**M.C.A Semester – II**

|  |  |
| --- | --- |
| **Roll No :** | **1233** |
| **Name of Student :** | **Kirti vichare** |
| **Subject :** | MCAL21 Artificial Intelligence and Machine Learning Lab |
| **Module:** | Module2: Introduction to Python Programming |
| **Date:** | 12/05/2022 |

**Q. 1)** Write a Python program to display Pythagorean triplet between numbers 1 to 100.

e.g. if a\*a+b\*b=c\*c then (a,b,c) is Pythagorean triplet.

**Source code:**

**def pythagoreanTriplets(limits) :**

**c, m = 0, 2**

**while c < limits :**

**# Now loop on n from 1 to m-1**

**for n in range(1, m) :**

**a = m \* m - n \* n**

**b = 2 \* m \* n**

**c = m \* m + n \* n**

**if c > limits :**

**break**

**print(a, b, c)**

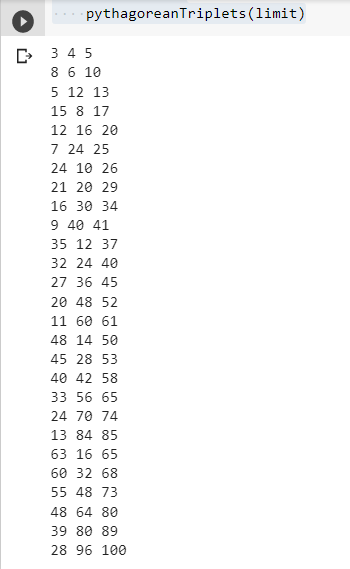
**m = m + 1**

**if \_\_name\_\_ == '\_\_main\_\_' :**

**limit = 100**

**pythagoreanTriplets(limit)**

**Output :**

****

**Q. 2)** Write a Python program to demonstrate set operations for the following sets

A={“beena”, 78, “shreya”, 87, 90,35}

B={“radha”,87,”yamini”,90,46,76}

C={89,90,”sudha”}

Perform the following operations

1. Add elements{“pooja”,80} in the set A
2. Find the union of A and B set
3. Find common elements of set B and C
4. Find the difference between B and C
5. Store all elements in set A, B, C set D and display set D

