LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Choose a Module

60% Individual Coursework

2023 Spring

Student Name: Arbin Rajbanshi

London Met ID: 22067054

College ID: np01nt4a220016

Assignment Due Date: Friday, May 12, 2023

Assignment Submission Date: Friday, May 12, 2023

Word Count:

Project File Links:

Google Drive Link:	Keep Google Drive URL of your Project Here with Anyone in Organization can View Option Enabled

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction

A laptop shop buys laptops/computers from manufacturers and sells it to various customers which may be individuals or companies is a project which help us to understand about Ordering from manufacturers, Selling to customers, Updating stock information and so on .

Orders for computers can be placed for the manufacturers. Likewise, customers can place orders for computers to our laptop shop (distributor). The laptop rental shop manages information about the available computers in a text file. A program that can read the text file to display all the laptops available and make changes to the text file according to the nature of the transaction (ordering from manufacturer/selling to customer) needs to be developed. With each order or sale made, a note/receipt should be generated with the details of the transaction. The stock of a particular laptop should also be updated in the main text file itself. For example, if the shop currently has 19 pieces of Razer Blade, and a customer buys 2 of it, the stock should be updated to 17. In case the rental shop purchases 5 Razer Blade laptops, the stock should then be updated from 17 to 22 i.e. the stock should be increased by 5.

The development of a program to manage laptop shop operations is essential for streamlining business processes, maintaining accurate inventory records, and facilitating smooth transactions with both manufacturers and customers. By leveraging a text file as a central database, the program can read, update, and append relevant information related to available laptops, orders, and stock quantities. Additionally, the program can generate comprehensive notes or receipts for each transaction, providing a clear overview of the details involved. With efficient management and automation of these processes, the laptop shop can enhance its operational efficiency, customer satisfaction, and overall profitability.

2. Discussion and Analysis

2.1. Algorithm

An Algorithm is step wise procedure to be followed to solve a problem. Developing algorithm is first step towards programming. It defines the steps to be taken which later leads the way towards creating a software. An algorithm is not limited to one platform. In fact, same algorithm can be used to develop programs in almost all programming language available till date. Here's the algorithm for this project:

Step 1: start

Step 2: set main loop true, with in main loop do the following

Step 3: Read the text file containing the information about available laptops.

Step 4: Display the list of laptops to the user, showing details such as brand, model, and current stock quantity.

Step 5: Ask the user whether they want to place an order with a manufacturer or sell a laptop to a customer.

Step 6: If the user chooses to place an order with a manufacturer:

a. Prompt the user to enter the details of the order, including the brand, model, and quantity of laptops.

b. Append the order details to the text file, reflecting the incoming stock.

c. Generate a note or receipt for the order, including the transaction details.

d. Save the changes to the text file.

Step 7: If the user chooses to sell a laptop to a customer:

a. Prompt the user to enter the details of the sale, including the brand, model, and quantity of laptops sold.

b. Check if the requested quantity is available in stock by comparing it to the current stock quantity in the text file.

c. If the requested quantity is available:

i. Deduct the sold quantity from the stock in the text file.

ii. Generate a note or receipt for the sale, including the transaction details.

iii. Save the changes to the text file.

d. If the requested quantity is not available, display an out of stock message indicating insufficient stock.

Step 8: Ask the user if they want to perform another transaction or exit the program.

Step 9: If the user wants to perform another transaction, go back to step 3.

Step 10: If the user wants to exit the program, end the program execution pressing 4.

