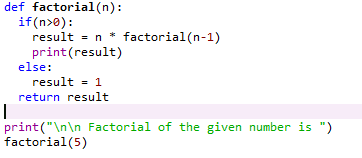
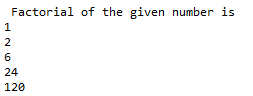
|  |  |
| --- | --- |
| **18CSC207J** | **ADVANCED PROGRAMMING PRACTICE** |
| **S.No** | **Paradigm** |
| 1 | Procedural |
| 2 | Structural |
| 3 | OOPS |
| 4 | Declarative |
| 5 | Imperative |
| 6 | Event driven & GUI |
| 7 | Functional |
| 8 | Parallel |
| 9 | Concurrent |
| 10 | Dependent Type |
| 11 | Logic |
| 12 | Network |
| 13 | Automata Based |
| 14 | Symbolic |

1. PROCEDURAL PROGRAMMING PARADIGM
   1. Find the factorial of the given number

**SOURCE CODE**

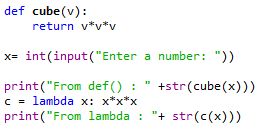


**OUTPUT**



* 1. Write the python program the calculate cube of a number by using def () and lambda () function.

**SOURCE CODE**

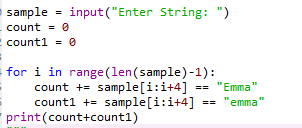
****

**Output**

****

* 1. Return the number of times that the string “Emma” appears anywhere in the given string.

**Source Code:**

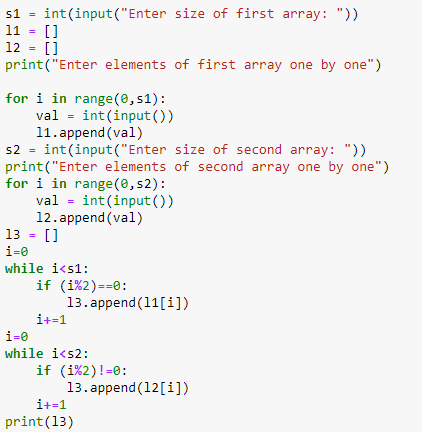
****

**Output**

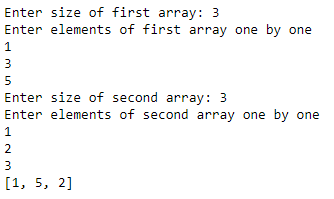


1. STRUCTURAL PROGRAMMING PARADIGM
2. Given a two list of integers. Create a third list such that should contain only odd numbers from the first list and even numbers from the second list

**Source Code:**

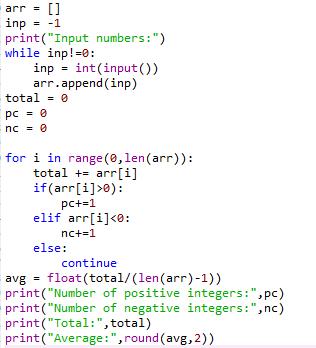
****

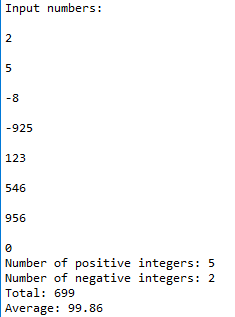
**Output**



1. Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input 0. Display the average as a floating-point number.

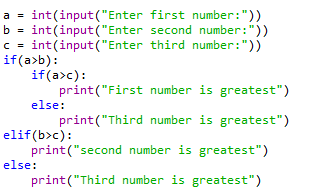
**Source Code:**

****

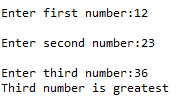
**Output**

1. Find the greatest of three numbers

SOURCE CODE

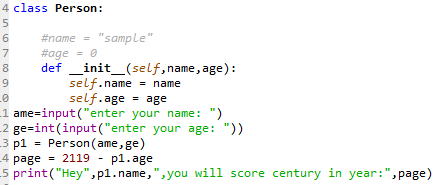


OUTPUT

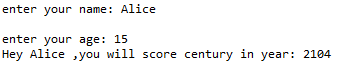


1. OBJECT ORIENTED PROGRAMMING PARADIGM
2. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

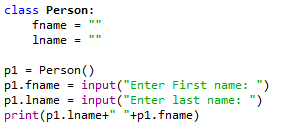
**Source Code:**

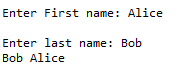


**Output**



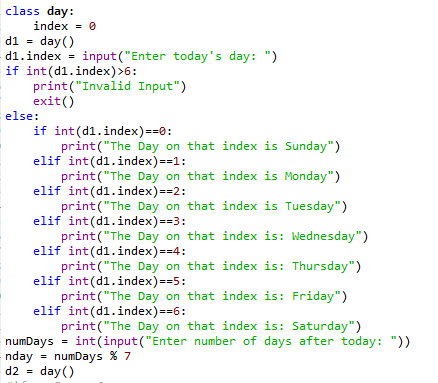
1. Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them.