**Source code:**

**A = {"beena",78,"shreya",87,90,35}**

**B = {"radha",87,"yamini",90,46,76}**

**C = {89,90,"sudha"}**

**D={}**

**A.add("pooja")**

**A.add(80)**

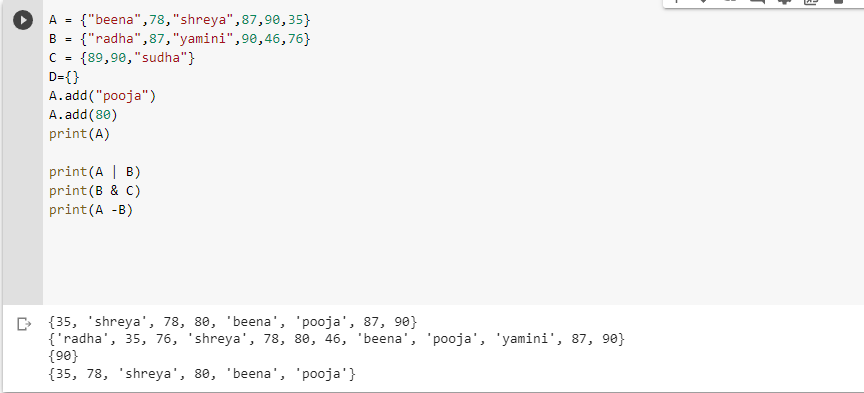
**print(A)**

**print(A | B)**

**print(B & C)**

**print(A -B)**

**Output :**

****

**Q. 3)** Write a python program to implement tower of Hanoi game.

**Source code:**

**def TowerOfHanoi(n , source, destination, auxiliary):**

**if n==1:**

**print ("Move disk 1 from source",source,"to destination",destination)**

**return**

**TowerOfHanoi(n-1, source, auxiliary, destination)**

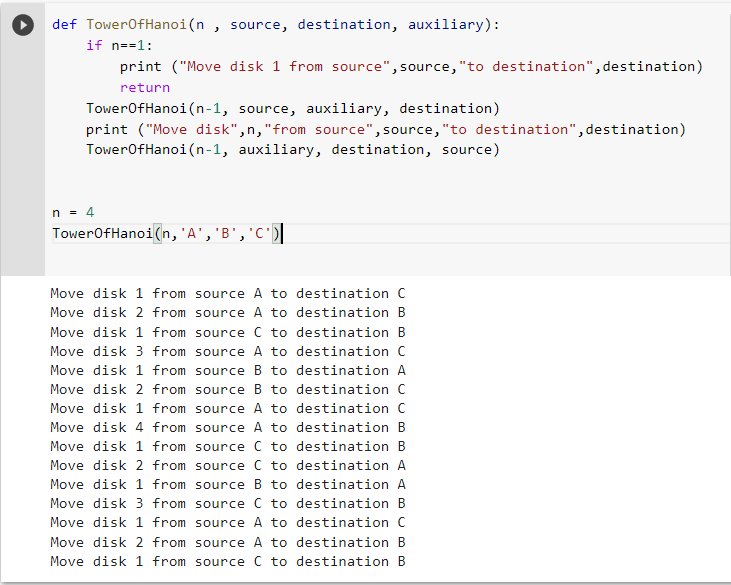
**print ("Move disk",n,"from source",source,"to destination",destination)**

**TowerOfHanoi(n-1, auxiliary, destination, source)**

**n = 4**

**TowerOfHanoi(n,'A','B','C')**

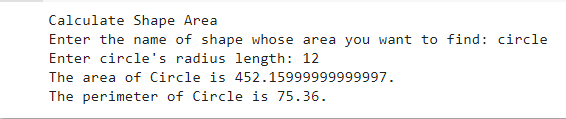
**Output :**

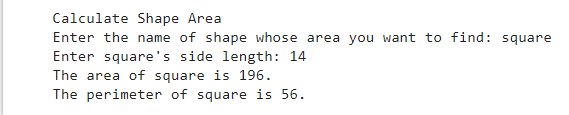
****

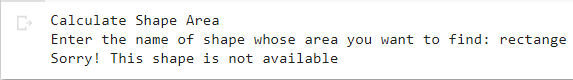
**Q. 4)** Write a menu driven program to calculate perimeter and area of circle, triangle, rectangle, trapezoid. (using lambda function)

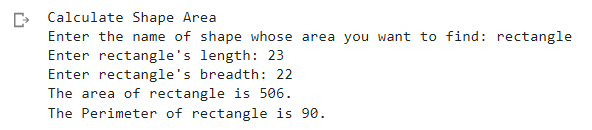
**Source code:**

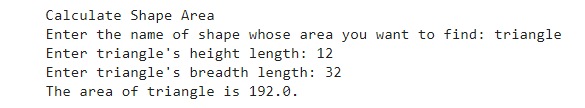
**Output :**

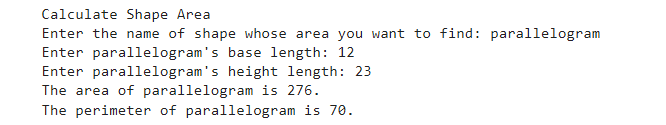
****

****

****

****





**Q. 5)** Write a Python program to display binary, octal and hexa decimal equivalent of given decimal number.

e.g. Decimal number = 35

Binary = 100011

Octal= 43

Hexadecimal= 23

**Source code:**

**def decimal\_into\_binary(decimal\_1):**

**decimal = int(decimal\_1)**

**print ("The given decimal number", decimal, "in Binary number is: ", bin(decimal))**

**def decimal\_into\_octal(decimal\_1):**

**decimal = int(decimal\_1)**

**print ("The given decimal number", decimal, "in Octal number is: ", oct(decimal))**

**def decimal\_into\_hexadecimal(decimal\_1):**

**decimal = int(decimal\_1)**

**print ("The given decimal number", decimal, " in Hexadecimal number is: ", hex(decimal))**

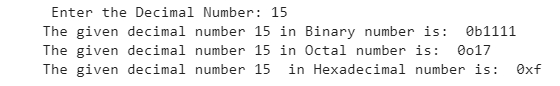
**decimal\_1 = int (input (" Enter the Decimal Number: "))**