Step 11: STOP

2.2 FLOWCHART

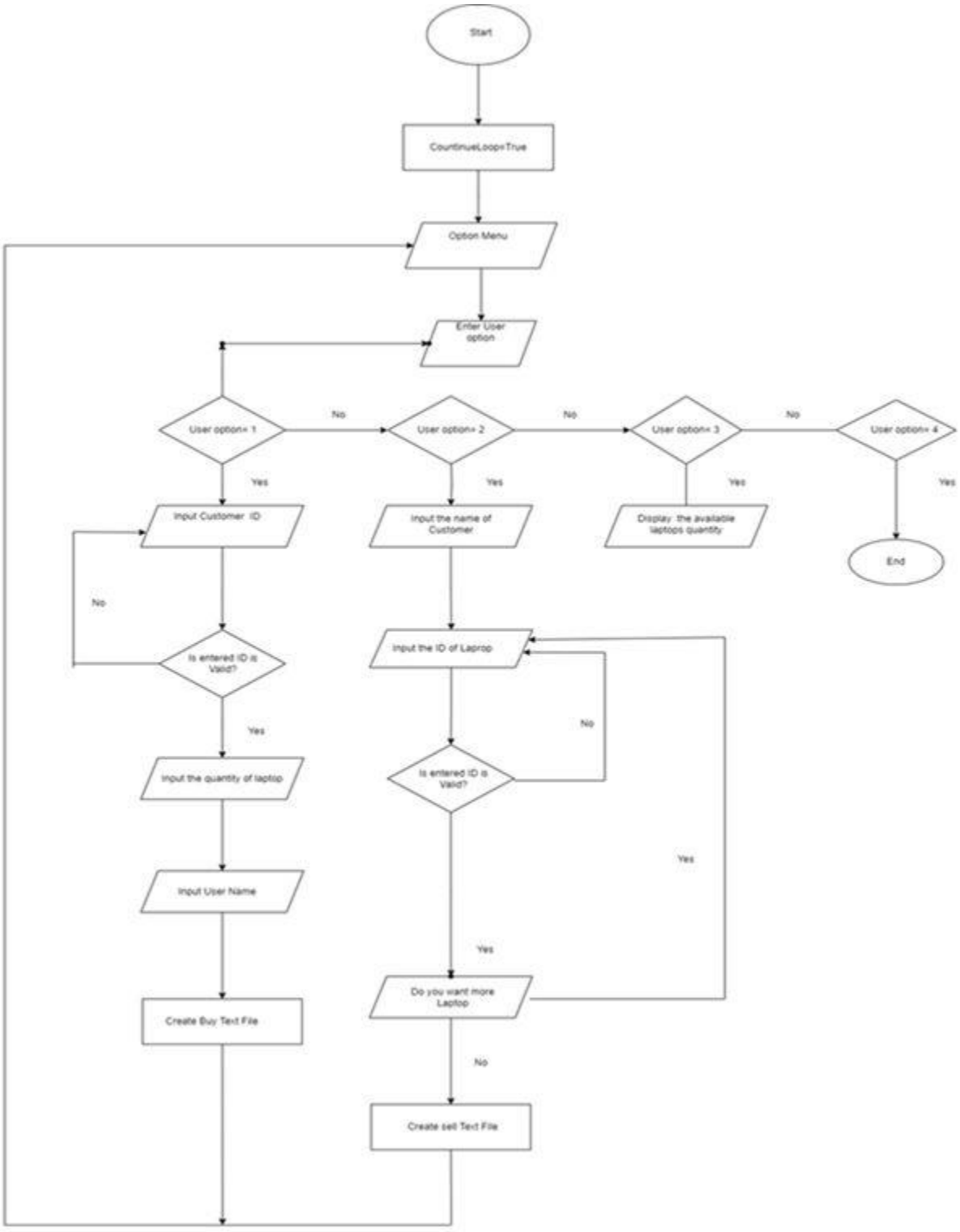


Figure 1: Flowchart

2.3 PSEUDOCODE

```
main.py
    Function: displayLaptops
        Print the table header
        Initialize a counter variable a to 1
        Open the "solitude.txt" file in read mode and assign it to the variable file
        For each line in the file
            Print the line number (a) and the line with commas replaced by tabs
            Increment the counter variable a by 1
        Close the file
        Print a separator line

    Print a welcome message

    Set continueLoop to True

    While continueLoop is True
        Print the menu options
        Try
            Read an integer input from the user and assign it to the variable userinput
            If userinput is 1
                Call the buyLaptop function and assign the returned values to variables
                Call the W_laptoppurchase function with the provided variables as
arguments
            Else if userinput is 2
                Call the sellLaptop function and assign the returned values to variables
                Call the W_laptopsell function with the provided variables as arguments
            Else if userinput is 3
```

Call the displayLaptops function

Else if userinput is 4

Set continueLoop to False

Print a farewell message

Else

Print an error message

Catch ValueError

Print an error message for invalid input

operation.py

Function: buyLaptop

Print a thank you message and prompt for name and phone number

Loop until a valid name is entered

Try to read a name from the user

If the name is not alphabetic, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Loop until a valid phone number is entered

Try to read a phone number from the user

If the phone number is not numeric or has a length other than 10, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Print the available laptops

Initialize an empty list for purchased laptops

Loop until the user decides not to purchase more laptops

Read a valid product ID from the user

Loop until a valid quantity is entered

Try to read a quantity from the user

If the quantity is not numeric, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Update the quantity in the text file

Calculate the details of the purchased laptop and add it to the purchased_laptops list

Ask the user if they want to purchase more laptops

If the user does not want to purchase more laptops, break the loop

Calculate the total and VAT

Get the current date and time

Print the purchase details and the final total

Return the name, phone number, date and time, purchased_laptops, VAT, and final_total

Function: sellLaptop

Print a thank you message and prompt for name and phone number

Loop until a valid name is entered

Try to read a name from the user

If the name is not alphabetic, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Loop until a valid phone number is entered

Try to read a phone number from the user

If the phone number is not numeric or has a length other than 10, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Print the available laptops

Initialize an empty list for sold laptops

Loop until the user decides not to sell more laptops

Read a valid product ID from the user

If the user enters 9, break the loop

Loop until a valid quantity is entered

Try to read a quantity from the user

If the quantity is not numeric or not within the available quantity, raise a ValueError

Break the loop

Catch the ValueError and print the error message

Update the quantity in the text file

Calculate the details of the sold laptop and add it to the sold_laptops list

Ask the user if they want to sell more laptops

If the user does not want to sell more laptops, break the loop

Get the current date and time

Print the selling details and the final total

Return the name, phone number, date and time, sold_laptops, shipping_cost, and final_total

read.py

Function: read_file

Open the "solitude.txt" file in read mode

Initialize laptop_sn as 1

Initialize an empty dictionary MyLap_dict

Loop through each line in the file

Remove the newline character from the line

Split the line by comma and assign the resulting list to MyLap_dict[laptop_sn]

