

IP Assignment 1

MAX MARKS: 100

SUPERMARKET

For this question, you have to implement an interactive supermarket scenario. Assume the supermarket has only the following 10 items and each item belongs to one of the three categories: Apparels, Electronics or Eatables.

Item Code	Item Name	Price	Category
0	Tshirt	500	Apparels
1	Trousers	600	Apparels
2	Scarf	250	Apparels
3	Smartphone	20,000	Electronics
4	iPad	30,000	Electronics
5	Laptop	50,000	Electronics
6	Eggs	5	Eatables
7	Chocolate	10	Eatables
8	Juice	100	Eatables
9	Milk	45	Eatables

The general flow of the program looks something like this:

1. Greet the customer and inform them about the available items
2. Decide the type of order (bulk or regular)
3. Take the order
4. Print the order summary
5. Print the category-wise cost
6. Print the discount details
7. Print the tax details
8. Ask the user for coupon (if any)
9. Print final costs
10. Say goodbye!

Don't worry, we'll discuss each step in detail :)

The first thing the program must do when a customer enters is to **greet the customer** and tell him about the list of things available at the store. This is to be done in the following manner:

```
=====
                        MY BAZAAR
=====
Hello! Welcome to my grocery store!
Following are the products available in the shop:

-----
CODE | DESCRIPTION | CATEGORY | COST (Rs)
-----
0  | Tshirt      | Apparels | 500
1  | Trousers    | Apparels | 600
2  | Scarf       | Apparels | 250
3  | Smartphone  | Electronics | 20,000
4  | iPad        | Electronics | 30,000
5  | Laptop      | Electronics | 50,000
6  | Eggs        | Eatables  | 5
7  | Chocolate   | Eatables  | 10
8  | Juice       | Eatables  | 100
9  | Milk        | Eatables  | 45
-----
```

The next step is to **ask the customer if he wants to buy in bulk or not**. This has to be done in the following manner:

```
Would you like to buy in bulk? (y or Y / n or N):
```

After asking this, you have to **take input from the customer**. The valid inputs are 'y' or 'Y' for Yes and 'n' or 'N' for No. If the customer enters any invalid input, keep asking the same question until a valid input is entered by the customer.

Depending on whether the customer is a bulk buyer or a regular buyer, you have to take the order from the customer.

Case 1 - Regular Customer

The first thing you have to do is take the order from the customer. This is to be done in the following manner:

```
-----  
ENTER ITEMS YOU WISH TO BUY  
-----  
Enter the item codes (space-separated): 0 1 1 3 4 8 1 8
```



Next, you have to **print an order summary** for the customer. This is to be done in the following manner:

```
-----  
ORDER DETAILS  
-----  
[1] Tshirt x 1 = Rs 500 * 1 = Rs 500  
[2] Trouser x 3 = Rs 600 * 3 = Rs 1800  
[3] Smartphone x 1 = Rs 20000 * 1 = Rs 20000  
[4] iPad x 1 = Rs 30000 * 1 = Rs 30000  
[5] Juice x 2 = Rs 100 * 2 = Rs 200
```

Note that the items are to be **printed in ascending order of their code**. If the customer enters an invalid code, you have to print an error message (decide yourself what the message should be) and skip this item. For e.g. if the user enters the item codes as **1 -2 3 5**, then just consider the items with code 1, 3 and 5.

The next thing you should do is **tell the customer about the category-wise cost**. This means that you should print the amount the customer is spending on each of the three categories, namely Apparels, Electronics or Eatables. This is to be done in the following manner:

CATEGORY-WISE COST

Apparels = Rs 2300

Electronics = Rs 50000

Eatables = Rs 200

The supermarket, having a lot of competition in the market, also offers **some discounts to its customers**. These discounts are as follows:

- Discount on Apparels: 10% off if the base amount of Apparels before taxes ≥ 2000
- Discount on Electronics: 10% off if the base amount of Electronics before taxes ≥ 25000
- Discount on Eatables: 10% off if the base amount of Eatables before taxes ≥ 500

You need to apply these discounts automatically after finding the category-wise cost and inform the customer about the applied discounts. You should do this in the following manner:

DISCOUNTS

[APPAREL] Rs 2300 – Rs 230 = Rs 2070

[ELECTRONICS] Rs 50000 – Rs 5000 = Rs 45000

TOTAL DISCOUNT = Rs 5230

TOTAL COST = Rs 47270

Note that since the cost of the Eatables category was < 500 , there was no discount applied on that category.

