| Sr. | Practical |
|---|---|
| 1 | WAP to convert seconds into hours, minutes & seconds and print in HH:MM:SS [e.g. 10000 seconds mean 2:46:40 (2 Hours, 46 Minutes, 40 Seconds)]. |
| 2 | WAP to convert number of days into year, week & days [e.g. 375 days mean 1 year, 1 week and 3 days]. |
| 3 | WAP to find the sum of 1 + (1+2) + (1+2+3) + (1+2+3+4)+ …+(1+2+3+4+….+n). |
| 4 | WAP to print Armstrong number from 1 to 1000. |
| 5 | WAP to print Pascal triangle. |
| 6 | WAP to find the largest two numbers in a given array. |
| 7 | WAP to sort the N names in an alphabetical order. |
| 8 | Write a program to insert a number at a given location in an array. |
| 9 | Write a program to delete a number from a given location in an array. |
| 10 | Write a program to insert a number in an array that is already sorted in an ascending order. |
| 11 | Write a program to delete a number from an array that is already sorted in an ascending order. |
| 12 | Write a program to merge two unsorted arrays. |
| 13 | WAP to replace lowercase characters by uppercase & vice-versa in a user specified string. |
| 14 | How stack can be used to recognize strings aca, bcb, abcba, abbcbba? Write a C program using stack which takes string as input and checks whether it follows above mentioned rule or not. |
| 15 | WAP to identify whether the given string is palindrome or not using stack. |
| 16 | Write a program to check nesting of parentheses using stack. |
| 17 | Implement Priority Queue using array that performs following operations: INSERT, DELETE, DISPLAY |
| 18 | Implement Dequeue (Double Ended Queue) using array that performs following operations: INSERT, DELETE, DISPLAY |
| 19 | WAP to print node information of a singly linked list in reverse. |
| 20 | WAP to check whether 2 singly linked lists are same or not. |

| | |
|---|---|
| 21 | WAP to remove duplicate elements from a singly linked list. |
| 22 | WAP to swap K$^{th}$ node from beginning with K$^{th}$ node from end in a singly linked list. |
| 23 | Write a program to count the number of nodes in a singly circularly linked list. |
| 24 | WAP to split a circular linked list into two halves. |
| 25 | WAP to delete alternate nodes of a doubly linked list. |
| 26 | WAP to find the smallest and largest elements in the Binary Search Tree. |