

Lab 6 - Report

Students:

1. Nguyễn Đình Khánh Ngân – ITCSIU22236
2. Phan Trần Thanh Huy – ITCSIU22056

1 Overview

2 Results

Main file:

```

1  def q1(x1, x2, y1, y2):
2      dst = lambda x1, y1, x2, y2: ((x1 - x2)**2 + (y1 - y2)**2)**0.5
3      return dst(x1, x2, y1, y2)
4
5  def q2():
6      A = [1, 2, 3, 4, 5, 6, 7, 8]
7      L=[]
8      L = list(map(lambda x:x ** 3,A))
9      print("Generate the power of 3 of each element in A", L)
10
11     R = list(filter(lambda x: 20 <= x <= 40, map(lambda x:x ** 2,A)))
12     print("Return the resultant square in the range of [20, 40]: ", R)
13
14     E = list(filter(lambda x: x % 2 == 0, A))
15     print("Even number in A is: ", E)
16
17 def q3():
18     inventory = [
19         {"name": "Laptop", "price": 1200, "stock": 10, "categories": ["electronics", "computers"]},
20         {"name": "Smartphone", "price": 800, "stock": 0, "categories": ["electronics", "mobiles"]},
21         {"name": "Headphone", "price": 150, "stock": 25, "categories": ["electronics", "audio"]},
22         {"name": "Desk Chair", "price": 100, "stock": 5, "categories": ["furniture", "office"]},
23         {"name": "Notebook", "price": 5, "stock": 100, "categories": ["stationery", "office"]},
24     ]
25     out_of_stock = list(filter(lambda x: x["stock"] == 0, inventory))
26     if(any(out_of_stock)):
27         print("Products that are out of stock: ", out_of_stock)
28     else:
29         print("There are not any products out of stock")
30
31     category = input("Input a category: ")
32     price = int(input("Input a price: "))
33
34     check_products = list(filter(lambda i: category in i["categories"] and i["stock"] > 0 and i["price"] > price, inventory))
35
36     if check_products:
37         print("Products in category", category, "with price >", price, "and in stock:", check_products)
38     else:
39         print("There are no products in this category that meet the conditions.")
40
41 def main():
42     num = int(input("Select the question: "))
43
44     if(num == 1):
45         x1 = int(input("Input x1: "))
46         x2 = int(input("Input x2: "))
47         y1 = int(input("Input y1: "))
48         y2 = int(input("Input y2: "))
49         print("The distance (dst) of two pixels is ", q1(x1, x2, y1, y2))
50
51     if(num == 2):
52         q2()
53
54     if(num == 3):
55         q3()
56
57 main()

```

Exercise 1

```
def q1(x1, x2, y1, y2):  
    dst = lambda x1, y1, x2, y2: ((x1 - x2)**2 + (y1 - y2)**2)**0.5  
    return dst(x1, x2, y1, y2)
```

Defines an anonymous function (dst) that

takes four numbers: the first point's x1, y1 and the second point's x2, y2 computes and then takes the square root to get the Euclidean distance

```
● PS E:\Homework\PP1\Lab\Lab6> & C:/Python312/python.exe e:/Homework/PP1/Lab/Lab6/p1.py  
Select the question: 1  
Input x1: 3  
Input x2: 4  
Input y1: 5  
Input y2: 6  
The distance (dst) of two pixels is 2.8284271247461903
```

Exercise 2

```
5 def q2():  
6     A = [1, 2, 3, 4, 5, 6, 7, 8]  
7     L=[]  
8     L = list(map(lambda x:x ** 3,A))  
9     print("Generate the power of 3 of each element in A", L)  
10  
11     R = list(filter(lambda x: 20 <= x <= 40, map(lambda x:x ** 2,A)))  
12     print("Return the resultant square in the range of [20, 40]: ", R)  
13  
14     E = list(filter(lambda x: x % 2 == 0, A))  
15     print("Even number in A is: ", E)
```

L is an anonymous function that returns the cube of its input. Map() applies that function to each element in A, produce an iterator of cube and list() collects those results into a list L. Finally, it prints out the resulting list

R applies $x**2$ to each element, then filter where resultant square in the range [20, 40] only. Then print out the list of satisfies numbers in the rage

E is an anonymous function that returns true if x is even. Filter() goes through each element of A, keeps only those which the lambda returns true, and produce an iterator of those elements. List() converts that iterator into a concrete list E. Finally, it prints out the list of even numbers

```
● PS E:\Homework\PP1\Lab\Lab6> & C:/Python312/python.exe e:/Homework/PP1/Lab/Lab6/p1.py  
Select the question: 2  
Generate the power of 3 of each element in A [1, 8, 27, 64, 125, 216, 343, 512]  
Return the resultant square in the range of [20, 40]: [25, 36]  
Even number in A is: [2, 4, 6, 8]
```

Exercise 3.1

```
17 def q3():
18     inventory = [
19         {"name": "Laptop", "price": 1200, "stock": 10, "categories": ["electronics", "computers"]},
20         {"name": "Smartphone", "price": 800, "stock": 0, "categories": ["electronics", "mobiles"]},
21         {"name": "Headphone", "price": 150, "stock": 25, "categories": ["electronics", "audio"]},
22         {"name": "Desk Chair", "price": 100, "stock": 5, "categories": ["furniture", "office"]},
23         {"name": "Notebook", "price": 5, "stock": 100, "categories": ["stationery", "office"]},
24     ]
25     out_of_stock = list(filter(lambda x: x["stock"] == 0, inventory))
26     if(any(out_of_stock)):
27         print("Products that are out of stock: ", out_of_stock)
28     else:
29         print("There are not any products out of stock")
```

This goes through each product in the inventory and take out any product that has stock equal to 0. Then put it inside a list called “out_of_stock”

After that, using function any() to test. If there is product that out-of-stock, it will returns true. Otherwise, return false. Hence, print the corresponding product.

```
PS E:\Homework\PP1\Lab\Lab6> & C:/Python312/python.exe e:/Homework/PP1/Lab/Lab6/p1.py
Select the question: 3
Products that are out of stock: [{'name': 'Smartphone', 'price': 800, 'stock': 0, 'categories': ['electronics', 'mobiles']}]
```

Exercise 3.2

Pick the function verify if all products in a specific category are available and have a price greater than a specified threshold δ , where δ is input by users

```
#If all products in a specific category are available and have a price greateer than
def check_category(inventory, category,  $\delta$ ):
    return (
        list(
            filter(lambda i: category
                    in i["categories"]
                    and i["stock"] > 0
                    and i["price"] > price, inventory))
    )
```

This goes through each product in the inventory, and only keeps those for which the lambda test returns True. Inside that test, we check three things all at once:

- category in i["categories"]: Does this product list include the category
- i["stock"] > 0: Is there at least one unit available
- i["price"] > δ : Is its price strictly greater than the user's threshold δ

Only if all three are true will the filter pass this item through. Filter by itself produces an on-the-fly sequence. Wrapping it with list(...) collects all the passing products into a real Python list. Finally, we hand back that list of category-matched, in-stock, above- δ products

```
Input a category: electronics
Input a price: 300
Products in category electronics with price > 300 and in stock: [{'name': 'Laptop', 'price': 1200, 'stock': 10, 'categories': ['electronics', 'computers']}]
```

