|  |
| --- |
| Implement working principle of Stack using DLL. |
| Create Doubly Linked List to store Employee details and implement Search operation to find particular employee salary using EID.  Employee data fields are: Eid, Ename, Salary |
| Convert the given infix expression to postfix form using Stack.  A\*B+C\*D |
| Demonstrate about how to verify the symbols ({}[]()) are balanced or not using Stack. |
| Create Singly Linked List to store Bank Customers details and implement Search operation to find the account balance of a Customer using his account number.  Customer data fields are: Acc\_No, Customer\_Name, Balance\_Amt |
| Create Doubly Linked List to store Product details and implement Search operation to find particular product’s price using PID.  Product data fields are: PID, PROD\_NAME, PRICE |
| Create Singly Linked List to store Movie details and implement Search operation to find the review rate of a Movie.  Movie data fields are: MID, MOVIE\_NAME,REVIEW\_RATE |
| Implement the CLL using DLL for the Vehicle Database.  Book Fields are: VID, V\_MODEL, VEHICLE\_COMPANY |
| Create SLL to store Product details and implement Search operation to find particular product’s price using PID.  Product data fields are: PID, PROD\_NAME, PRICE |
| Implement the CLL using SLL for the Books Database. And demonstrate searching ‘Book\_Title’ operation for given BID .  Book Fields are: BID, BOOK\_TITLE, AUTHOR\_NAME |
| Create a SLL for the given set of numbers. And then rotate the list as per the given no. of rotations towards right side.  Input: 5, 6, 2, 3, 10, 11, 9, 8  No. of Rotations: 2  Output: 8, 9, 5, 6, 2, 3, 10, 11 |
| Create a DLL for the given set of numbers. And then rotate the list as per the given no. of rotations towards left side.  Input: 5, 6, 2, 3, 10, 11, 9, 8  No. of Rotations: 2  Output: 2, 3, 10, 11, 9, 8,5,6 |
| Create a CLL using SLL for the given set of numbers. And then rotate the list as per the given no. of rotations towards left side.  Input: 5, 6, 2, 3, 10, 11, 9, 8  No. of Rotations: 2  Output: 8, 9, 5, 6, 2, 3, 10, 11 |
| Create a SLL for the given set of numbers. And then rotate the list as per the given no. of rotations towards left side.  Input: 5, 6, 2, 3, 10, 11, 9, 8  No. of Rotations: 2  Output: 2, 3, 10, 11, 9, 8,5,6 |
| Create a DLL for the given set of numbers. And then rotate the list as per the given no. of rotations towards right side.  Input: 5, 6, 2, 3, 10, 11, 9, 8  No. of Rotations: 2  Output: 8, 9, 5, 6, 2, 3, 10, 11 |
| Create a DLL for the given set of numbers. And then create two lists such as list-1 should have odd numbers and list-2 should have even numbers.  Input: 5, 6, 2, 3, 10, 11, 9, 8  Output: List-1: 5, 3, 11, 9 List-2: 6, 2, 10, 8 |
| Create a SLL for the given set of numbers. And then create two lists such as list-1 should have odd numbers and list-2 should have even numbers.  Input: 5, 6, 2, 3, 10, 11, 9, 8  Output: List-1: 5, 3, 11, 9 List-2: 6, 2, 10, 8 |
| Create a CLL using SLL for the given set of numbers. And then create two lists such as list-1 should have odd numbers and list-2 should have even numbers.  Input: 5, 6, 2, 3, 10, 11, 9, 8  Output: List-1: 5, 3, 11, 9 List-2: 6, 2, 10, 8 |
| Create a DLL for the given set of numbers. And create a list by filtering the prime numbers in the given input list.  Input: 5, 6, 2, 3, 8, 7, 9  Output:  Prime List: 5, 2, 3, 7 |
| Create a SLL for the given set of numbers. And create a list by filtering the prime numbers in the given input list.  Input: 5, 6, 2, 3, 8, 7, 9  Output:  Prime List: 5, 2, 3, 7 |
| Create a CLL using SLL for the given set of numbers. And create a list by filtering the prime numbers in the given input list.  Input: 5, 6, 2, 3, 8, 7, 9  Output:  Prime List: 5, 2, 3, 7 |
| Create a DLL for the given set of numbers. And create a list by finding their Factorial numbers.  Input: 0, 1, 2, 3, 4, 5  Output:  Factorial List: 1, 1, 2, 6, 24, 120 |
| Create a SLL for the given set of numbers. And create a list by finding their Factorial numbers.  Input: 0, 1, 2, 3, 4, 5  Output:  Factorial List: 1, 1, 2, 6, 24, 120 |