Increment laptop_sn by 1

Close the file

Return MyLap_dict

Write.py

Function: W_laptopsell

Open a new file using the provided name and phone number

Write the shop's welcome message

Write the shop's address and contact information

Write section separators and headers for selling details

Write customer information: name, phone number, date and time

Write section separators and headers for buying details

Write table headers for laptop details

Loop through each laptop sold

Write the laptop name, quantity, unit price, and total in the table format

Write section separators

Write shipping cost and total amount information

Close the file

Function: W_laptoppurchase

Open a new file using the provided name and phone number

Write the shop's welcome message

Write the shop's address and contact information

Write section separators and headers for laptop details

Write customer information: name, phone number, date and time

Write section separators and headers for buying details

Write table headers for laptop details

Loop through each laptop bought

Write the laptop name, quantity, unit price, and total in the table format

Write section separators

Write VAT amount and total amount information

Close the file

3. DATA STRUCTURES

The data structure is an idea to organize various types of data in memory. In other words, data structures are several ways to efficiently organize data in memory to perform several operations. We use it to manage, process, and efficiently get relevant information. There will be two primary components in every data structure: data and various operations working on the data. Data is information, and operations are algorithms working on that data to get valuable insights.

Strings:

A string is a data type used in programming, such as an integer and floating point unit, but is used to represent text rather than numbers. It is comprised of a set of characters that can also contain spaces and numbers. For example, the word "hamburger" and the phrase "I ate 3 hamburgers" are both strings. Even "12345" could be considered a string, if specified correctly. Typically, programmers must enclose strings in quotation marks for the data to recognized as a string and not a number or variable name.

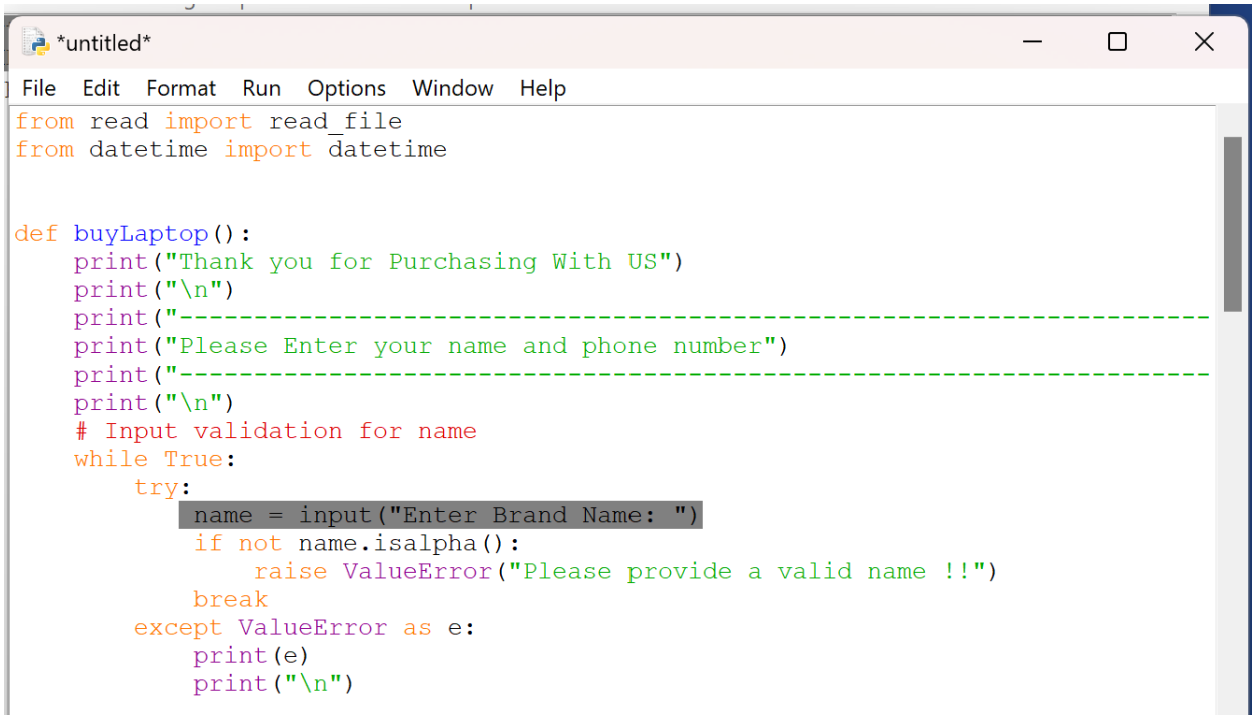


Figure 2: String Data in python

List:

A list in Python is used to store the sequence of various types of data. A list can be defined as a collection of values or items of different types. Python lists are mutable type which implies that we may modify its element after it has been formed. The items in the list are separated with the comma (,) and enclosed with the square brackets [].Although Python has six data types that may hold sequences, the list is the most popular and dependable form. The collection of data is stored in a list, a sequence data type. Similar sequence data formats are Tuples and String. Python lists are identical to dynamically scaled arrays that are specified in other languages, such as Java's ArrayList and C++'s vector. A list is a group of items that are denoted by the symbol [] and subdivided by commas.

