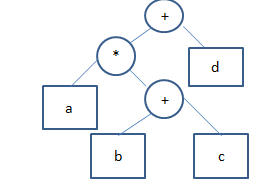
**CSE207 – Data Structures**

**Project :** Expression Tree

**Project Description:**

Expression tree is a binary tree in which each internal node corresponds to operator and each leaf node corresponds to operand. Each sub trees of the expression tree are sub expressions with the root being an operator. For example, expression tree for the infix expression ***a \*(b + c) +d*** would be:



Inorder traversal of expression tree produces infix expression and preorder traversal of it gives prefix expression, whereas postorder traversal of it gives postfix expression.

Now For constructing expression tree, loop through input expression and do following for every character.

1. If character is operand insert that into stack data structure
2. If character is operator then make the operator as parent node of expression tree and take two values from stack, make them its child and insert again in the stack.
3. At the end, only element of stack will be root of expression tree.

In this project, you have to do the followings:

1. Take infix expression as input

2. Convert the expression into appropriate expression tree

3. Show both prefix and postfix expression from the expression tree.

4. Evaluate the expression represented by the expression tree

**CSE207 – Data Structures**

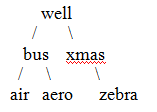
**Project :** Word Dictionary

**Project Description:**

Dictionary means a reference book listing alphabetically terms or names important to a particular subject or activity along with discussion of their meanings and applications. Dictionary should have different functionalities such as add new word, search word, delete word and find similar meaning word. For constructing dictionary you have to do the following:

1. Use Binary Search Tree to store the list of 50 words.
2. Each node of BST stores a key of a dictionary. Key 'k' of a node is always greater than the keys present in its left sub tree. Similarly, key 'k' of a node is always lesser than the keys present in its right sub tree.

**Example:**



In the above example, keys present at the left sub-tree of the root node are lesser than the key of the root node.  And also, the keys present at the right sub-tree are greater than the key of the root node.

In this project you have to do the following,

1. Add new word, search word, delete word and find similar meaning word.
2. All the data may input from a text file where each entry include a word and its definition(meaning)
3. When you search for a particular word then it will give you suggestion of nearest words.

Example: if you search for "Carn"  
Output: the word "carn" was not found,  
Did u mean "cart", " card ", " carrot", "carrom"

**CSE207 – Data Structures**

**Project : Queue Simulator**

**Project Description:**

An important application of queue is queue simulator, a modeling activity used to generate statistics about performance of queue. For this you have to build a single-server single queue simulator. For example, you have to build this simulator for a super shop. This shop has one window and a clerk can serve one customer at a time. The simulator needs to check three events; the arrival of customer, the start of customer’s processing and the completion of customer’s processing. For constructing queue simulator you have to do the following:

1. Use queue for constructing customer’s queue
2. For the server each call history is saved in a stack
3. Use separate structure for current customer status and simulation statistics.

In this project you have to do the following:

1. Print statistics gathered per day during the simulation along with the average queue wait time and average queue service time
2. Show the arrival time, start time, wait time, service time and the number of elements in the queue, each time a customer is completely served.

**CSE207 – Data Structures**

**Project : Call distributor in Call center**

**Project Description:**

A call center is a centralized office used for receiving or transmitting a large volume of requests by telephone. This project will contain multiple servers that will solve different problem over phone and one queue that will hold call of customer. Each of the servers can attend call of customer for 10 min. After finishing the call, another call is given to the free server.For constructing virtual call center you have to do the following:

1. Use linked list for constructing multiple server
2. For each server each call history is saved in a stack
3. Customer call will store in a single queue

In this project you have to do the following:

1. Forward call to different server
2. Count total call received by per server during operating system
3. Total time per customer waiting for their calls

**CSE207 – Data Structures**

**Project : Social Network Graph:**

**Project Description:**

Create a social network graph where users can add friends, find mutual friends, and perform operations like finding the shortest path between two users using graph algorithms like Breadth-First Search (BFS) or Dijkstra's algorithm. The Social Network Graph project aims to build a simple social networking platform that models user connections as a graph. The project allows users to create profiles, add friends, find mutual friends, and perform other basic social networking operations. The focus is on implementing graph data structures and graph algorithms to efficiently manage social network connections.

