<table>
<thead>
<tr>
<th>Code</th>
<th>II B. Tech I Semester (III Semester)</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0502</td>
<td>Object Oriented Programming through C++</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Data Structures</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>A0504</td>
<td>Digital Logic Design</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>A0505</td>
<td>Discrete Mathematics</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>A0011-14</td>
<td>Foundation Elective</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Unix &amp;Shell programming</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>A0010</td>
<td>Skill Development(Generic)</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Practice</td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>C</td>
</tr>
<tr>
<td>A0582</td>
<td>C++ Programming Lab</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>A0583</td>
<td>Data Structures Lab</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>A0584</td>
<td>Digital Logic Design Lab</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>A0591</td>
<td>FOSS LAB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>28</strong></td>
</tr>
</tbody>
</table>
OBJECT-ORIENTED PROGRAMMING THROUGH C++

Objectives: Expertise in object oriented principles and their implementation in C++

UNIT I:
Objectives: Exposure to basics of object oriented mode, C++ programming and I/O in C++


INPUT AND OUTPUT IN C++:
Introduction, Streams In C++, Pre-Defined Streams, Stream Classes, Formatted And Unformatted Data, Unformatted Console I/O Operations, Member Functions Of Istream Class, Formatted Console I/O Operations, Bit Fields, Flags Without Bit Field, Manipulators, User Defined Manipulators.

UNIT II:
Objectives: Focus on Basic concept in C++ programming, Operators, control structures, functions, overloading, recursion
Tokens In C++, Variable Declaration And Initialization, Data Types, Operators In C And C++, Scope Access Operator, Namespace, Memory Management Operators, Comma Operator, Revision Of Decision Statements, Control Loop Statements.

FUNCTIONS IN C++:

UNIT III:
Objectives: Acquaintance with classes, objects and member functions
CLASSES AND OBJECTS:
Introduction, Classes In C++, Declaring Objects, Access Specifiers And Their Scope, Member Functions, Outside Member Function As Inline, Data Hiding or Encapsulation, Classes, Objects and Memory, Static Member Variables, Static Member Functions Static Object, Array Of Objects, Objects As Function Arguments, Friend Functions, The Const Member Functions, The Volatile Member Function, Recursive Member Function, Local Classes, Empty, Static And Const Classes, Member Function and Non- Member Function, Overloading Member Functions, Nested Class.

UNIT IV:
Objectives: Focus on constructors, destructors, variants in them, operator overloading, type conversions
CONSTRUCTORS AND DESTRUCTORS:
Introduction, Characteristic Of Constructors...
Destructors, Applications With Constructors, Parameterized Constructor, Overloading Constructors (Multiple Constructors), Array Of Objects Using Constructors, Constructors With Default Arguments, Copy Constructors, The Const Objects, Destructors, Calling Constructors And Destructors, Qualifier And Nested Classes, Objects, Dynamic Operators and Constructors, Recursive Constructor, Constructor and Destructor With Static Members, Local Vs. Global Object.


**UNIT V :**
**Objective:** Concentration on inheritance, types of inheritance, polymorphism, virtual functions
**INHERITANCE** : Introduction, Reusability, Access Specifiers And Simple Inheritance, Protected Data With Private Inheritance, Types Of Inheritances(Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Multipath Inheritance), Virtual Base Classes, Constructors, Destructors And Inheritance, Object As A Class Member, Abstract Classes, Qualifier Classes And Inheritance, Constructor In Derived Class, Overloading Member Function, Advantages Of Inheritance, Disadvantages Of Inheritance.

**BINDING, POLYMORPHISM AND VIRTUAL FUNCTIONS** : Introduction, Binding In C++, Static (Early) Binding, Dynamic (Late) Binding, Virtual Functions, Rules For Virtual Functions, Array Of Pointers, Pure Virtual Functions, Abstract Classes, Working Of Virtual Functions, Virtual Functions In Derived Classes, Object Slicing.