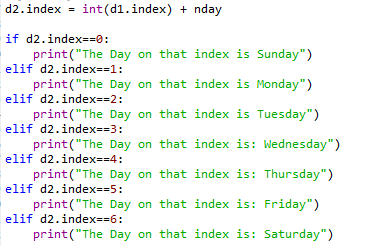
**Source Code:**

**Output**

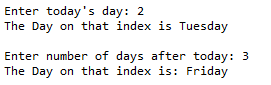
1. Write a program that prompts the user to enter an integer for today’s day of the week (Sunday is 0, Monday is 1, ..., and Saturday is 6). Also prompt the user to enter the number of days after today for a future day and display the future day of the week.

**Source Code:**

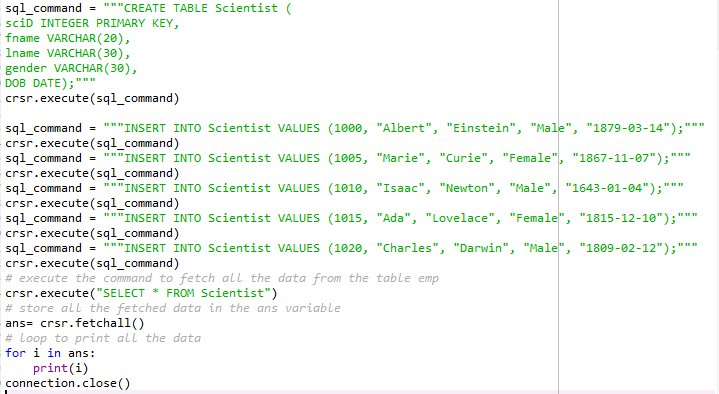
****

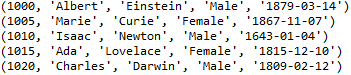
****

**Output**

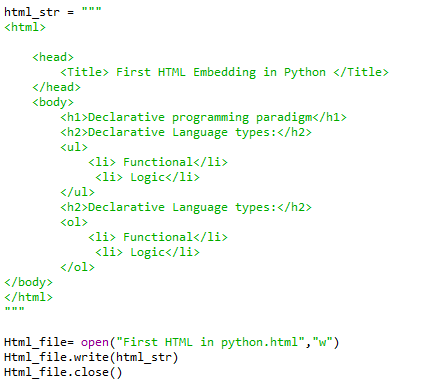


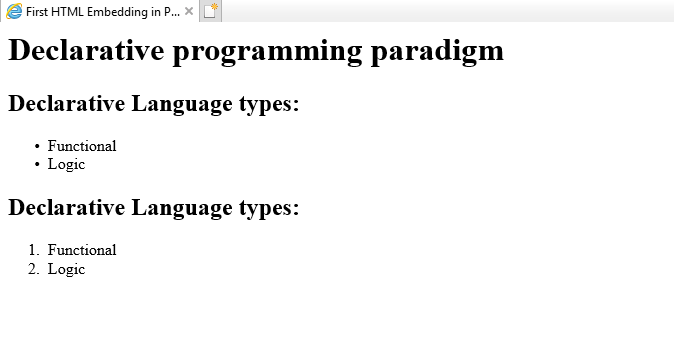
1. DECLARATIVE PROGRAMMING PARADIGM
2. Write a program to create a table for storing the details of few scientists with the columns Scientist ID, First name, Last name, Gender and Date of Birth. Enter at least 5 members’ details and display the same





1. Give a python code which is embedded with some HTML tag





1. Create a student database which collect name, gender, email ID, PCM marks.

SOURCE CODE

1. 

