LAB: Artificial Intelligence

Lab Instructor	Ms. Saba Aslam
Department	Computer Science

Python Lab-1: Python Basics to Advance

Instructions

1. Make a word document paste your code and output there.
2. Comments in the code explaining chunks of the code is important.
3. Plagiarism is strictly prohibited; 0 marks would be given to students who cheat.
4. Late submissions are not allowed.
5. There are two modules of this manual, in the first module you will explore some python environments and in the second module you will make some of the basic programs in python.
6. For the code part make sure your code is error-free and running. Failed to compile code will lead to zero marks in that question.
7. For coding part you can use Jupiter Notebook or Pycharm
8. **For programming questions, you can use any built-in functions to perform the task.**
9. Increase the readability of your code by writing meaningful names of functions and variables.

Non-Coding Part

Task 1

Define Anaconda and Pip in your own words package managers in your own words?

Task 2

PyCharm is an IDE mostly used for compiling python code. How we can install a python packages in PyCharm?

Note: Screen short is required for both manual process and installation using terminal, you can install any two packages using different processes

Task 3

Difference between Module, Library and API in perspective of python programming language?

Task 4

Can a variable number of arguments be passed to a function? Create a prototype function that takes variable number of arguments in python. For instance, print() function in python can take as many arguments as you want.

Hint:

- You are required to provide a prototype function that can take any number of arguments. E.g. print("1"), print("1", "2"), print("1", "2", "3"), print("1", "2", "3", "4") and so on.
- Your function should be as generic that it can accept arguments from 0 to infinity.

Task 5

I. Provide two ways of compiling code in PyCharm?
   Note: You can write shortcut for manual process and screenshot for code compilation using terminal

II. Check the installed version of python on your machine and add the screenshot of it to the answer. Also create a python program to display python version on console. [Hint: You may just print the version after checking it by storing the version info as a string]

Coding Part

Task 6 (Personal Information)

Write a program that displays the following information:

• Your name

• Your address, with city, state, and ZIP

• Your telephone number

• Your university and major degree

Task 7 (Day calculator)

Write a Python program to calculate the number of days between two dates.
Sample date: (2023, 1, 1), (2023, 1, 30)
Expected output: 30 days
Sample date: (2022, 1, 1), (2023, 1, 30)
Expected output: 394 days

Task 8 (Intersection of sets)
Write a python program to find the interaction of sets. Consider the sample data below Test Data :
color_list_1 = set(["White", "Black", "Red"])
color_list_2 = set(["Red", "Green"]) Expected
Output :
{'Black', 'White'}
Note: Your program should work for all sorts of input sets, it should not only work for provided sets

Task 9 (Conditional operations)
Write a Python program to sum three given integers. However, if two values are equal, the sum will be zero.

Task 10 (User input)
Write a Python program to input two integers on a single line

Task 11 (Intersection of sets) list = ["apple", "cherry", "orange", "kiwi", "melon", "mango"] i. Remove “cherry” and “melon” from list.
ii. Add “banana” at second last index.

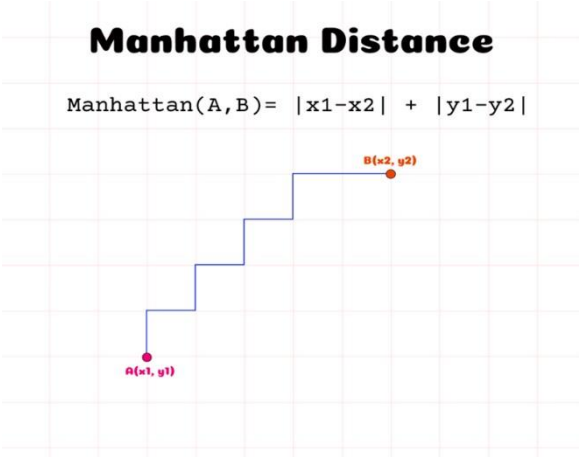
Task 12 (Restaurant bill calculator)
Write a program that calculates the total amount of a meal purchased at a restaurant. The program should ask the user to enter the charge for the food, then calculate the amounts of a 18 percent tip and 7 percent sales tax. Display each of these amounts and the total.