For every category, there will be a **tax that the customer will have to pay**. These taxes are as follows:

- Apparels: 10% on the cost of apparels after discount
- Electronics: 15% on the cost of electronics after discount
- Eatables: 5% on the cost of eatables after discount

Again, you will have to tell the customer about these taxes. Do it in the following manner: 4

TAX

[APPAREL] Rs 2070 * 0.10 = Rs 207

[ELECTRONICS] Rs 45000 * 0.15 = Rs 6750

[EATABLES] Rs 200 * 0.05 = Rs 10

TOTAL TAX = Rs 6967

TOTAL COST = Rs 54237

The next thing to do is to **ask the customer for any coupon codes**. There are 2 valid coupon codes that the supermarket accepts. These are:

- **HELLE25:** The terms and conditions of HELLE25 are as follows:
 - It is valid only if the cost after discounts and taxes is ≥ 25000
 - It offers a discount of 25%, upto Rs. 5000
- **CHILL50:** The terms and conditions of CHILL50 are as follows:
 - It is valid only if the cost after discounts and taxes is ≥ 50000
 - It offers a discount of 50%, upto Rs. 10000

NOTE: Coupon codes are **case-sensitive**.

If the customer enters an invalid coupon code, keep asking for the coupon code until either a valid coupon code is entered or the customer leaves it blank, in which case assume that the customer does not have any coupon code and proceed further.

You have to take inputs from the customer in the following manner:

- Example for when the customer enters an invalid coupon code:

COUPON CODE

Enter coupon code (else leave blank): ABCD99
Invalid coupon code. Try again.

Enter coupon code (else leave blank):
No coupon code applied.

TOTAL COUPON DISCOUNT = Rs 0
TOTAL COST = Rs 54237

- Example for when the customer has a valid coupon code:

Input from user

COUPON CODE

Enter coupon code (else leave blank): CHILL50
[CHILL50] $\min(10000, \text{Rs } 54237 * 0.50) = \text{Rs } 10000$

TOTAL COUPON DISCOUNT = Rs 10000
TOTAL COST = Rs 44237

Note that in this case, the user enters the coupon code CHILL50. Since this is a valid code, you have to communicate to the customer that his code has been accepted and what is the logic behind the discounted amount. This is done in the above screenshot through the line:

[CHILL50] $\min(10000, \text{Rs } 0.5 * 27158) = \text{Rs } 10000$

Now that you have communicated the final amount the customer has to pay, all that is left to do is to say goodbye to the customer. Do this in the following manner:

Thank you for visiting!

Case 2 - Bulk Customer

In case the customer is a bulk buyer, the only change is in the way you have to take the order from the customer. This means that in the general flow of the assignment explained in the beginning, only step 3 behaves differently if the customer is a bulk buyer.

If the customer is a bulk buyer, it would be very tedious to enter space separated integers for ordering. For instance, suppose the customer wants to buy 50 trousers; it makes no sense to make him type the integer 1 fifty times. So in this case, we will have to take the order in a slightly different manner.

In this case, you have to keep on asking the customer which item he wishes to buy, and its quantity. For every question you ask, the customer has to enter 2 space separated integers denoting item code and item quantity respectively. You have to keep on taking input from the customer until the customer leaves the input blank, in which case you assume that the customer has ordered everything he wishes to and his order is complete.

ENTER ITEM AND QUANTITIES

Enter code and quantity (leave blank to stop): 1 50
You added 50 Trousers

Enter code and quantity (leave blank to stop): 3 10
You added 10 Smartphone

Enter code and quantity (leave blank to stop): 1 20
You added 20 Trousers

Enter code and quantity (leave blank to stop): 4 -5
Invalid quantity. Try again.

Enter code and quantity (leave blank to stop): 25 2
Invalid code. Try again.

Enter code and quantity (leave blank to stop): 25 -5
Invalid code and quantity. Try again.

Enter code and quantity (leave blank to stop):
Your order has been finalized.

