II Year II Semester

L	T	P	C
0	0	3	2

Code: 17CS412

OS AND CD LAB

OS Lab Experiments:

1. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority
2. Multiprogramming-Memory management- Implementation of Fork(), Wait(), Exec() and Exit()
   System calls
3. Simulate all File allocation strategies a) Sequenced b) Indexed c) Linked
4. Simulate MVT and MFT
   a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention.
7. Simulate all page replacement algorithms.
   a) FIFO b) LRU c) LFU etc....

CD Lab Experiments:

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines
2. Simulate First and Follow of a Grammar.
3. Develop an operator precedence parser for a given language.
4. Construct a recursive descent parser for an expression.
5. Construct a LL(1) parser for an expression
6. Design predictive parser for the given language
7. Implementation of shift reduce parsing algorithm.
8. Design a LALR bottom up parser for the given language.2

MINI PROJECT LIST

1. A project on implementation of communication between processes.
2. Given 2 process i and j, you need to write a program to simulate that can guarantee mutual exclusion between the two without any additional hardware support.
3. Implement Banker's algorithm for deadlock avoidance

1. When a Grammar is given as input, write a program to find out the following:
   1. Total Number of Productions in that grammar
   2. Total Number of variables in that grammar
   3. Total Number of terminals in that grammar.
2. When a Grammar is given as input, write a program to find whether there are any recursive productions in the grammar, if there are any find out whether they are left recursive or right recursive.
3. Program for identifying regular expressions: The program should check whether the word entered by user satisfies the following regular expressions or not over the alphabet {a, b}:
   1. Check whether all symbols of the word are in the given alphabet.

2. Contains at least 2 occurrences of any symbol or not.
3. Contains at least 2 consecutive occurrences of any symbol or not.

4. Program for identifying regular expressions: The program should check whether the word entered by user satisfies the following regular expressions or not over the alphabet {0, 1}:

1. The sequence represents a prime number in decimal equivalent or not.
2. Number of occurrences of each symbol are equal to other symbol or not.

5. Generate a symbol table when variables are entered by user.Symbol table contains 2 columns, variable name and its id number Prompt the user with Four Options:

1. Enter new Variable
2. Display entire symbol table
3. search for a variable name using id number
4. Exit

Implement these four options in the program.