**UNIT VI :**
**Objectives:** Focus on generic programming, templates, function templates.
**GENERIC PROGRAMMING WITH TEMPLATES** : Introduction, Need Of Template, Definition Of Class Template, Normal Function Template, Working Of Function Templates, Class Template With More Parameters, Functions Templates With More Arguments, Overloading Of Template Functions, Member Function Templates, Recursion With Template Function, Class Template With Overloaded Operators, Class Template Revisited, Class Templates And Inheritance, Container Classes, Types Of Containers, Container Adaptors, Iterators

**Course Outcome:**
- Gain the basic knowledge on Object Oriented concepts.
- Ability to develop applications using Object Oriented Programming Concepts.
- Ability to implement features of object oriented programming to solve real world problems

**TEXT BOOKS :**
2. Object Oriented Programming C++, Joyce Farrell, Cengage
3. Mastering C++, Venugopal, Rajkumar, Ravi Kumar TMH
4. Object Oriented Programming with C++, 2nd ed, Sourav Sahay, OXFORD

REFERENCE BOOKS:
1. The Complete Reference, C++, 4ed, Herbert Schildt, TMH
UNIT I:
Preliminaries of algorithm, Algorithm analysis and complexity,
Data structure- Definition, types of data structures: Stacks, Queues, Linked List, Trees, Graphs-
Advantages & disadvantages.
Recursion: Definition, Design Methodology and Implementation of recursive algorithms, Linear
and binary recursion, recursive algorithms for factorial function, GCD computation, Fibonacci
sequence, Towers of Hanoi, Tail recursion

UNIT II:
SEARCHING & SORTING TECHNIQUES
List Searches using Linear Search, Binary Search, Fibonacci Search
Sorting Techniques: Basic concepts, Sorting by : insertion (Insertion sort), selection (heap sort),
exchange (bubble sort, quick sort), distribution (radix sort ) and merging (merge sort ) Algorithms.

UNIT III:
Stacks and Queues: Basic Stack Operations, Representation of a Stack using Arrays, Stack
Applications: Reversing list, Factorial Calculation, Infix to postfix Transformation, Evaluating
Arithmetic Expressions.
Queues: Basic Queues Operations, Representation of a Queue using array, Implementation of
Queue Operations using Stack, Applications of Queues-Round robin Algorithm, Circular Queues,
Priority Queues.

UNIT IV:
Linked Lists: Introduction, single linked list, representation of a linked list in memory,
Operations on a single linked list, Reversing a single linked list, applications of single linked list
to represent polynomial expressions and sparse matrix manipulation, Advantages and
disadvantages of single linked list, Circular linked list, Doubly linked list

UNIT V:
Trees: Basic tree concepts, Binary Trees: Properties, Representation of Binary Trees using arrays
and linked lists, operations on a Binary tree , Binary Tree Traversals (recursive), Creation of
binary tree from in, pre and post order traversals
Advanced concepts of Trees: Tree Traversal using stack (non recursive), Threaded Binary Trees.
Binary search tree, Basic concepts, BST operations: insertion, deletion

UNIT VI:
Graphs: Basic concepts, Representations of Graphs: using Linked list and adjacency matrix,
Graph algorithms.
Graph Traversals (BFS & DFS), applications: Dijkstra’s shortest path, Transitive closure, Minimum Spanning Tree using Prim’s Algorithm, warshall’s Algorithm

TEXT BOOKS:
1. Data Structure with C, Seymour Lipschutz, TMH
2. Data Structures using C, Reema Thareja, Oxford
3. Data Structures, 2/e, Richard F, Gilberg, Forouzan, Cengage
4. Data structures and algorithm analysis in C, 2nd ed, mark allen weiss