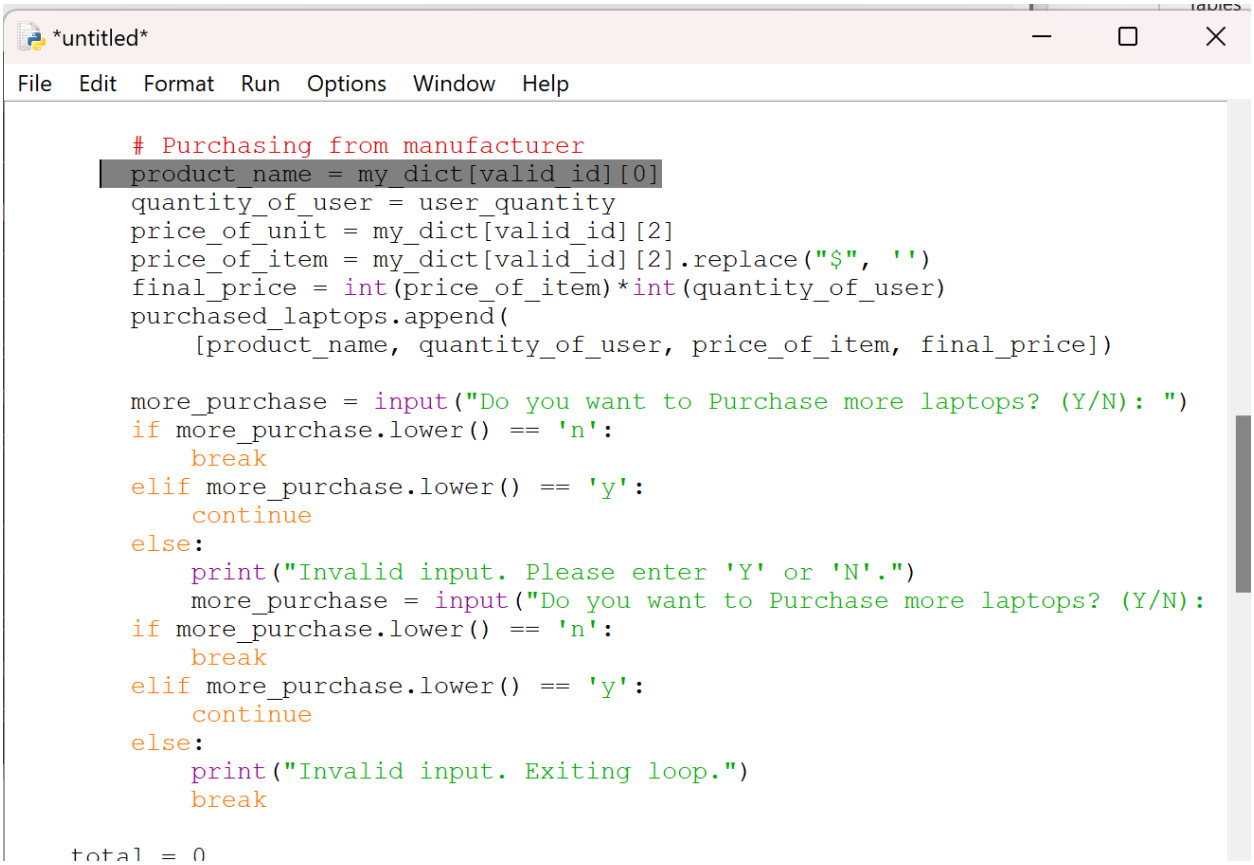


Figure 3: Creating List

Dictionary:

Dictionary is an unsorted collection of items. Dictionaries are created by specifying key-value pairs separated by colons inside curly braces. The values of dictionaries can be accessed using keys. The key must be of an immutable type like string, number or tuple. The value can be anything. Dictionary is a mutable data type, thus we can add, remove and modify key-value pairs. Dictionaries can be used efficiently to add values like details of a person/item in keys like name of the person/item.

