ROBOTICS – Python Development – Lab 2 Homework

1. Considering the following electricity bill calculation rules:

UNIT	PRICE
First 100 units	5.0 EUR per unit
Next 100 units	7.5 EUR per unit
After 200 units	10.0 EUR per unit

Please calculate the price of 500 electricity units.

---

2. We need to perform payment processing and we need to scrutinize very large payment amounts. Considering a list of payments to be done, we need create two lists:

- If a payment is below or equal to a certain value it will be added to the standard payments list;
- If a payment is above a certain value it will be added to the large payments list.

Please implement this requirement for a list of your choice, considering that standard payments are all payments that are below the three quarters of the maximum value of all payments in the list.

---

3. We need to implement a quality monitoring mechanism that will detect if the total quantity of impurities in a batch of product samples exceeds a certain limit.

Please implement the following:

a) Simulate a product sample using a dictionary using two keys "sample_mass" and "impurity_mass" which are float values;

b) Simulate a product batch of 100 products with various sample_mass and impurity_mass values. The sample_mass is between 50 and 100 and the impurity_mass is between 5 and 15. Impurity mass cannot be more than 20% of sample_mass for each sample;
c) Create a function that explores the product batch in reverse "sample_mass" order and stops as soon as the total weight of impurities exceeds 10% of the total sample mass of products explored so far, otherwise continue until the end of samples batch.
d) Return a list of the products explored, the values of the total sample_mass and total impurity_mass along with the ratio of total impurity_mass and total sample_mass.

Use a dictionary to pack this information.

To simulate random numbers you can use the following code:

import random

x = random.random()

The random function return a float value between 0 and 1.

---

4. Consider a robot used to collect objects from an assembly line, objects needed to weigh less than 10 kg (configurable). The robot can handle the following commands:

a) **start** - starts the robot, the first command accepted by the robot;
b) **move** - moves the assembly line to the next object;
c) **evaluate** - evaluates the object on the assembly line, reading its weight;
d) **collect** - collects the current object from the assembly line. Only objects weighting less or equal than maximum allowable weight can be collected from the assembly line. If the robot attempts to collect objects weighing more than this maximum allowable weight, an error is generated and the robot enters a forced stop condition;
e) **mark** - marks the current object from the assembly line as too heavy and will not pick it up;
f) **stop** - stops the robot and releases the robot's control (the robot will not accept any more commands).

Please implement the following:

a) Generate a list of objects on the assembly line, objects weighing between 5 and 15 kg. There should be at least 10 objects on the assembly line;
b) Write a function that receives the list of objects from the assembly line and generates a list of commands for the robot (use a list of strings containing commands as a list) for collecting all the appropriate objects from the assembly line and stamping all the objects it cannot collect.
The robot most not enter a forced stop condition;
c) Write several tests by calling the function with various assembly line objects values and verify that the generated commands are equal to the expected ones.