Task 13 (Sales Prediction)
A company has determined that its annual profit is typically 23 percent of total sales. Write a program that asks the user to enter the projected amount of total sales, then displays the profit that will be made from that amount. Hint: Use the value 0.23 to represent 23 percent.

Task 14 (Valid Number Information)
Design a program that uses a loop to build a list named valid numbers that contains only the numbers between 0 and 100 from the numbers list below. The program should then determine and display the total and average of the values in the valid_numbers list. numbers = [74, 19, 105, 20, −2, 67, 77, 124, −45, 38]

Task 15(Sum of Digits in a String)
Write a program that asks the user to enter a series of single-digit numbers with nothing separating them. The program should display the sum of all the single digit numbers in the string. For example, if the user enters 2514, the method should return 12, which is the sum of 2, 5, 1, and 4

Task 16 (Distance calculator)
Design a python program that calculates the **Manhattan distance** between two points. The user shall enter the coordinates of two points and your program should compute the distance between points. Note: You should create a separate function named *compute_distance* that will return control to your main function.
Given images illustrate the distance between two points using Manhattan distance

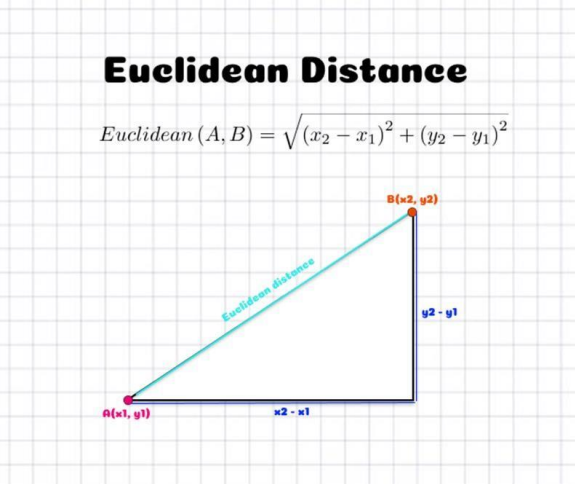


Task 17 (Distance calculator)

Design a python program that calculates the **Euclidian distance** between two points. The user shall enter the coordinates of two points and your program should compute the distance between points.

Note: You should create a separate function named *compute_distance* that will return control to your main function.

Given images illustrate the distance between two points using Euclidian distance



Task 17 (quadratic equation solver)

You are required to design a quadratic equation solver. You can prompt the user for the required parameters and in the end, you should display roots for the given equation.

Hint:

Recall all your concepts about quadratic equations.

Prompt user for values of a,b, and c.

You should also verify that given parameters form a quadratic equation.

Task 18 (dry run)

Write the dry run of the following code fragments

```
def main():
    num = 10
    halver(num)

def halver(number):
    print(number)
    half = number / 2
    if half >= 1:
        halver(half)
main()
```

```
def main():
    word = 'test'
    show_me(word)

def show_me(word):
    print(word)
    new_word = word[1:]
    if len(new_word) > 0:
        show_me(new_word)
main()
```

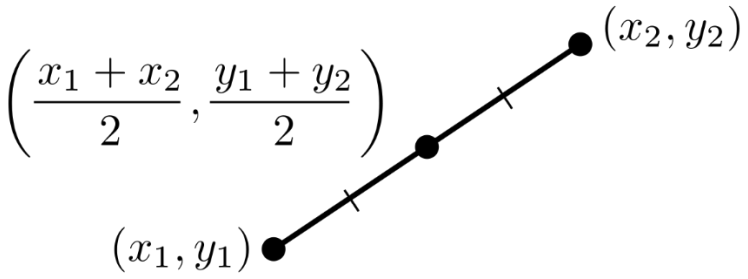
Task 19 (vegetable shop automation)

Write a program that keeps vegetable names and prices in a dictionary as key-value pairs. The program should display a menu that lets the user see a list of all vegetables and their prices, add a new vegetable and price, change the price of an existing vegetable, and delete an existing vegetable and price. The program should pickle the dictionary and save it to a file when the user exits the program. Each time the program starts, it should retrieve the dictionary from the file and unpickle it.