OUTPUT



1. IMPERATIVE PROGRAMMING PARADIGM
2. Print the following pattern

1

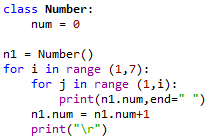
2 2

3 3 3

4 4 4 4

5 5 5 5 5

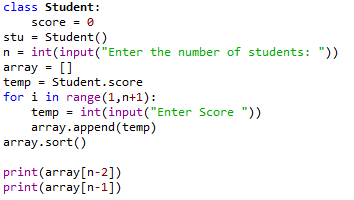
**Source Code:**



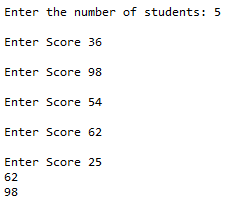
**Output**

1. Write a program that prompts the user to enter the number of students and each student’s score, and displays the highest and second highest scores.

**Source Code:**

****

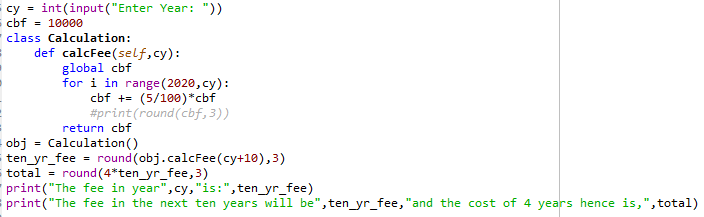
**Output**

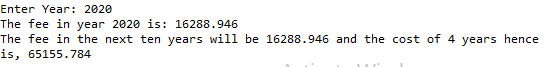


C. Suppose that the tuition for a university is $10,000 this year and increases 5% every year. Write a program that computes the tuition in ten years and the total cost of four years’ worth of tuition starting ten years from now.

#Calculation of this problem is done by the assumption that the present year is #2020 and the user is providing input for any year in the future, hence the tuition fee for 2020 is #assumed to be $10,000 and will be incremented in subsequent years

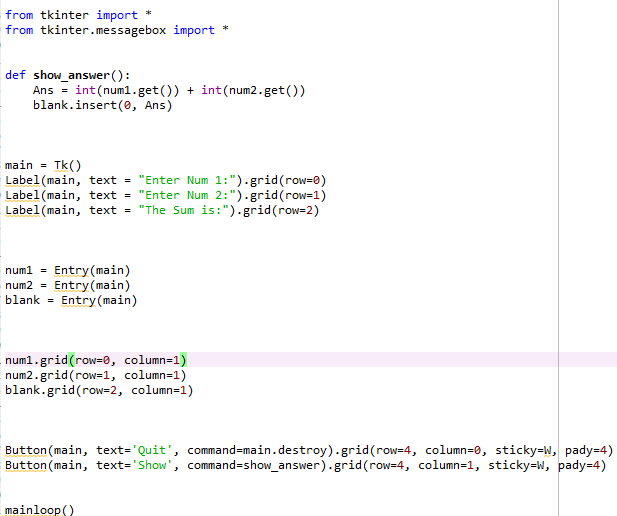
**Source Code:**

****

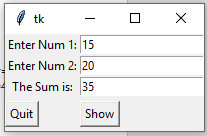
**Output**

1. EVENT DRIVEN PROGRAMMING PARADIGM and GRAPHICAL USER INTERFACE PROGRAMMING PARADIGM
   1. Give a simple program using GUI and Event handling methods to implement a simple calculator

SOURCE CODE

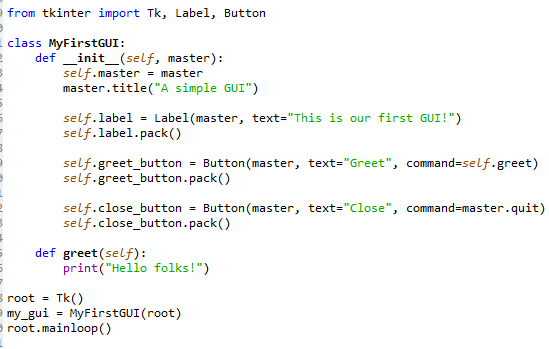


OUTPUT

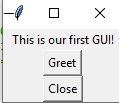


* 1. Give a simple program using GUI and Event handling methods to Greet the users