In this project, you have to do the followings:

User Profiles: Users can create profiles with basic information such as name, age, and location.

Add Friends: Users can send friend requests to other users, and once accepted, they become friends and are connected in the social network graph.

Remove Friends: Users can unfriend or remove connections with existing friends.

Find Mutual Friends: The system can identify mutual friends between two users.

Shortest Path: Implement an algorithm to find the shortest path between two users in the social network graph. This feature enables users to see how they are connected with others.

Friend Suggestions: Suggest potential friends to users based on their existing connections or common interests.

**CSE207 – Data Structures**

**Project : Train Ticket reservation**

**Project Description:**

Train Ticket reservation system will allow to buy your ticket through online. In this project customer can view total number of ticket available and reserve advanced ticket for their journey. The ticket will be provided upon customers’ request in First Come First Served basis. The train ticket reservation system has the following requirements:

1. There are many trains available for each destination which can be stored in linked list.
2. Detail of each train is saved in a stack
3. The customers are organized as a Queue served FCFS.

In this project, you have to do the followings:  
1. Install the train record  
2. Train ticket reservation  
3. Show the train details  
4. Show all train available report

**CSE207 – Data Structures**

**Project : Inventory Management**

**Project Description**:

There is an online shop that contains many kinds of electronic tools and they sell their product depending on the customers’ request. Each client gives an order to buy a tool. The shop employees will search for the required tools upon customers’ request in First Come First Served basis.

The online shop has the following requirements:

1. There are many kinds of tools in their stock. They may have more than one category of each kind and they want store all the information of a tool.
2. The customers have to serve as FCFS basis.
3. Retrieve data of the last sold tools and able to retrieve monthly product wise sell report.

In this project, you have to do the followings:

1. Reserve different kinds of electronic tool’s detail in linked list
2. Store the information of customer’s order in queue
3. Store the information of sold tools of per category in stack
4. Build report that will show monthly selling reports of tools and also show the last sold report.

**CSE207 – Data Structures**

**Project : Mini Parser**

**ProjectDescription:**

A parser is a software component that takes input data (frequently text) and builds a data structure – often some kind of parse tree, abstract syntax tree or other hierarchical structure – giving a structural representation of the input, checking for correct syntax in the process.In this project, you have to build a small parser that will parse branching statement (*if and if-else-else if*) and algebraicexpression of a source program and check for correct syntax in the program.

For example consider the below code.

1. int main()

2. {

3. int a, b,c;

4. a= 3; b=2;

5. if[d>b)

6. c = (a+b)/a;

7. else

8. c = (a+b/b;

9. return 0;

10. }

Here we have found three errors. They are: **if [d>b)** in line no 5**, c = (a+b/b** in line no 8 **and d is an undefined variable** in line no 5. So, in this project you have to build a parser that will check syntactical error of a source code. To construct the **syntax parser** you may consider the following data structure,

1. A file that contains a small program containing mathematical expression and branching statements.
2. Stack for checking correct syntactic structure of given source code

In this project you have to do the following:

1. Build a parser that will parse algebraic expression
2. Parse structure of branching statement
3. If error found then provide an error message with line number of the source program.

**CSE207 – Data Structures**

**Project : Rat in a Maze**

**Project Description:**

Rat in a maze is a popular puzzle game where a mouse start from a start point to final point by visiting different internal spots. To get to his final destination the mouse need to visit one point two or three times or need to backtrack.