REFERENCE BOOKS:
2. Classic Data Structures, 2/e, Debasis, Samanta, PHI, 2009
3. Fundamentals of Data Structure in C, 2/e, Horowitz, Sahni, Anderson Freed, University Press
UNIT I:
Binary, Octal, Decimal, Hexadecimal Number Systems. Conversion of Numbers From One Radix To Another Radix, r’s Complement and (r-1)’s Complement Subtraction of Unsigned Numbers, Problems, Signed Binary Numbers, Weighted and Non weighted codes

UNIT II:
Logic Gates And Boolean Algebra
Basic Gates NOT, AND, OR, Boolean Theorms, Complement And Dual of Logical Expressions, Universal Gates, Ex-Or and Ex-Nor Gates, Minimizations of Logic Functions Using Boolean Theorems, Two level Realization of Logic Functions Using Universal Gates

UNIT III:
Gate Level Minimization:
SOP, POS, Karnaugh Map Method (K-Map): Minimization of Boolean Functions maximum upto Four Variables, POS and SOP, Simplifications withDon’t Care Conditions Using K-Map.

UNIT IV:
Combinational Logic Circuits

UNIT V:
Introduction to Sequential Logic Circuits

UNIT VI: Registers and Counters
Design of Registers, Buffer Register, Control Buffer Registers, Bidirectional Shift Registers, Universal Shift Register, Design of Ripple Counters, Synchronous Counters and Variable Modulus Counters, Ring Counter,

TEXT BOOKS:
1. Digital Design ,4/e, M.Morris Mano, Michael D Ciletti, PEA
2. Fundamentals of Logic Design, 5/e, Roth, Cengage
REFERENCE BOOKS:
2. Digital Logic Design, Leach, Malvino, Saha, TMH
3. Modern Digital Electronics, R.P. Jain, TMH
UNIT-I: Mathematical Logic and Set Theory


UNIT-II: Relations and Functions

Relations: Ordered pairs and n-tuples, Product Sets and Partitions, Relations and Digraphs, Matrix of Relation, Paths in Relations and Digraphs, Properties of Relations, Equivalence Relations & Partitions, Compatible Relation, Manipulation of Relations, Composition of Relations

Functions: Definition, Composition of functions, Types of Functions, Invertible Function, Permutation Function, Characteristics function of a set with Theorems.

UNIT-III: Group Theory

Binary Operations, Properties, Semigroups, Monoids, Subsemigroup, Submonoid, Isomorphism & Homomorphism, Groups (only definitions and examples) Subgroups and Homomorphism, Cosets and Lagrange’s Theorem, Normal subgroups.

UNIT-IV: Rings, Lattices & Boolean Algebra

Rings, Fields, Integral Domain, Ring Homomorphism (definitions & examples), Lattices: Properties, Types of Lattices, Sub lattices, Isomorphic Lattices, Complemented & Modular Lattices (definitions & examples), Boolean Algebra: Definition, Properties, Simplification of Switching Circuits.

UNIT-V: Graph Theory

Basic concepts of Graph Theory, Digraphs, Basic definitions, Paths and Circuits, Reachability and Connectedness, Matrix representation of graphs, Subgraphs & Quotient Graphs, Isomorphic digraphs & Transitive Closure digraph, Euler’s Path & Circuit (only definitions and examples).

UNIT-VI: Trees and Graphs

Trees, Binary Tree, Labeled Trees, Undirected Trees, Spanning Trees of Connected Relations, Prim’s Algorithm to construct Spanning Trees, Weighted Graphs, Minimal Spanning Trees by Prim’s Algorithm & Kruskal’s Algorithm.