SOURCE CODE

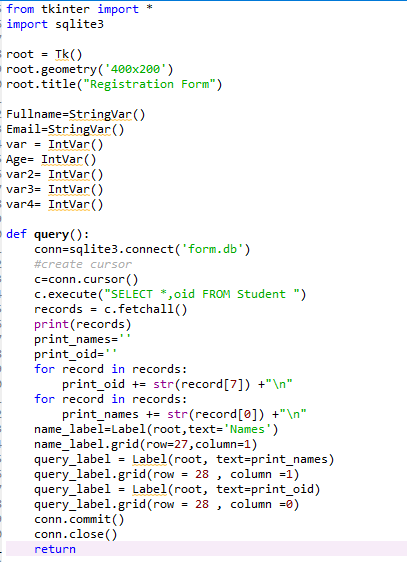


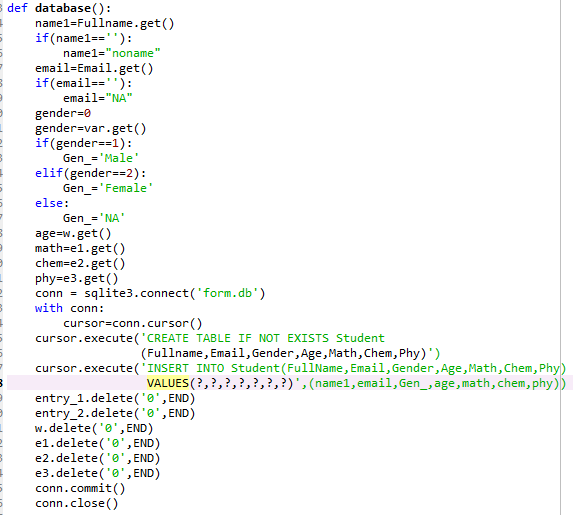
OUTPUT

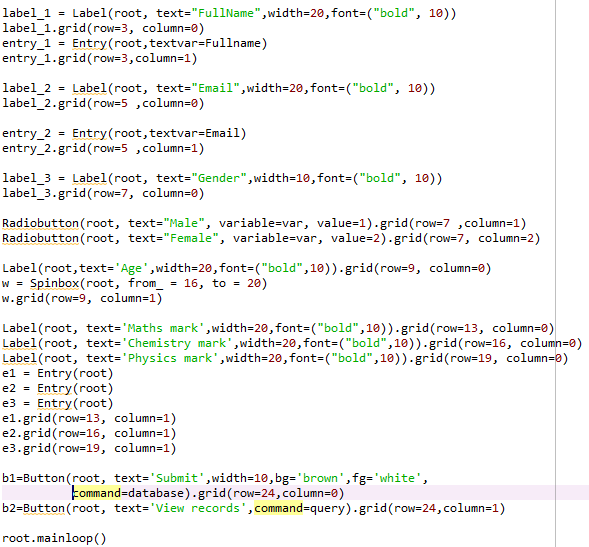




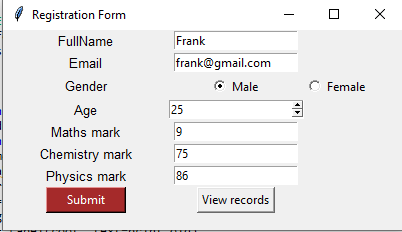
* 1. Create a registration form to collect name, email address, gender, PCM marks. Store and retrieve from database.







OUTPUT

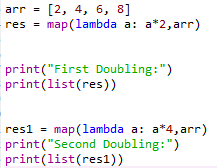


When “View records” button is clicked

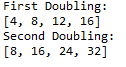


1. FUNCTIONAL PROGRAMMING PARADIGM
2. For the given input numbers = {2, 4, 6, 8}, use map () function to double all the input numbers and double it again using python program

**Source Code:**

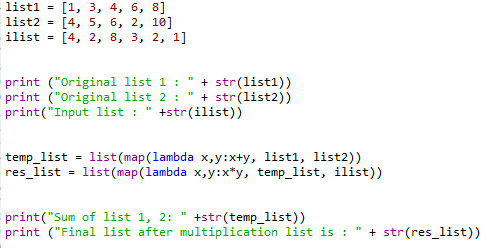


**Output**

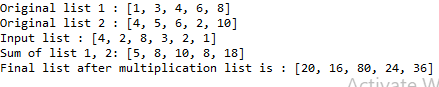


1. Write the python program to add two lists of input [1,2,3,4,5,6] and [4,6,12,3,2,1] using map and lambda and then perform multiplication for the resultant value off add with the input list [4,2,8,3,2,1]

**Source Code:**

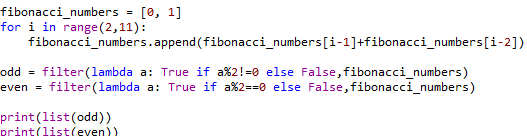


**Output**



1. Write the python program to calculate the sequence of the first 11 Fibonacci numbers and filter out first the odd and then the even elements from the sequence of the first 11 Fibonacci numbers.