A Maze is given as N\*N binary matrix of blocks where source block is the upper left most block i.e., maze [0][0] of the above figure and destination block is lower rightmost block i.e., maze[N-1][N-1]. A rat starts from source and has to reach destination. The rat can move in four directions: right, left, top and down. In the maze matrix, 0 means the block is dead end and 1 means the block can be used in the path from source to destination. To develop the game you may use the following data structure:

* 1. Matrix for the representation of the Maze
  2. Stack or graph to figure out the movement of the rat where a rate cannot move towards the dead block.

In this project you have to do the following:

1. Simulate a mouse movement through the maze
2. Show the path through the maze moved by the mouse

**CSE207 – Data Structures**

**Project : Leave Manager**

**Project Description:**

A multinational company needs leave management system to manage leave of the employees of their company. Generally they offer three types of leave i.e. casual leave (15), sick leave (12) and annual leave (2). Another type of special leave is maternity leave (2) which is only applicable for female employee. They need a leave management system from which they can check available or already used leave. The leave manager has the following requirements:

**a**. Leave information will be stored for each employee

**b.** All employee’s leave request will be processed through the system

**c.** Employees will be able to show their leave details

In this project, you have to do the followings:

1. Store leave detail of all employees in linked list
2. Store the information of employee’s order in queue
3. Store the information of used leave per employee in stack
4. Build report that will show used leave per year of each employee

**CSE207 – Data Structures**

**Project: Email Simulator**

**Project Description:**

Queue is commonly used in network system. An important application of queue is email simulator that processes mail at an average rate of 10 messages per minute. As messages are received they are placed in a queue. The message must arrive randomly, so you need to use a random number to determine whether a message is received. Each minute you can dequeue up to 5 messages per minute. If any messages cannot be send in any processing cycle then it will remain in the queue. At this moment if any new messages come then enqueue it in the queue. For constructing email simulator you have to do the following:

1. Use queue for constructing messages’ queue
2. For the mail history saved email in a stack

In this project you have to do the following:

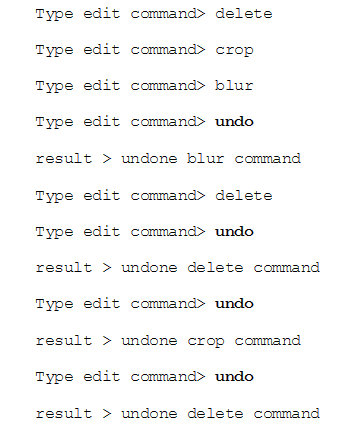
1. Run the simulator for 10 minutes
2. Print total messages processed
3. The average arrival rate
4. The average number of messages sent per minute
5. The average number of messages in the queue in a minute
6. The number of messages sends on the first attempt, the number sent on second attempt and so forth.

**CSE207 – Data Structures**

**Project: Photoshop Edit History**

**Project Description:**

Everybody loves digital photos nowadays. You often use photo editing software such as Adobe Photoshop to create interesting images. Unfortunately your computer got a notorious computer virus, the Undo History Killer. It corrupted all the undo history tracking modules from all applications on your machine. However, you really want to use Photoshop again to touch up your birthday party pictures with the undo/redo feature, when you suddenly realize you have the ability to implement it yourself! The Photoshop app provides history state of up to 20 entries because each editing operation may require large memory space to be stored. For this assignment you should maintain a maximum of 10 entries in the edit history manager. We will simulate this editing operation with simple text strings such as “delete”, “blur”, “crop”, and so on. These text commands are user entered inputs from the terminal (or command line tool). There are two special commands, “undo” and “quit”. When the edit history manager receives an “undo” command, it shows the last command in the edit history. See the below execution example.



For constructing the edit history you have to do the following:

1. Use stack for constructing edit history and performing undo command
2. Your edit manager can store only 10 entry

In this project you have to do the following:

* 1. Take user action as input
  2. Perform the undo operation and back to the previous action
  3. Count how many time manager performs the undo operation