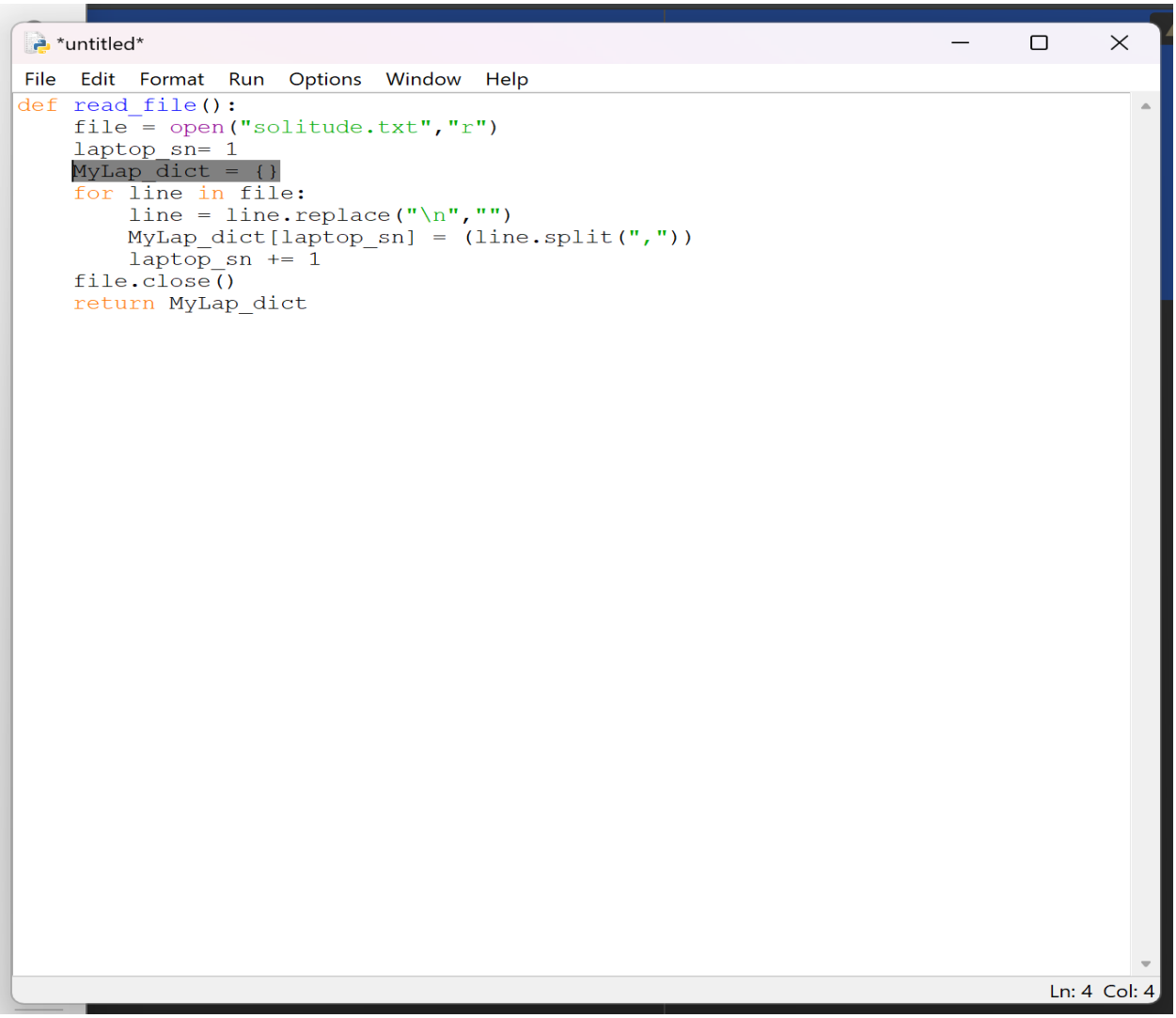


Figure 4: Dictionary creation

Integer:

Integers data type are whole numbers containing number 0 and other numbers in both positive and negative. Such as …. -1, 0,1,2,3 and so on.

Boolean:

Boolean data types are those types of data which hold only either True or False value. Example: access = true It is used in different parts program to check user input password, ask whether the user wants to continue or not.

Float:

Float data type is that type of type which can hold numbers with decimal point value in them. Example: 12.5, 12.32, 0.2. In the program float is used to hold the data of price of the books.

4.Program

The definition of a program at its most basic is a sequence of Python statements that have been crafted to do something. Even our simple hello.py script is a program. It is a one-line program and is not particularly useful, but in the strictest definition, it is a Python program. It might be easiest to understand what a program is by thinking about a problem that a program might be built to solve, and then looking at a program that would solve that problem.

Lets say you are doing Social Computing research on Facebook posts and you are interested in the most frequently used word in a series of posts. You could print out the stream of Facebook posts and pore over the text looking for the most common word, but that would take a long time and be very mistake prone. You would be smart to write a Python program to handle the task quickly and accurately so you can spend the weekend doing something fun.