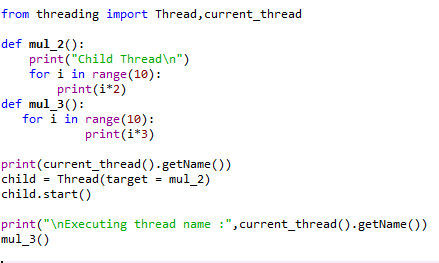
**Source Code:**

****

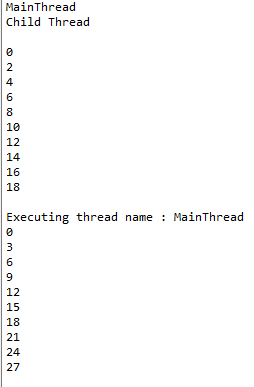
**OUTPUT**

1. PARALLEL PROGRAMMING PARADIGM
2. Create a program using threading concept which uses target method.

SOURCE CODE

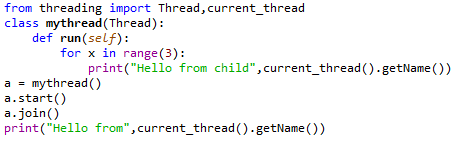


**OUTPUT**

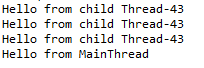


1. Create a program using threading concept which uses run method.

SOURCE CODE

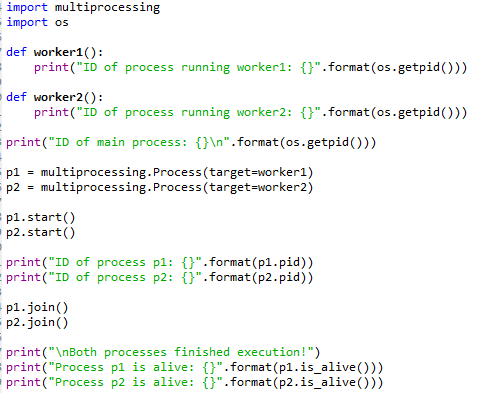


**OUTPUT**

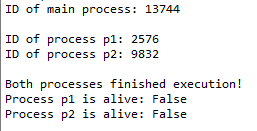


1. Create a program using threading concept which uses Process method.

SOURCE CODE

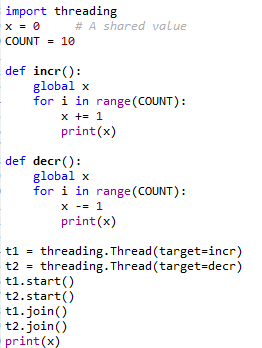


OUTPUT



1. CONCURRENT PROGRAMMING PARADIGM
2. A python code for implementing the race condition

**SOURCE CODE**

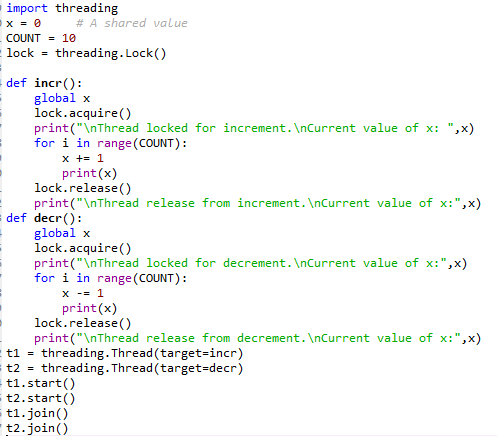


**OUTPUT**

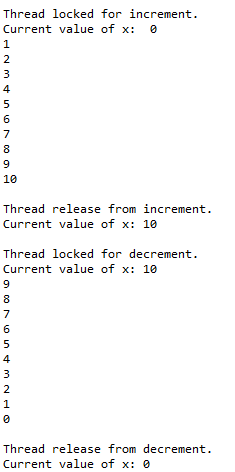


1. Give a program which emphasis on synchronizing in python using Lock

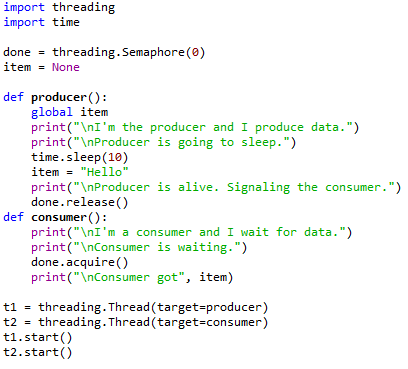
**SOURCE CODE**



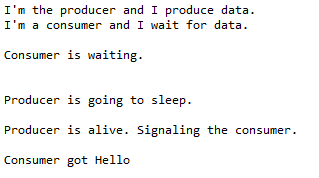
**OUTPUT**



1. Write a program to implement Semaphore