**decimal\_into\_binary(decimal\_1)**

**decimal\_into\_octal(decimal\_1)**

**decimal\_into\_hexadecimal(decimal\_1)**

**Output :**

****

**Q. 6)** Write a menu driven program to check number is Prime number, Perfect number, Armstrong number. (user defined functions function. i.e. isPrime(n), isPerfect(n), isArmstrong(n))

**Source code:**

**num = int(input("Enter Number: "))**

**def Armstrong(n,o):**

**sum = 0**

**temp = n**

**while temp > 0:**

**digit = temp % 10**

**sum += digit \*\* o**

**temp = temp//10**

**if n == sum:**

**print(n,"is an Armstrong number")**

**else:**

**print(n,"is not an Armstrong number")**

**order = len(str(num))**

**Armstrong(num,order)**

**#prime number**

**def checkPrime(x):**

**for i in range(2, x):**

**if n%i==0:**

**return 1**

**print("Enter a Number: ", end="")**

**n = int(input())**

**p = checkPrime(n)**

**if p==1:**

**print(str(n)+ " is not a Prime Number")**

**else:**

**print(str(n)+ " is a Prime Number")**

**#perfect number**

**def perfect\_numbers(N):**

**sum = 0**

**for i in range(1,N):**

**if(N%i == 0):**

**sum = sum+i**

**return sum**

**N = int(input("Enter a number : "))**

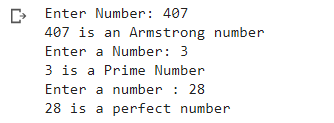
**if(N == perfect\_numbers(N)):**

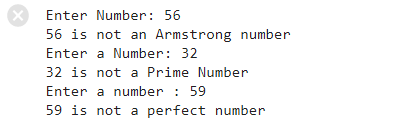
**print(N, "is a perfect number")**

**else:**

**print(N, "is not a perfect number")**

**Output :**

****

****

**Q. 7)** Design a mini game using Python. Write a program in which the computer randomly chooses a number between 1 to 10, 1 to 100, or any range. Then give users a hint to guess the number. Every time the user guesses wrong, he gets another clue, and his score gets reduced. The clue can be multiples, divisible, greater or smaller, or a combination of all.

**Source code:**

**import random**

**import math**

**lower = int(input("Enter Lower bound:- "))**

**upper = int(input("Enter Upper bound:- "))**

**x = random.randint(lower, upper)**

**print("\n\tYou've only ",**

**round(math.log(upper - lower + 1, 2)),**

**" chances to guess the integer!\n")**

**count = 0**

**while count < math.log(upper - lower + 1, 2):**

**count += 1**

**guess = int(input("Guess a number:- "))**

**if x == guess:**

**print("Congratulations you did it in ",**

**count, " try")**

**break**

**elif x > guess:**

**print("You guessed too small!")**

**elif x < guess:**

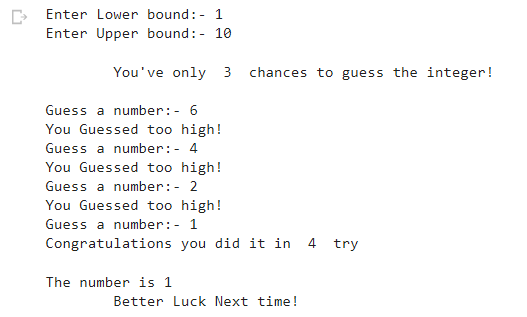
**print("You Guessed too high!")**

**if count >= math.log(upper - lower + 1, 2):**

**print("\nThe number is %d" % x)**

**print("\tBetter Luck Next time!")**

**Output:**

****

**Q. 8)** Design a Rock Paper Scissors in Python.

If you’re unfamiliar, rock paper scissors is a hand game for two or more players. Participants say “rock, paper, scissors” and then simultaneously form their hands into the shape of a rock (a fist), a piece of paper (palm facing downward), or a pair of scissors (two fingers extended). The rules are straightforward:

Rock smashes scissors.