Text Books

1. Discrete Mathematical Structures(3rd Edition) by Kolman, Busby & Ross PHI.


Reference Books


2. Elements of Discrete Mathematics by C. L. Liu.


II Year – I SEMESTER

L T C
2 1 3

FOUNDATION ELECTIVE
II Year – I SEMESTER

L T C
1 1 2

SKILL DEVELOPMENT (GENERAL)
1. Write a C++ program illustrating Variable Scope.
2. Write a C++ program illustrating Swap integer values by reference.
3. Write a C++ program illustrating Checking whether the number is even or odd using Ternary operator.
4. Write a C++ program illustrating interactive program to multiply 2 variables after checking the compatibility.
5. Write a C++ program illustrating interactive program for computing the roots of a quadratic equation by handling all possible cases. Use streams to perform I/O operations.
6. Write a C++ program illustrating to sort integer numbers.
7. Write a C++ program illustrating factorial using recursion.
8. Write a C++ program illustrating pass by value, pass by reference, pass by address.
9. Write a C++ program illustrating Function overloading.
10. Write a C++ program illustrating an interactive program for swapping integer, real, and character type variables without using function overloading. Write the same program by using function overloading features and compare the same with its C counterpart.
11. Write a C++ program illustrating inline functions.
12. Write a C++ program illustrating Friend function.
13. Write a C++ program illustrating Function template.
14. Write a C++ program illustrating Overloading increment, decrement, binary+&<< operator.
15. Write a C++ program illustrating Virtual function.
16. Write a C++ program illustrating Constructor overloading (Both parameterised and default).
17. Write a C++ program illustrating Copy constructor.
18. Write a C++ program illustrating for overloading ++ operator to increment data
19. Write a C++ program illustrating Abstract classes.
20. Write a C++ program illustrating Inheritance (Multiple, Multilevel, Hybrid).
21. Write a C++ program illustrating Virtual classes & virtual functions.
22. Write a C++ program illustrating overloading function template.
23. Write a C++ program illustrating Class template.
DATA STRUCTURES LAB

Exercise 1:
Write recursive program which computes the \( n \)th Fibonacci number, for appropriate values of \( n \). Analyze behavior of the program Obtain the frequency count of the statement for various values of \( n \).

Exercise 2:
Write recursive program for the following
a) Write recursive and non recursive C program for calculation of Factorial of an integer  
b) Write recursive and non recursive C program for calculation of GCD \((n, m)\)
c) Write recursive and non recursive C program for Towers of Hanoi : \( N \) disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

Exercise 3:
a) Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.
c) Write C program that use both recursive and non recursive functions to perform Fibonacci search for a Key value in a given list.

Exercise 4:
a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order  
b) Write C program that implement Quick sort, to sort a given list of integers in ascending order  
c) Write C program that implement Insertion sort, to sort a given list of integers in ascending order

Exercise 5:
a) Write C program that implement heap sort, to sort a given list of integers in ascending order  
b) Write C program that implement radix sort, to sort a given list of integers in ascending order  
c) Write C program that implement merge sort, to sort a given list of integers in ascending order

Exercise 6:
a) Write C program that implement stack (its operations) using arrays

b) Write C program that implement stack (its operations) using Linked list

**Exercise 7:**

a) Write a C program that uses Stack operations to Convert infix expression into postfix expression
b) Write C program that implement Queue (its operations) using arrays.

b) Write C program that implement Queue (its operations) using linked lists

**Exercise 8:**

a) Write a C program that uses functions to create a singly linked list

b) Write a C program that uses functions to perform insertion operation on a singly linked list

c) Write a C program that uses functions to perform deletion operation on a singly linked list

**Exercise 9:**

a) Adding two large integers which are represented in linked list fashion.

b) Write a C program to reverse elements of a single linked list.

c) Write a C program to store a polynomial expression in memory using linked list

d) Write a C program to representation the given Sparse matrix using arrays.

e) Write a C program to representation the given Sparse matrix using linked list

**Exercise 10:**

a) Write a C program to Create a Binary Tree of integers

b) Write a recursive C program for Traversing a binary tree in preorder, inorder and postorder.

c) Write a non recursive C program for Traversing a binary tree in preorder, inorder and postorder.

d) Program to check balance property of a tree.