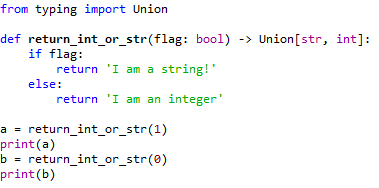


**OUTPUT**



1. DEPENDENT TYPE PROGRAMMING PARADIGM
   1. Give an example using Union in dependent type programming

SOURCE CODE

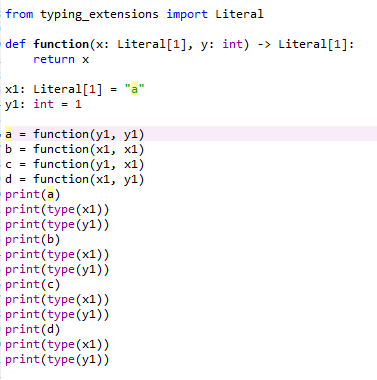


OUTPUT

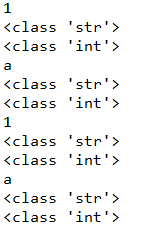


* 1. Give an example using Literal in dependent type programming

SOURCE CODE

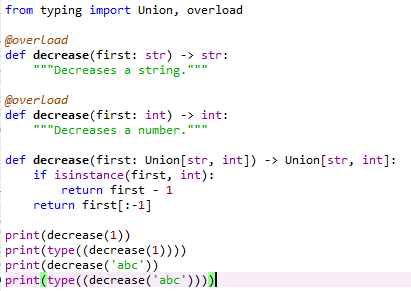


OUTPUT

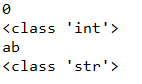


* 1. Give an example using overload in dependent type programming

SOURCE CODE

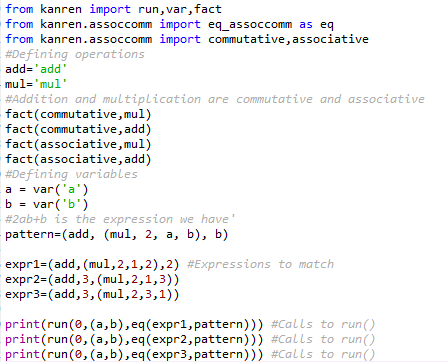


OUTPUT



1. LOGIC PROGRAMMING PARADIGM
2. Write a python code for matching the mathematical expression

**SOURCE CODE**

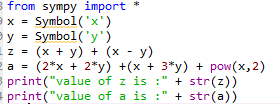


**OUTPUT**



1. Write a python code for evaluation of expression

SOURCE CODE

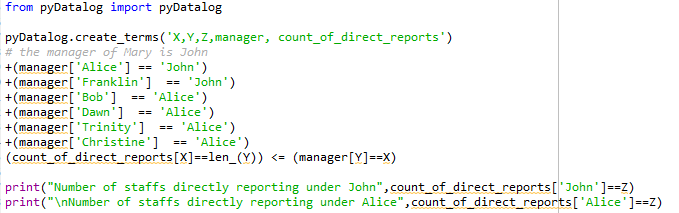


**OUTPUT**

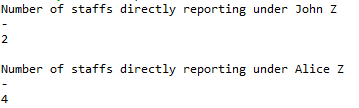
****

1. If Alice and Franklin are the Team Leaders under the Project Head of John. Bob, Dawn, Trinity and Christine are the Team members of Alice. Find the number of staffs directly reporting to Alice and John.

