**PRACTICAL PROGRAMS**

1. Write a menu driven program using functions, to find
2. **Factorial of a number**

The factorial of a number is the product of all the integers from 1 to that number.

**Example,** the factorial of 6 is 1\*2\*3\*4\*5\*6 = 720.

Factorial is not defined for negative numbers, and the factorial of zero is one, 0 != 1

1. **Fibonacci series up tp nth term**

It’s a unique sequence where the next number is the sum of the previous two numbers.  
Where the first two terms are always 0 and 1

The series Looks like : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …

1. **Whether a given number is Palindrome or not**

A palindrome is nothing but any number or a string which remains unaltered when reversed.

**Example:**12321  
**Output:**Yes, a Palindrome number

1. Write a menu driven program using functions, to find
2. **Whether a given number is Armstrong number or not**

A positive integer is called an Armstrong number of order n if

**abcd….. = an + bn + cn + dn …………**

In case of an Armstrong number of 3 digits, the sum of cubes of each digit is equal to the number itself.

**Example:** 153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3

153 is an Armstrong number

1. **To display Floyd's triangle**

Eg:

**Enter the number of Rows: 5**

**Floyd's Triangle:**

**1**

**2        3**

**4        5         6**

**7        8         9         10**

1. **12       13       14       15**
2. Write a menu driven program for the following:
3. Define a function MSEARCH(STATES) to display all the names from the list STATES, which are starting with alphabet M.

For eg. If list STATES contains

**L = ["MP", "UP", "DL", "UK", "KA", “MH”]**

The output should be:

**[“MP”, “MH”]**

1. A function with the following signature:

**remove\_letter(sentence, letter)**

This function should take a string & a letter as arguments, returning a copy

of that string with every instance of the indicated letter removed.

     For eg:

**remove\_letter("Hello there!", "e")**

     Should return the string **"Hllo thr!"**

1. Enter a string & count number of words in the given string.

**Suppose the string is:**

**“I am fine**”

Output should be

**Number of words = 3**

1. Write a menu driven program in Python using user defined functions that take a list as parameter and return (Do not use built-in functions):
2. Maximum of the element
3. Minimum of the element
4. Sum of the elements.
5. Write a menu driven program in Python using user defined functions for the following:
6. To read and display the text file content line by line with each word separated by “😊”.
7. To remove all the lines that contain the character ‘a’ in a file and write it to another file.
8. Write a menu driven program using function:
9. CreateTextFile( ) - create a text file "data.txt" with several lines of text
10. CopyVowelWord( ) - create another text file "vowel.txt" which will store all the words starting with vowel from "data.txt".
11. Read & display the contents of both the files.
12. Display the total number of words starting with vowel.
13. Write a menu driven program with the following functions:
14. CreateTextFile( ) - create a text file " content.txt" with few lines.
15. CountAll( ) - count number of lines, consonants, digits, spaces & words.
16. ReplaceSpace( ) - create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.
17. Read & display the contents of both the files.
18. Write a menu driven program with the following functions:
19. DISPLAYWORDS( ) to read lines from text file STORY.TXT , and display those words , which are less than 4 characters.
20. SEARCHWORD() to search a word and its frequency in a text file.
21. Write a menu driven program in Python using Pickle library and
22. Create a binary file “STUDENT.DAT” with following structure:

* Admission\_number
* Student Name
* Percentage

1. Write a function countrec( ) in Python that would read contents of the file “STUDENT.DAT” and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%.
2. Write a menu driven program in Python using Pickle library and
3. Create a binary file with following structure

* CompanyID
* Company name
* Turnover

1. Display the contents of the binary file.
2. Display the Company whose turnover is above user given value.
3. Search a Company by Company ID given by user.
4. Write a menu driven program in Python using Pickle library and
5. Create binary file with following structure

* Travelid
* From
* To

1. Append data to the file.
2. Update a record based on travelid.
3. Display the contents of the binary file.
4. Write a menu driven program in Python:
5. Define a function to write the following data into a CSV file

* Roll no
* Name of student
* Mark in Sub1
* Mark in sub2
* Mark in sub3
* Mark in sub4
* Mark in sub5

Perform following operations on the CSV file:

1. Define a function to read the CSV file and calculate total and percentage for each student.
2. Define a function to display the name of student if in any subject marks are greater than 80% (Assume marks are out of 100)
3. Write a menu driven program in Python:
4. Define a function to write the following data into a CSV file “emp.CSV”

* empid
* empName
* Salary
* Department

1. Define a function to search the record from “emp.CSV” file on the basis of empid. If record found display the record, otherwise display appropriate message.
2. A list contains the following record of a Hostel:

**[Hostel\_No, Total\_Students, Total Rooms]**

Write a menu driven program, with the following user defined functions to perform given operations on the stack named **‘Hostel’**:

1. Push\_element() - To push an object containing Hostel\_No and Total Students along with Total Rooms to the stack
2. Pop\_element() - To pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.

**For example:**

 If the lists of Hostel details are:

**[1, 2000, 1000]**

**[2, 1500, 800]**

**[3, 5000, 2000]**

The output should be:

**[3, 5000, 2000]**

**[2, 1500, 800]**

**[1, 2000, 1000]**

**Stack Empty**

1. Implement all the stack operations
2. Mr. John wants to store book details like book ID and Book Name in a dictionary format. Write a menu driven program, with separate user defined functions to perform the following operations:
3. Push the values (Name of the book) of the dictionary into a stack where book names begin with A or C.
4. Pop and display the content of the stack.

For example if the content of Dictionary is as follows:

**Book={"B001":"C++", "B002":"Python", "B003":"Ada", "B004":"C",**

**"B005":"Java", "B006":"Oracle", "B007":"HTMl"}**

The output of the program should be : **C Ada C++**

1. Implement all the stack operations

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*