Practice these lab tasks

Let us take an integer from user as input and check whether the given value is even or not. If the given value is not even then it means that it will be odd. So here we need to use if-else statement an demonstrated below

Write a Python code to keep accepting integer values from user until 0 is entered. Display sum of the given values.

Write a Python code to accept an integer value from user and check that whether the given value isprime number or not.

Accept 5 integer values from user and display their sum. Draw flowchart Before coding in python.

Calculate the sum of all the values between 0-10 using while loop.

Take input from the keyboard and use it in your program.

Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. (**Hint:** remember to use the user input lessons from the very first exercise)
Extras: Keep the game going until the user types “exit”.
Keep track of how many guesses the user has taken, and when the game ends, print this out.

Write a program that prompts the user to input an integer and then outputs the number with the digitsreversed. For example, if the input is 12345, the output should be 54321.

Write a program that reads a set of integers, and then prints the sum of the even and odd integers.

Create a fibonacci series
Extra: Fibonacci series is that when you add the previous two numbers the next number is formed.
You have to start from 0 and 1.
E.g. 0+1=1 → 1+1=2 → 1+2=3 → 2+3=5 → 3+5=8 → 5+8=13
So the series becomes
0 1 1 2 3 5 8 13 21 34 55 ……
Steps: You have to take an input number that shows how many terms to be displayed. Then use loops for displaying the Fibonacci series up to that term e.g. input no is =6 the output should be
0 1 1 2 3 5

Write a Python code to accept marks of a student from 1-100 and display the grade according to the following formula.

Grade F if marks are less than 50
Grade E if marks are between 50 to
60 Grade D if marks are between 61
to 70 Grade C if marks are between
71 to 80 Grade B if marks are
between 81 to 90 Grade A if marks
are between 91 to 100

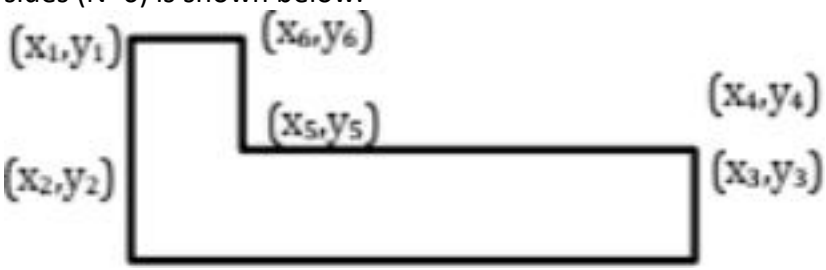
Write a program that takes a number from user and calculate the factorial of that number.

Accept two lists from user and display their join.

Write a function in python that receives a string and returns True if that string is a palindrome and False otherwise. Remember that difference between upper and lower case characters are ignored during this determination.

Write a python code that finds another matrix/2D list that is a product of and b, i.e., C=a*b
Imagine two matrices given in the form of 2D lists as under; a = [[1, 0, 0], [0, 1, 0], [0, 0, 1]] b = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

A closed polygon with N sides can be represented as a list of tuples of N connected coordinates, i.e.,[(x1,y1), (x2,y2), (x3,y3), . . . , (xN,yN)]. A sample polygon with 6 sides (N=6) is shown below.



Write a python function that takes a list of N tuples as input and returns the perimeter of the polygon. Remember that your code should work for any value of N.
Hint: A perimeter is the sum of all sides of a polygon.

Imagine two sets A and B containing numbers. Without using built-in set functionalities, write your own function that receives two such sets and returns another set C which is a symmetric difference of the two input sets. (A symmetric difference between A and B will return a set C which contains only those items that appear in one of A or B. Any items that appear in both sets are not included in C). Now compare the output of your function with the following builtin functions/operators.

A.symmetric_difference(B)
B.symmetric_difference(A)
A ^ B
B ^ A

Create a Python program that contains a dictionary of names and phone numbers. Use a tuple of separate first and last name values for the key field. Initialize the dictionary with at least three names and numbers. Ask the user to search for a phone number by entering a first and last name. Display the matching number if found, or a message if not found.

Create two lists based on the user values. Merge both the lists and display in sorted order.

Repeat the above activity to find the smallest and largest element of the list. (Suppose all the elements are integer values)

For this exercise, you will keep track of when our friend's birthdays are, and be able to find that information based on their name. Create a dictionary (in your file) of names and birthdays.
When you run your program it should ask the user to enter a name, and return the birthday of that person back to them. The interaction should look something like this:
>>> Welcome to the birthday dictionary. We know the birthdays of:
Albert Einstein
Benjamin
FranklinAda
Lovelace41
>>> Who's birthday do you want to look up?
Benjamin Franklin
>>> Benjamin Franklin's birthday is 01/17/1706.

Create a dictionary by extracting the keys from a given dictionary
Write a Python program to create a new dictionary by extracting the mentioned keys from the below dictionary.
Given dictionary:
sample_dict = {
"name": "Kelly",
"age": 25,
"salary": 8000,
"city": "New york"}
# Keys to extract
keys = ["name", "salary"]
Expected output:
{'name': 'Kelly', 'salary': 8000}