Paper covers rock.

Scissors cut paper.

Now that you have the rules down, you can start thinking about how they might translate to Python code.

**Source code:**

**import random**

**a="rock"**

**b="paper"**

**c="scissor"**

**abc=[a,b,c]**

**comp= random. choice(abc)**

**player=input("Enter your choice: ")**

**if comp=="rock" and player=="paper":**

**print("HURRAY!PLAYER WINS")**

**elif comp=="rock" and player=="scissor":**

**print("HURRAY!COMPUTER WINS")**

**elif comp=="paper" and player=="rock":**

**print("HURRAY!COMPUTER WINS")**

**elif comp == "paper" and player == "scissor":**

**print("HURRAY!PLAYER WINS")**

**elif comp == "scissor" and player == "rock":**

**print("HURRAY!PLAYER WINS")**

**elif comp == "scissor " and player == "paper":**

**print("HURRAY!COMPUTER WINS")**

**elif comp==player:**

**print("TIE")**

**Output:**

****

****

****

**Q. 9)** Design a FLAMES is a popular game named after the acronym: Friends, Lovers, Affectionate, Marriage, Enemies, Sibling. This game does not accurately predict whether or not an individual is right for you, but it can be fun to play this with your friends.

Get the count:

Take the two names.

Remove the common characters with their respective common occurrences.

Get the count of the characters that are left.

**Source code:**

**def remove\_match\_char(list1, list2):**

**for i in range(len(list1)) :**

**for j in range(len(list2)) :**

**if list1[i] == list2[j] :**

**c = list1[i]**

**list1.remove(c)**

**list2.remove(c)**

**list3 = list1 + ["\*"] + list2**

**return [list3, True]**

**list3 = list1 + ["\*"] + list2**

**return [list3, False]**

**if \_\_name\_\_ == "\_\_main\_\_" :**

**p1 = input("Player 1 name : ")**

**p1 = p1.lower()**

**p1.replace(" ", "")**

**p1\_list = list(p1)**

**p2 = input("Player 2 name : ")**

**p2 = p2.lower()**

**p2.replace(" ", "")**

**p2\_list = list(p2)**

**proceed = True**

**while proceed :**

**ret\_list = remove\_match\_char(p1\_list, p2\_list)**

**con\_list = ret\_list[0]**

**proceed = ret\_list[1]**

**star\_index = con\_list.index("\*")**

**p1\_list = con\_list[ : star\_index]**

**p2\_list = con\_list[star\_index + 1 : ]**

**count = len(p1\_list) + len(p2\_list)**

**result = ["Friends", "Love", "Affection", "Marriage", "Enemy", "Siblings"]**

**while len(result) > 1 :**

**split\_index = (count % len(result) - 1)**

**if split\_index >= 0 :**

**right = result[split\_index + 1 : ]**

**left = result[ : split\_index]**

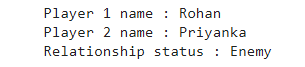
**result = right + left**

**else :**

**result = result[ : len(result) - 1]**

**print("Relationship status :", result[0])**

**Output:**

****

**Q. 10)** Write a Python program to encrypt and decrypt password using Caesarcipher.

e.g. Password: ATTACK

Shift: 4

Encrypted password: EXXEGO

**Source code:**

**def encypt\_func(txt, s):**

**result = ""**

**for i in range(len(txt)):**

**char = txt[i]**

**if (char.isupper()):**

**result += chr((ord(char) + s - 64) % 26 + 65)**

**else:**

**result += chr((ord(char) + s - 96) % 26 + 97)**

**return result**

**txt = "ARTIFICIAL INTELLIGENCE"**

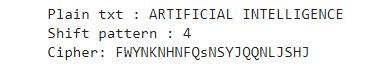
**s = 4**

**print("Plain txt : " + txt)**

**print("Shift pattern : " + str(s))**

**print("Cipher: " + encypt\_func(txt, s))**

**Output :**

****

**🞛🞛🞛🞛🞛🞛🞛🞛🞛🞛🞛**