**Exercise 11:**

a) Write a C program to Create a BST

b) Write a C program to insert a node into a BST.

c) Write a C program to delete a node from a BST.
II Year – I SEMESTER

DIGITAL LOGIC DESIGN LAB

List of Experiments:

1) Verification of Basic Logic Gates.

2) Implementing all individual gates with Universal Gates NAND & NOR.

3) Design a circuit for the given Canonical form, draw the circuit diagram and verify the De-Morgan

4) Design a Combinational Logic circuit for 4x1 MUX and verify the truth table.

5) Design a Combinational Logic circuit for 1x4 De-MUX and verify the truth table.

6) Verify the data read and data write operations for the IC 74189.

7) Design a Gray code encoder and interface it to SRAM IC 74189 for write operation display on 7-segment.

8) Design a Gray code De-coder and interface it to SRAM IC 74189 for read operation display it on 7-segment.

9) Construct Half Adder and Full Adder using Half Adder and verify the truth table.

10) Verification of truth tables of the basic Flip- Flops with Synchronous and Asynchronous modes.

11)Implementation of Master Slave Flip-Flop with J-K Flip- Flop and verify the truth table for race around condition.

12) Design a Decade Counter and verify the truth table.

13) Design the Mod 6 counter using D-Flip -Flop.

14) Construct 4-bit ring counter with T-Flip –Flop and verify the truth table.

15) Design a 8 – bit right Shift Register using D-Flip -Flop and verify the truth table.
FREE OPEN SOURCE SOFTWARE (FOSS LAB)

PROGRAMS:

1. Session -1:
   a) Log into the system  
   b) Use vi editor to create a file called myfile.txt which contains some text
   c) Correct typing errors during creation  
   d) Save the file  
   e) Logout of the system

Session-2:
   a) log into the system  
   b) open the file created in session-1  
   c) add some text  
   d) delete some text  
   e) save the changes  
   f) logout of the system

2.
   a) log into the system  
   b) use the cat command to create a file containing the following data call it mytable use tabs to separate the fields
      1425  ravi  15.65
      4320  ramu  26.27
      6830  sita  36.15
      1450  raju  21.86
   c) Use the cat command to display the file,mytable  
   d) Use the vi command to correct any errors in the file,mytable
   e) Use the sort command to sort the file mytable according to the first field. Call the sorted file mytable(same name).  
   f) Print the file mytable.  
   g) Use the cut and paste commands to swap fields 2 and 3 of mytable. Call it mytable (same name).  
   h) Print the new file,mytable  
   i) Logout of the system.

3.
   1) a)Login to the system  
   b) Use the appropriate command to determine your login shell  
   c) Use the /etc/password file to verify the result of step b.  
   d) Use the who command and redirect the result of a file call myfile 1
   Use the more command to see the contents of myfile 1  
   e) Use the data and who commands in sequence(in one line)such that the output of data of will display on the screen and the output of who will be redirected to a file called myfile 2.Use the more command to check the contents of myfile 2
   2) a)Write a sed command that deletes the first character in each line in a file.  
   b) Write a sed command that deletes the character before the last character in each line in a file  
   c) Write a sed command that swaps the first and second words in each line in a file.

4.
   a)Pipe your/etc/passwd file to awk,and print out the home directory of each user.
b) Develop an interactive grep script that asks for a word and a file name and then tells how many lines contain that word.
c) Repeat
d) Part using awk

5.
a) Write a shell script that takes a command-line argument and reports on whether it is directory, a file, or something else.
b) Write a shell script that accepts one or more file name as argument and converts all of them to uppercase, provided they exist in the current directory.
c) Write a shell script that determines the period for which a specify user is working on the system.

6.
a) Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

7.
a) Write a shell script that computes the gross salary of an employee according to the following rules:
   i) If basic salary is <1500 then HRA=10% of the basic and DA=90% of the basic.
   ii) If basic salary is >=1500 then HRA=Rs500 and DA=98% of the basic.