SOURCE CODE



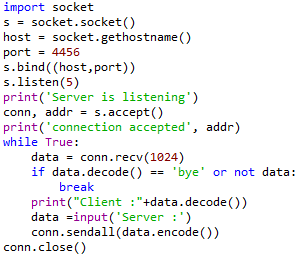
**OUTPUT**

****

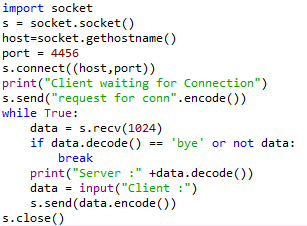
1. NETWORK PROGRAMMING PARADIGM
2. Create a TCP Client Server Chat application

SOURCE CODE

SERVER

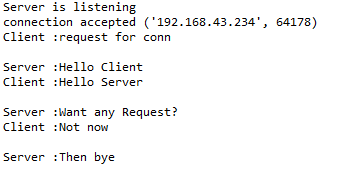


CLIENT

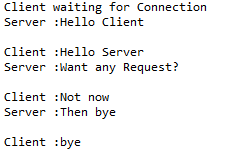


OUTPUT

SERVER



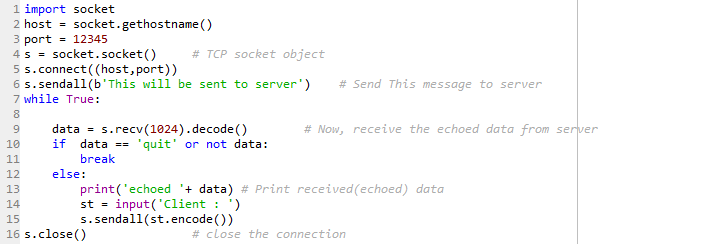
CLIENT



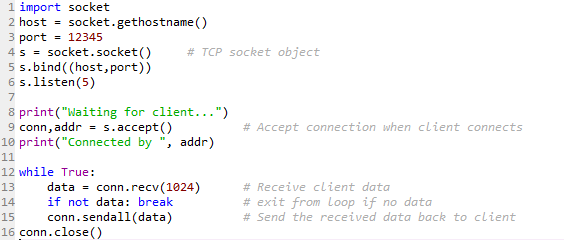
1. Implement a Echo Client server Application

SOURCE CODE

SERVER

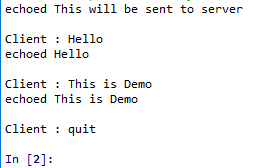


CLIENT

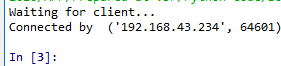


OUTPUT

CLIENT



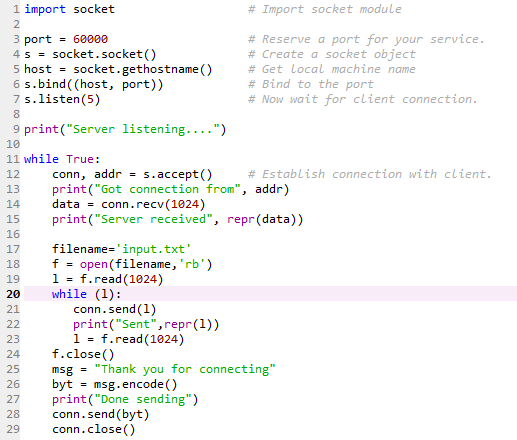
SERVER



1. Implement a FTP client server application

SOURCE CODE

SERVER

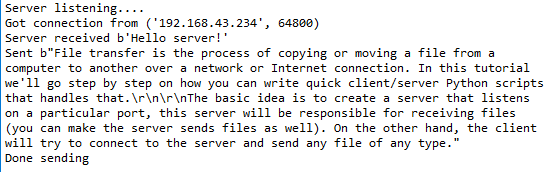


CLIENT

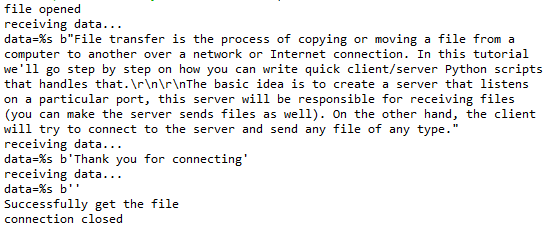


OUTPUT

SERVER

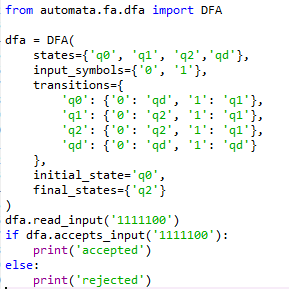


CLIENT



1. AUTOMATA BASED PROGRAMMING PARADIGM
2. Implement Deterministic Finite Automata which accepts strings over {0,1}. Which starts with 1 and ends with 0

