**AMITY INTERNATIONAL SCHOOL**

**PRACTICAL LIST 2023-24**

**CLASS XII - COMPUTER SCIENCE**

**File Handling, Functions and Data Structures:**

1. Write a function to create a text file containing following data:

*Neither apple nor pine are in pineapple. Boxing rings are square.*

*Writers write, but fingers don’t fing. Overlook and oversee are opposites.*

*A house can burn up as it burns down. An alarm goes off by going on.*

1. Read back the entire file content using read() or readlines() and print it.
2. Append more text of your choice in the file and display the content of file with line numbers prefixed to line.
3. Display last line of file.
4. Display first line from 10th character onwards.
5. Read and display a line from the file. Ask user to provide the line no. to be read.
6. Find the frequency of words beginning with every letter i.e. (For the above example)

Words beginning with a: 5

Words beginning with n: 2

Words beginning with p: 2

Words beginning with o: 5 and so on

1. Assume that a text file named file1.txt contains some text, write a function named isvowel( ) that reads the file file1.txt and creates a new file named file2.txt, which shall contain only those words from the file file1.txt which don’t start with a vowel.

For example, if the file1.txt contains:

Carry Umbrella and Overcoat When it Rains

Then the file file2.txt shall contain

Carry When Rains

1. A file containing data about a collection of students has the following format. Each line contains a first name, a second name, a registration number, no of years and a department separated by tabs.

Rajat Sen 12345 1 CSEE

Jagat Narain 13467 3 CSEE

Anu Sharma 11756 2 Biology

Sumita Trikha 23451 4 Biology

Sumder Kumra 11234 3 MME

Kanti Bhushan 23211 3 CSEE

1. Write a Python program that will copy the contents of the file into a list of tuples
2. Display full details of the student sorted by registration number
3. The names of all students with no of year less than 3
4. The number of people in each department
5. Write a program that reads myfile.txt, and builds a histogram (a dictionary having key value pair as word: occurrence) of the words in the file.
6. Now use histogram to print :

i) Total number of words

ii) Number of different words

iii) The most common words

b) Using above text file, myfile.txt,

1. Write a program that maps a list of words read from the file to an integer representing the length of the corresponding words (Use a dictionary having key value pair as length : list of word).

Now using this dictionary,

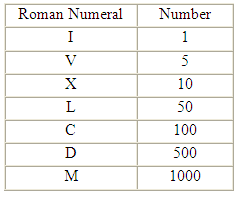
1. Design a function find\_longest\_word() to display a list of longest words from file.
2. Define a function filter\_long\_words(n) that takes an integer n and returns the list of words that are longer than n from file.
3. A dictionary Customer contains the following keys: roomno, name, duration. A binary file ‘hotel.dat’ contains details of customer checked in the hotel. Write a program to perform the following using pickle module:
4. Read n dictionary objects and load them into the file
5. Read all the dictionary objects from the file and print them
6. Counts the number of customers present in the hotel. (Counts the total number of customers present in the hotel.(Assume that file might have few record before adding n records in part (a))
7. Display those customers from the file, who have stayed more than 2 days in the hotel.
8. Sun Microsystems held a recruitment test. The file, placement.csv, contains the below format of data: The marks are from 5 different tests conducted and each col is out of 5 marks:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SNO | NAME | MARKS1 | MARKS2 | MARKS3 | MARKS4 | MARKS5 |
| 1 | JOHN | 4 | 3 | 4 | 2 | 5 |
| 2 | PETER | 3 | 4 | 4 | 3 | 5 |

1. Read the above file and print the data.
2. Write the UDF to find total no. of people who came for the placement test.
3. Write the UDF to find the top n Names on basis of total Marks.
4. Write a program to input a number and then call the functions
5. count(n) which returns the number of digits
6. reverse(n) which returns the reverse of a number
7. hasdigit(n) which returns True if the number has a digit else False
8. show(n) to show the number in its expanded form (sum of place values of the digits in n) (E.g. 124 = 100 + 20 + 4)
9. A Number is a perfect number if the sum of all the factors of the number (including 1) excluding itself is equal to number. (E.g. 6 = 1+2+3 and 28=1+2+4+7+14). Number is a prime number if its factors are 1 and itself. Write functions:
10. Generatefactors() to populate a list of factors
11. isPrimeNo() to check whether the number is prime number or not
12. isPerfectNo() to check whether the number is perfect number or not

Save the above as a module perfect.py and use in the program main.py as a menu driven program.

1. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.



For e.g., 2 is written as II in Roman numeral, just two one's added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II. Roman numerals are usually written largest to smallest from left to right.

However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five, we subtract it making four. The same principle applies to the number nine, which is written as IX.

There are six instances where subtraction is used:

* I can be placed before V (5) and X (10) to make 4 and 9.
* X can be placed before L (50) and C (100) to make 40 and 90.
* C can be placed before D (500) and M (1000) to make 400 and 900.

Write a UDF which takes a string (Roman Numeral) as an argument and returns the integer equivalent.

def romanToInt(s):

return Ans

print(romanToInt(‘LVIII’)) #Should print 58

print(romanToInt(‘MCMXCIV’)) #Should print 1994

1. Data can be represented in memory in different ways Binary, Decimal, Octal, and Hexadecimal. Input number in decimal and desired type (Specify B for Binary, O for Octal, H for Hexadecimal) for output. Write a function to perform the conversions:

SAMPLE INPUT: 12

DESIRED TYPE: B

RESULT: 1100

SAMPLE INPUT: 25

DESIRED TYPE: O

RESULT: 31