Task 20 (Mid points in line)

Write a Python program to calculate the midpoints of a line. You can prompt user for required parameters.

Hint: Following image briefly explain how mid points of line can be determined

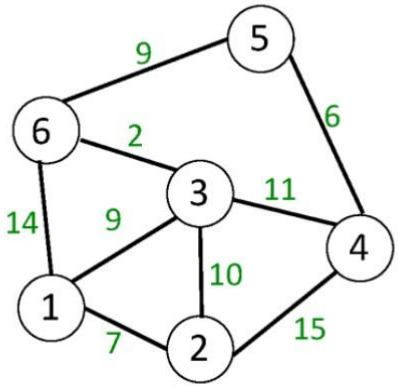


Task 21 (Date printer)

Write a program that reads a string from the user containing a date in the form mm/dd/yyyy. It should print the date in the format March 12, 2018

Task 22 (Graphs)

Implement the given graph using adjacency matrix



Task 22 (trivia game)

Create a simple trivia game for two players. The program will work like this:

- Starting with player 1, each player gets a turn at answering 5 trivia questions. (There should be a total of 10 questions.) When a question is displayed, 4 possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer, he or she earns a point.

• After answers have been selected for all the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

Task 23 (*RetailItem* Class)

Write a class named *RetailItem* that holds data about an item in a retail store. The class should store the following data in attributes: item description, units in inventory, and price. Once you have written the class, write a program that creates three *RetailItem* objects and stores the following data in them:

	Description	**Units in Inventory**	**Price**
Item#1	Jacket	12	59.95
Item#2	Designer Jeans	40	34.95
Item#3	Shirt	20	24.95

Task 2*4* (Statistical analysis)

Write a program that takes 10 values from the user and performs statistical analysis on data by computing mean, median and mode.

Task 2*4* (Magic date)

The date June 10, 1960, is special because when it is written in the following format, the month times the day equals the year:

6/10/60

Design a program that asks the user to enter a month (in numeric form), a day, and a two digit year. The program should then determine whether the month times the day equals the year. If so, it should display a message saying the date is magic. Otherwise, it should display a message saying the date is not magic.

Task 2*5* (Lambda Functions)

Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and prints the result.

Sample Output:

Input: 10

Output: 25

Note: You are only required to use lambda in this task

Task 2*6* (Lambda Functions)
Write a Python program to filter a list of integers using Lambda. The
original list of integers:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Even numbers from the said list:
[2, 4, 6, 8, 10]
Odd numbers from the said list:
[1, 3, 5, 7, 9]
Note: You are only required to use lambda in this task

Task 2*7* (Encryption & Description of messages)
Write a python program to encrypt and decrypt messages. You can follow following algorithm for encryption of messages:

Algorithm:

1. Take key value from user as
2. For encryption shift the message bit to the right by provided key 3. For decryption shift the encrypted message to left by provided key Example:

Key = 2
A → C (encryption, shifting A to right by 2) ; C → A (decryption, shifting C to left by 2)
Z → B (encryption, shifting Z to right by 2) ; B → Z (decryption, shifting B to left by 2)

Task 2*7* (XOR Logic gate)
Write a python program to take XOR operation of two binary bits provided by user

Task 2*9* (Inverse of matrix)
Write a python program to determine the inverse of a matrix $A^{-1} = \frac{1}{|A|} \text{adj } A$
Note:
You can use any built-in library for this task
For now, the order of the matrix should be 2 x 2

Task 30 (Capital Quiz)

Write a program that creates a dictionary containing the U.S. states as keys, and their capitals as values. (Use the Internet to get a list of the states and their capitals.) The program should then randomly quiz the user by displaying the name of a state and asking the user to enter that state's capital. The program should keep a count of the number of correct and incorrect responses. (As an alternative to the U.S. states, the program can use the names of countries and their capitals.)