SOURCE CODE

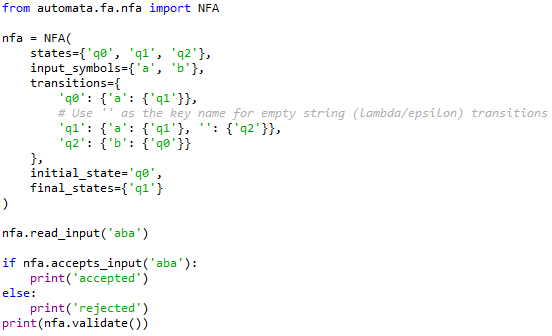


OUTPUT



1. Implement NFA which matches strings beginning with 'a', ending with 'a', and containing no consecutive 'b's

SOURCE CODE

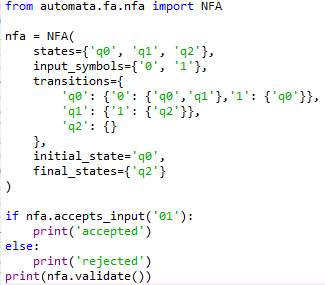


OUTPUT



1. Implement NFA which recognizes the language of strings that end with 01

SOURCE CODE

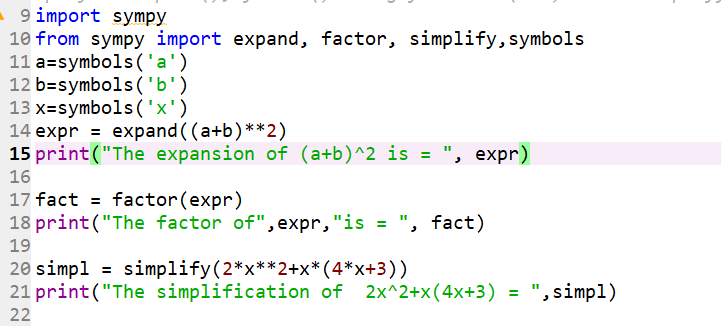


OUTPUT

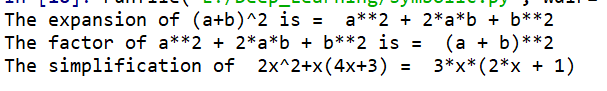


1. GRAPHICAL USER INTERFACE PROGRAMMING PARADIGM
2. SYMBOLIC PROGRAMMING PARADIGM
   1. Perform expand(), factor() using function (a+b)^2 and simplify() using 2x^2+x(4x+3) in Python

SOURCE CODE

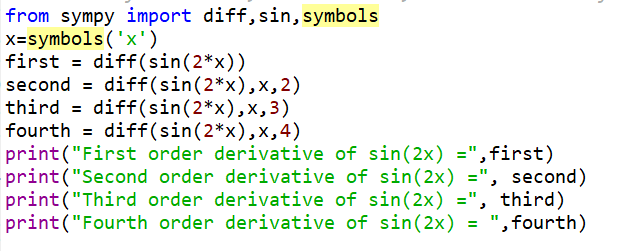


OUTPUT

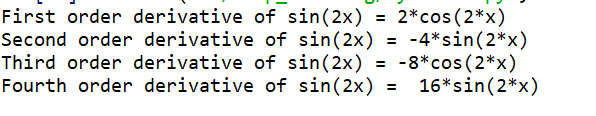


* 1. determine first-order, second-order, third-order and fourth order derivative of the given function - sin(2x^3-9x) using Python

SOURCE CODE

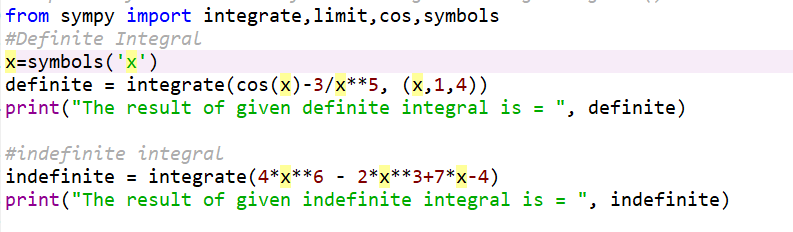


OUTPUT



* 1. Compute definite and indefinite Integrals using integrate() using python

SOURCE CODE



OUTPUT

