

In the name of God

Project1

- Write your code in an ipynb file so we can see your codes and your results together.
- If you want to answer a question like 8.b, you can easily use a markdown cell and write your answer.

1) An auto repair shop charges as follows. Inspecting the vehicle costs \$75. If no work needs to be done, there is no further charge. Otherwise, the charge is \$75 per hour for labor plus the cost of parts, with a minimum charge of \$120. If any work is done, there is no charge for inspecting the vehicle.

Generate a **data frame** of 5 samples about repair shop then write a program to read values for hours worked and cost of parts (either of which could be 0) and print the charge for the job.

2) Write a program to read two pairs of integers (use **readline**). Each pair represents a fraction. For example, pair **3 5** represent the fraction $\frac{3}{5}$. Your program should print the sum of the given fractions. For example, given the pairs **3 5** and **2 3**, your program should print **19/15**.

Modify the program so that it prints the sum with the fraction reduced to a proper fraction; for this example, your program should print $1\frac{4}{15}$.

3) A country charges income tax as follows based on one's gross salary. No tax is charged on the first 20% of salary. The remaining 80% is called taxable income. Tax is paid as follows:

- 10% on the first \$15,000 of taxable income;
- 20% on the next \$20,000 of taxable income;
- 25% on all taxable income in excess of \$35,000;

Write a program to **read** a value for a person's salary and print the amount of tax to be paid. Also print the average tax rate, that is, the percentage of salary that is paid in tax. For example, on a salary of \$20,000, a person pays \$1700 in tax. The average tax rate is $1700/20000 \times 100 = 8.5\%$.

4) The manager of a hotel wants to calculate the cost of carpeting the rooms in the hotel. All the rooms are rectangular in shape. You should generate a **dataframe** that contains data for 400 rooms. The first column contains the number of rooms, n. The second and third columns contain the **length** and **breadth** of the room (in meters), and the fourth column includes the cost per square meter of the carpet for that room. For example, a row like **325 3.0 4.5 40.00** means that room 325 is 3.0 meters by 4.5 meters, and the cost of the carpet for that room is \$40.00 per square meter. Write a program to read the data and do the following:

- Produce another **dataframe** containing the room number, the area of the room, and the cost of the carpet for the room.
- Print the total cost of carpeting all the rooms.
- Print the number of the room which will cost the most to the carpet (ignore ties).

5) Generate an array that contains 5000 numbers (each number is one of the numbers from 1 to 9). A number can appear zero or more times and can appear anywhere in the array. Write a program to print the number which appears the most in **consecutive** positions and the number of times it appears.

6) Write a function **isPerfectSquare** which, given an integer n, returns true if n is a perfect square (e.g. 25, 81) and false if it is not. Use only elementary arithmetic operations.

7) You are required to generate a sequence of integers from a given positive integer n , as follows. If n is even, divide it by 2. If n is odd, multiply it by 3 and add 1. Repeat this process with the new value of n , stopping when $n = 1$. Write a function that, given n , returns the **length and sequence** generated. For example, if n is 13, the function should generate:

10

13 40 20 10 5 16 8 4 2 1

Then write a program to read two integers m and n ($m < n$), and print the **maximum sequence length** for the numbers between m and n , inclusive. Also, print the number which gives the maximum length. For example, if $m = 1$ and $n = 10$, your program should print this:

9 generates the longest sequence of length 20

8) First, consider a square of side 1 centered at the origin and a circle of radius 1 inscribed into the square. Draw two random numbers x and y from a uniform distribution between $[0,1)$ and calculate its norm. Count the number of pairs (x, y) fall inside and outside the circle. You'll see that a good approximation for π is achieved after you draw many pairs (x, y) using the expression **$\pi = 4 \cdot \text{inside} / \text{total}$** . Second, explain why this code gives you π . Third, use PyPlot or any other plotting package to show the random numbers filling the circle and square (use different colors if inside or outside the circle) as you draw random pairs (x, y) .

9) Write a code that read an integer number in range 1 to 100 from you and your machine will try to guess the number as below:

For example, we choose 14. At first, it asks you whether your number is less than 50 or not, Then you should answer the question with yes or no. In our example, we answer yes. This time your machine questions whether your number is less than 25 or not and you should answer it. This process will go on just **3 times** and if your machine can guess truly print **Successful** otherwise print **Failed**.

Good luck