**PYTHON ACTIVITY 8:**

**1)The surface of the Earth is curved, and the distance between degrees of longitude varies with latitude. As a result, finding the distance between two points on the surface of the Earth is more complicated than simply using the Pythagorean theorem. Let (t 1 , g 1 ) and (t 2 , g 2 ) be the latitude and longitude of two points on the Earth’s**

**surface. The distance between these points, following the surface of the Earth, in kilometers is:**

**distance = 6371.01 × arccos(sin(t 1 ) × sin(t 2 ) + cos(t 1 ) × cos(t 2 ) × cos(g 1 − g 2 ))**

**HINT:The value 6371.01 in the previous equation wasn’t selected at random. It is the average radius of the Earth in kilometers.**

**Create a program that allows the user to enter the latitude and longitude of two**

**points on the Earth in degrees. Your program should display the distance between the points, following the surface of the earth, in kilometers.**

**• 2)The horoscopes commonly reported in newspapers use the position of the sun at the**

**• time of one’s birth to try and predict the future. This system of astrology divides the**

**• year into twelve zodiac signs, as outline in the table below:**

**• Zodiac sign Date range**

**• Capricorn December 22 to January 19**

**• Aquarius January 20 to February 18**

**• Pisces February 19 to March 20**

**• Aries March 21 to April 19**

**• Taurus April 20 to May 20**

**• Gemini May 21 to June 20**

**• Cancer June 21 to July 22**

**• Leo July 23 to August 22**

**• Virgo August 23 to September 22**

**• Libra September 23 to October 22**

**• Scorpio October 23 to November 21**

**• Sagittarius November 22 to December 21**

**•**

**• Write a program that asks the user to enter his or her month and day of birth. Then your program should report the user’s zodiac sign as part of an appropriate output message.**

**•**

**• 3)Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years.**

**• The rules for determining whether or not a year is a leap year follow:**

**• i)Any year that is divisible by 400 is a leap year.**

**• ii)Of the remaining years, any year that is divisible by 100 is not a leap year.**

**• iii)Of the remaining years, any year that is divisible by 4 is a leap year.**

**• iv)All other years are not leap years.**

**• Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.**

**•**

**• A prime number is an integer greater than 1 that is only divisible by one and itself. Write a function that determines whether or not its parameter is prime, returning True if it is, and False otherwise. Write a main program that reads an integer from the user and displays a message indicating whether or not it is prime. Ensure that the main program will not run if the file containing your solution is imported into another program.**

**•**

**• Write a program that reads integers from the user and stores them in a list. Your program should continue reading values until the user enters 0. Then it should display all of the values entered by the user (except for the 0) in order from smallest to largest, with one value appearing on each line. Use either the sort method or the sorted function to sort the list.**

**•**

**• A proper divisor of a positive integer, n, is a positive integer less than n which divides evenly into n. Write a function that computes all of the proper divisors of a positive integer. The integer will be passed to the function as its only parameter. The function will return a list containing all of the proper divisors as its only result. Complete this exercise by writing a main program that demonstrates the function by reading a value from the user and displaying the list of its proper divisors. Ensure that your main program only runs when your solution has not been imported into another file.**

**•**

**• 7)A standard deck of playing cards contains 52 cards. Each card has one of four suits along with a value. The suits are normally spades, hearts, diamonds and clubs while the values are 2 through 10, Jack, Queen, King and Ace.**

**• Each playing card can be represented using two characters. The first character is the value of the card, with the values 2 through 9 being represented directly. The characters “T”, “J”, “Q”, “K” and “A” are used to represent the values 10, Jack, Queen, King and Ace respectively. The second character is used to represent the suit of the card. It is normally a lowercase letter: “s” for spades, “h” for hearts, “d” for diamonds and “c” for clubs. The following table provides several examples of cards and their two-character representations.**

**• Card Abbreviation**

**• Jack of spades Js**

**• Two of clubs 2c**

**• Ten of diamonds Td**

**• Ace of hearts Ah**

**• Nine of spades 9s**

**• Begin by writing a function named createDeck. It will use loops to create a**

**• complete deck of cards by storing the two-character abbreviations for all 52 cards into a list. Return the list of cards as the function’s only result. Your function will not take any parameters.**

**• Write a second function named shuffle that randomizes the order of the cards**

**• in a list. One technique that can be used to shuffle the cards is to visit each element in the list and swap it with another random element in the list. You must write your own loop for shuffling the cards. You cannot make use of Python’s built-in shuffle function. Use both of the functions described in the previous paragraphs to create a main program that displays a deck of cards before and after it has been shuffled. Ensure that your main program only runs when your functions have not been imported into another file.**

**•**

**• 8)Write a function that determines whether or not a list of values is in sorted order (either ascending or descending). The function should return True if the list is already sorted. Otherwise it should return False. Write a main program that reads a list of numbers from the user and then uses your function to report whether or not the list is sorted.**

**•**

**• 9)Two words are anagrams if they contain all of the same letters, but in a different order. For example, “evil” and “live” are anagrams because each contains one e, one i, one l, and one v. Create a program that reads two strings from the user, determines whether or not they are anagrams, and reports the result.**