Inputs
from
user

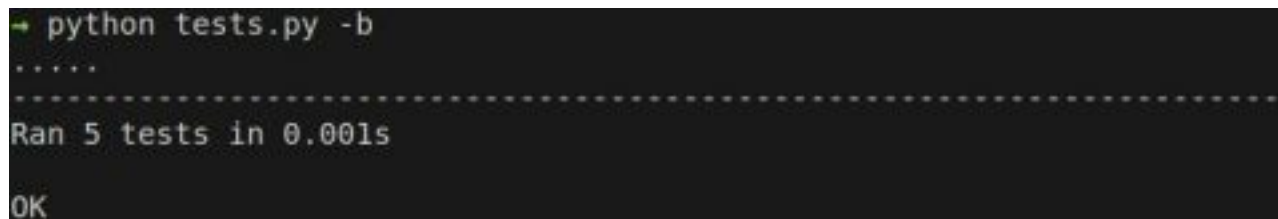
Please note the following points:

- In the above example, the customer entered 1 50 in one query and 1 20 in another query. 1 is the item code for a trouser. In this case, you have to make sure that the final order contains 70 trousers.
- The customer may enter some invalid values, in which case you have to print the corresponding error message.
- For every valid input from the customer, you have to print a small summary for every item, as can be seen in the above screenshot.
- In the second last line, since the user has input an empty string, we finalize the user's order. 8

Procedure to complete the assignment:

- Make sure you are using Python 3 for completing this assignment.
- Do not rename any files that are provided.
- Fully understand the PDF first. Everything is explained in a detailed manner. Check how the output is being shown on the console in a user-friendly way.
- After this, check out the a1.py file, where you need to write the code. There are functions defined in the file along with the following things:
 - Description: explaining what has to be done in the function.
 - Parameters: explaining the arguments and their data types. You need to adhere to these data types properly since the grading depends on the data types you consider.
 - Returns: explaining what a function will return, if it is returning something.
- In each function, there is only one statement in their body, which is "pass". Please remove the statement before writing any code in the function.
- We have provided a sample run of both the cases, bulk and regular, in the files bulk.txt and regular.txt respectively. You can refer to these files to check how the program should ask for input and display output.
- If your input prompts (e.g. "Would you like to buy in bulk? (y or Y / n or N):") or output messages (e.g. "TOTAL COST = Rs 12345") do not exactly match with the ones that have been provided to you, marks will not be deducted. But we strongly recommend that you keep the output as close to that provided in bulk.txt and regular.txt as possible. However, if we find that you are not at all, in any format, printing a particular piece of information (e.g. the total tax), then marks will be deducted. So make sure not to miss printing any information.
- You will be graded on the correctness of your implementation of the functions, the proper return types and values etc. Note that input prompts and print statements do not affect the testing of the functions. You just need to make sure that you are taking input in the correct format (e.g. space-separated) and returning the correct values in each of the functions. What input message you display, how you print etc. does not matter for the testing part. But try to keep it neat.
- For the grading part, the functions that you implement will be tested independently. That is, we can test a specific function without calling any other function (not even the main function). So, for example if we are testing the calculate_tax function, it will be the very first thing that is executed in our test, even though, logically speaking, many functions should run before this function is called. This is possible because the functions are self-sufficient and need only the parameters provided to achieve the mentioned functionality. Make sure that your code is not written such that for the correct functioning of a particular given function, it assumes that it was called only after some other functions were run. For example, the calculate_tax function should not assume that any other function (let's say calculate_discounted_prices) must have executed before it. So you can call other functions inside a given function, but you cannot assume that some function must have already been executed before the particular function is executed. However, you can assume that any global variables and import statements will definitely be executed before the testing script tests a particular function.

- For some parts, you will need to calculate discounts, coupon discounts and tax. For these purposes, we have provided two pre-defined functions in a1.py, namely get_discount and get_tax. You must use these instead of defining your own. Read their description to understand how to use them.
- After completing the a1.py file, you can check out the file “tests.py”. You don’t need to understand this file as it just contains some example test cases which will test the logic of your function implementations in the “a1.py” file. You can run the tests file as shown in the below screenshot. As you can see, there are 5 tests in the file. If everything works fine, you will get the same result. There will be more test cases, covering different corner cases, which will be tested after you submit the assignment.



```
→ python tests.py -b
.....
-----
Ran 5 tests in 0.001s
OK
```

What you need to submit:

A zip file containing only the completed a1.py file. Name the zip file as **a1_<Roll Number>.zip**. So for example if your roll number is 2020001, name the zip file as **a1_2020001.zip**.