main.py

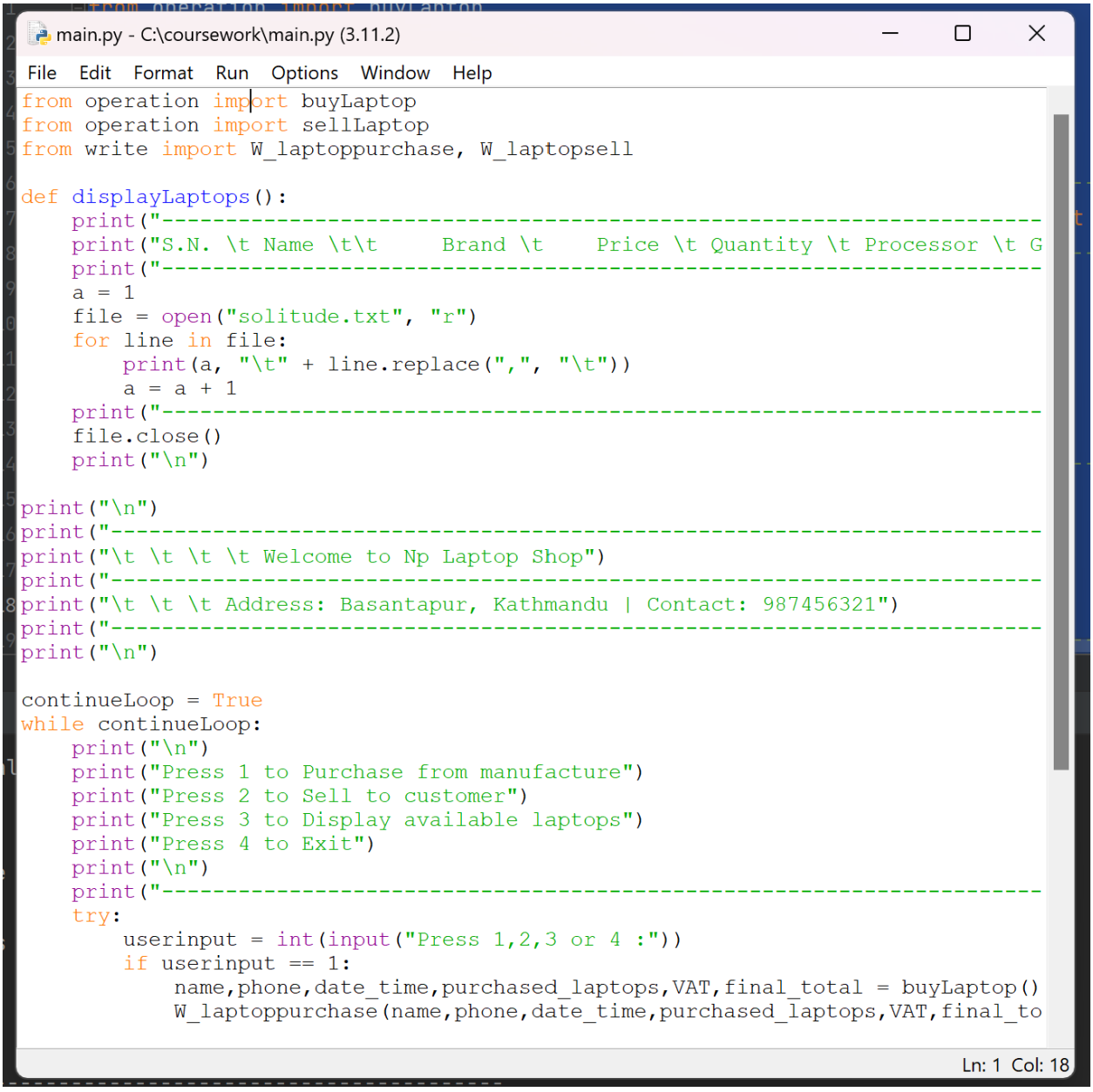


Figure 5:main.py code

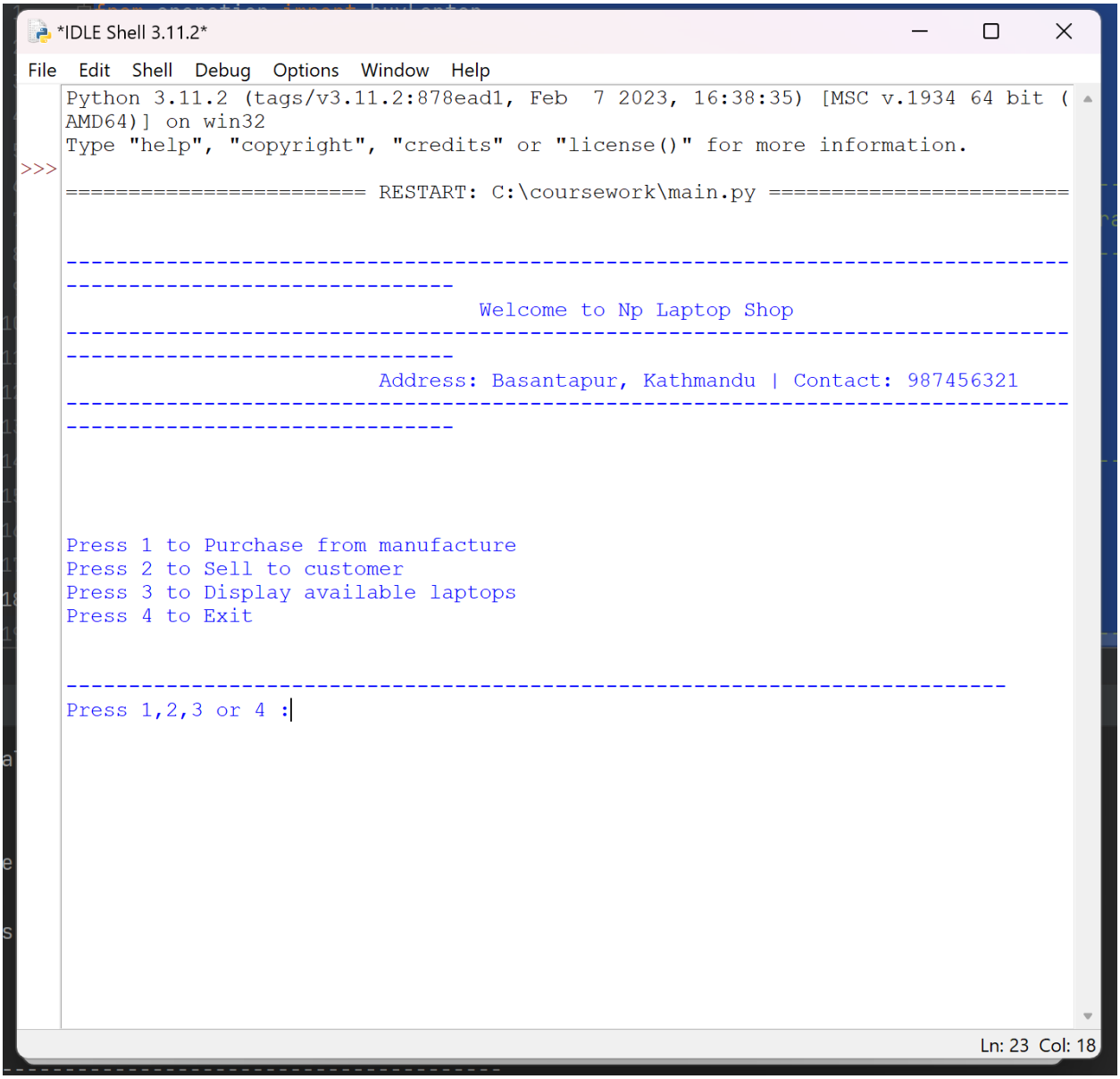


Figure 6:Output of main.py

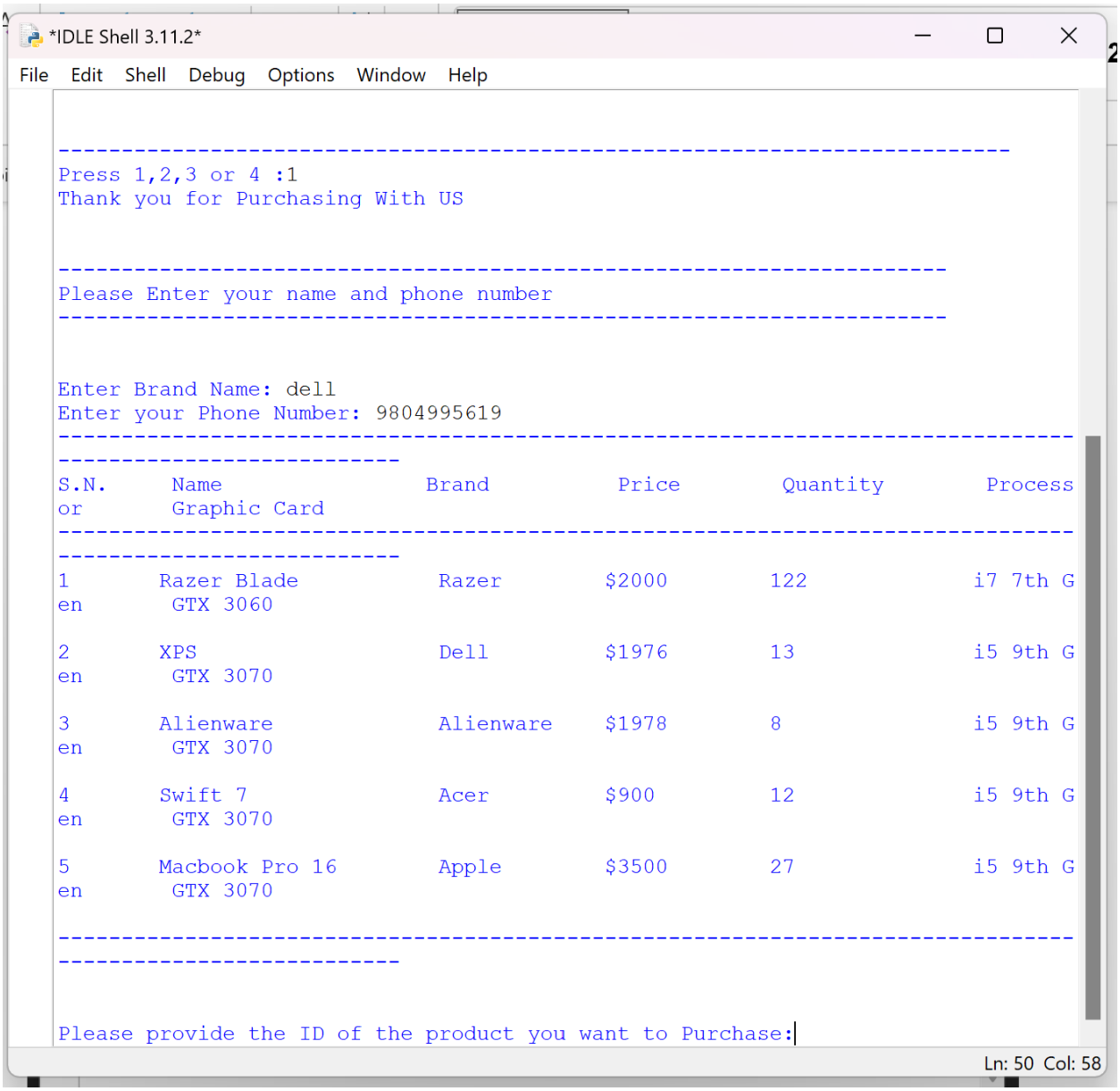


Figure 7:Displaying all items

