# Lab 1

A).

1. Source code:

lab1_1.py > ...

```python
#lab1_1.py - Displays a name in two different lines - Jhonatan Parada Torres

name = 'Jhonatan'
last_name = 'Parada'

print(f'{name}\n{last_name}')
```

Output:

```
@jhonatanparada499 ➜/workspaces/Jhonatan_LAB1 (main) $ /home/codespace/.python/current/bin/python3
Jhonatan
Parada
```

2. Source code:

lab1_2.py > ...

```python
#lab1_2.py - Displays a Functional and Customizable Receipt - Jhonatan Parada Torres
#Customization

tab = "\t\t" #The space or distance between the columns
line_divider = ["-----","-----"] #The characters for the lines between rows
header = ["Items","Price"] #The titles displayed at the top of each column(This model only supports 2 columns)

#Items

final_price = 0
items = [
    header,
    line_divider,
    ["Apple",1.75],
    ["Banana",2.25],
    ["Cherry",3.5],
    #Add more items here following the format list, such as ["your item", "the price of it"],
    line_divider,
    ["Total",1]
]

#Calcaulation and Display

for item in items:
    #Identifies the prices of the items
    item_price = item[1] if item[1] and isinstance(item[1],(int,float)) else 0

    if item_price:
        if item != items[-1]:
            #Adds the prices of the items to the final price if it hasn't reached the last element of the list
            final_price += item_price
        else:
            #The final price is assigned to the last element of the receipt, being "Total" in this case
            items[-1][1] = final_price

        #Concatenates the '$' symbol to all the numbers of the receipt
        item[1] = f"${item[1]}"

    print(*item,sep=tab)
```

Output:

```
@jhonatanparada499 ➜/workspaces/Jhonatan_LAB1 (main) $ /home/codespace/.python/current/bin/python3
atan_LAB1/lab1_2.py
Items           Price
-----           -----
Apple           $1.75
Banana          $2.25
Cherry          $3.5
-----           -----
Total           $7.5
```

3.Source code:

lab1_3.py

```python
#lab1_3.py - Displays a quote - Jhonatan Parada Torres

print(
    'Albert Einstein once said,\n'
    '"A person who never made a mistake\n'
    'never tried anything new."'
)
```

Output:

```
@jhonatanparada499 ➜/workspaces/Jhonatan_LAB1 (main) $ /home/codespace/.python/current/bin/python3
Albert Einstein once said,
"A person who never made a mistake
never tried anything new."
```

4.Source code:

lab1_4.py > ...

```python
#lab1_4.py - Discounted price - Jhonatan Parada Torres

price = 99.99
discountPersent = 25
markdown = discountPersent / price * 100
price = price - markdown

print(f"Price = {round(price,2)}")
```

Output:

```
@jhonatanparada499 ➜/workspaces/Jhonatan_LAB1 (main) $ /home/codespace/.python/current/bin/python3
Price = 74.99
```

5. Source code:

lab1_5.py > ...

```python
#lab1_5.py - Miles per Galon a car averaged between two fillings - Jhonatan Parada Torres

#Customization
initial_mile = 23456
last_mile = 23678
gallon_used = 10
rounded_answer = 3

#Calculation
distance_traveled = last_mile - initial_mile
miles_per_galon = distance_traveled / gallon_used

answer = round(miles_per_galon, rounded_answer)

print(
    f"Distance Traveled: {distance_traveled} Miles\nGallon Used: {gallon_used} Gallon(s)\n"
    "How many miles per gallon did the car average between two fillings?\n"
    f"Answer: {answer} Miles/Gallon"
)
```

Output:

```
@jhonatanparada499 →/workspaces/Jhonatan_LAB1 (main) $ /home/codespace/.python/current/bin/python3
Distance Traveled: 222 Miles
Gallon Used: 10 Gallon(s)
How many miles per gallon did the car average between two fillings?
Answer: 22.2 Miles/Gallon
```

B)

I learned that by putting a list into a print function, it will display all the elements in it including the brackets, commas and marks quotation (if applicable), but if the asterisk (*) symbol is put before typing the list or the variable that defines it, it will only show the contents of the list without the brackets, commas and mark quotation.

I also learned that the print function includes built-in parameters such as ‘sep’, which in the context of lists, allows to define the characters that will be in between the contents of the list.

The most challenging question in Lab1 to me was number 2. The reason why I found it so hard was because of the use of lists and the condition of only using one print statement. This is the only question in the lab on which I saw the potential need of python lists due to the scalable and changing nature of a receipt and its items.