program to eliminate <i>comment lines</i> in a C program and copy the resulting program into a separate file.	<a href="https://youtu.be/6yt-tc7cyh8">https://youtu.be/6yt-tc7cyh8</a>
8	7- Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms. Experiment with different quantum sizes for RR algorithm	<a href="https://youtu.be/ocUTpSH5ohI">https://youtu.be/ocUTpSH5ohI</a>
9	8- Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results	<a href="https://youtu.be/1bFqXLCgYz4">https://youtu.be/1bFqXLCgYz4</a>
10	9- Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.	<a href="https://youtu.be/gOLQK7mvj_k">https://youtu.be/gOLQK7mvj_k</a>