   The basic salary is entered interactively through the keyboard.
b) Write a shell script that accepts two integers as its arguments and computes the value of first number raised to the power of the second number.

8.
a) Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
b) Write a shell script that takes a login name as command-line argument and reports when that person logs in.
c) Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

9.
a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
b) Develop an interactive script that asks for a word and a file name and then tells how many times that word occurred in the file.
c) Write a shell script to perform the following string operations:
   i) To extract a sub-string from a given string
   ii) To find the length of a given string.

10. Write a C program that takes one or more file or directory names as command line input and reports following information on the file:
1) File type 2) Number of links 3) Read, write and execute permissions 4) Time of last access
(Note: Use stat/fstat system calls)
11. Write C programs that simulate the following Unix commands:  
   a) mv  
   b) cp  
   (Use system calls)  
12. Implement in C the following Linux commands using system calls:  
   (a) cat  
   (b) ls  
   (c) mv  
13. Do the following Shell programs also  
1) Write a shell script to check whether a particular user has logged in or not. If he has logged in, also check whether he has eligibility to receive a message or not.  
2) Write a shell script to accept the name of the file from standard input and perform the following tests on it  
   a) file executable  
   b) file readable  
   c) File writeable  
   d) Both readable and writable  
3) Write a shell script while will display the user name and terminal name who login recently into the Unix system  
4) Write a shell script to find number of files in directory  
5) Write a shell script to check whether a given number is perfect or not  
6) Write a menu driven shell script to copy, edit, rename and delete a file  
7) Write a shell script for concatenation of two strings  
8) Write a shell script which will display fibonacci series up to a given number of argument  
9) Write a shell script to accept student number, name, marks in 5 subjects. Find total, average and grade. Display the result of the student and store in a file called stu.dat  
   Rules: avg >= 80 then grade A  
   Avg < 80 && avg >= 70 then grade B  
   Avg < 70 && avg >= 60 then grade C  
   Avg < 60 && avg >= 50 then grade D  
   Avg < 50 && avg >= 40 then grade E  
   Else grade F  
10) Write a shell script to accept empno, empname, basic. Find DA, HRA, TA, PF using following rules. Display empno, empname, basic, DA, HRA, TA, PF, GROSS SAL and NETSAL. Also store all details in a file called emp.dat  
   Rules: HRA is 18% of basic if basic > 5000 otherwise 550  
   DA is 35% of basic  
   PF is 13% of basic  
   IT is 14% of basic  
   TA is 10% of basic  
11) Write a shell script to demonstrate break and continue statements  
12) Write a shell script to satisfy the following menu options  
   a. Display current directory path  
   b. Display today's date  
   c. Display users who are connected to the Unix system  
   d. Quit  
13) Write a shell script to delete all files whose size is zero bytes from current directory  
14) Write a shell script to display string palindrome from given arguments  
15) Write a shell script which will display amstrong numbers from given arguments  
16) Write a shell script to display reverse numbers from given argument list
17) Write a shell script to display factorial value from given argument list
18) Write a shell script which will find maximum file size in the given argument list
19) Write a shell script which will greet you “Good Morning”, “Good Afternoon”, “Good evening” and “Good Night” according to current time
20) Write a shell script to sort the elements in an array using bubble sort technique
21) Write a shell script to find largest element in an array
22) Write an awk program to print sum, average of student marks list
23) Write an awk program to display students pass/fail report
24) Write an awk program to count the number of vowels in a given file
25) Write an awk program which will find maximum word and its length in the given input file
26) Write a shell script to generate the mathematical tables
27) Write a shell script to sort elements of given array by using selection sort
28) Write a shell script to search given number using binary search
29) Write a shell script to find number of vowels, consonants, numbers, white spaces and special characters in a given string
30) Write a shell script to lock the terminal.