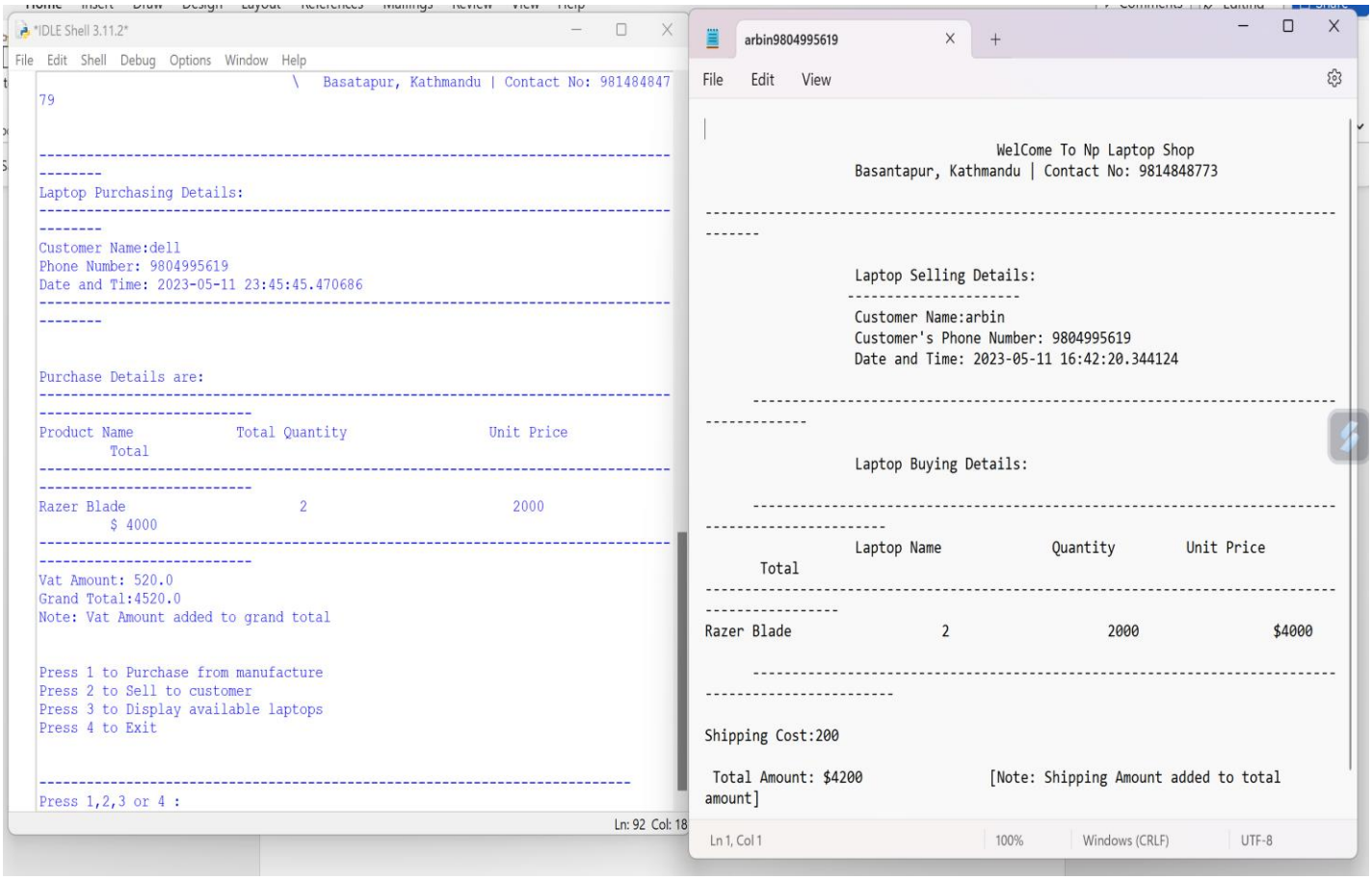


Figure 8:Screen dumps main.py

5. Description

5.1 Testing

5.1.1 Test1

Test No.	1
Objective	To show the implementation of try, except
Action	• Request the user to input 1,2,3 or 4 • If the user input (quantity) doesn't match show appropriate message
Expected Result	Appropriate message is display
Actual Result	Appropriate message is displayed
Conclusion	The test was successful

Table 1: To show the implementaion of try,except

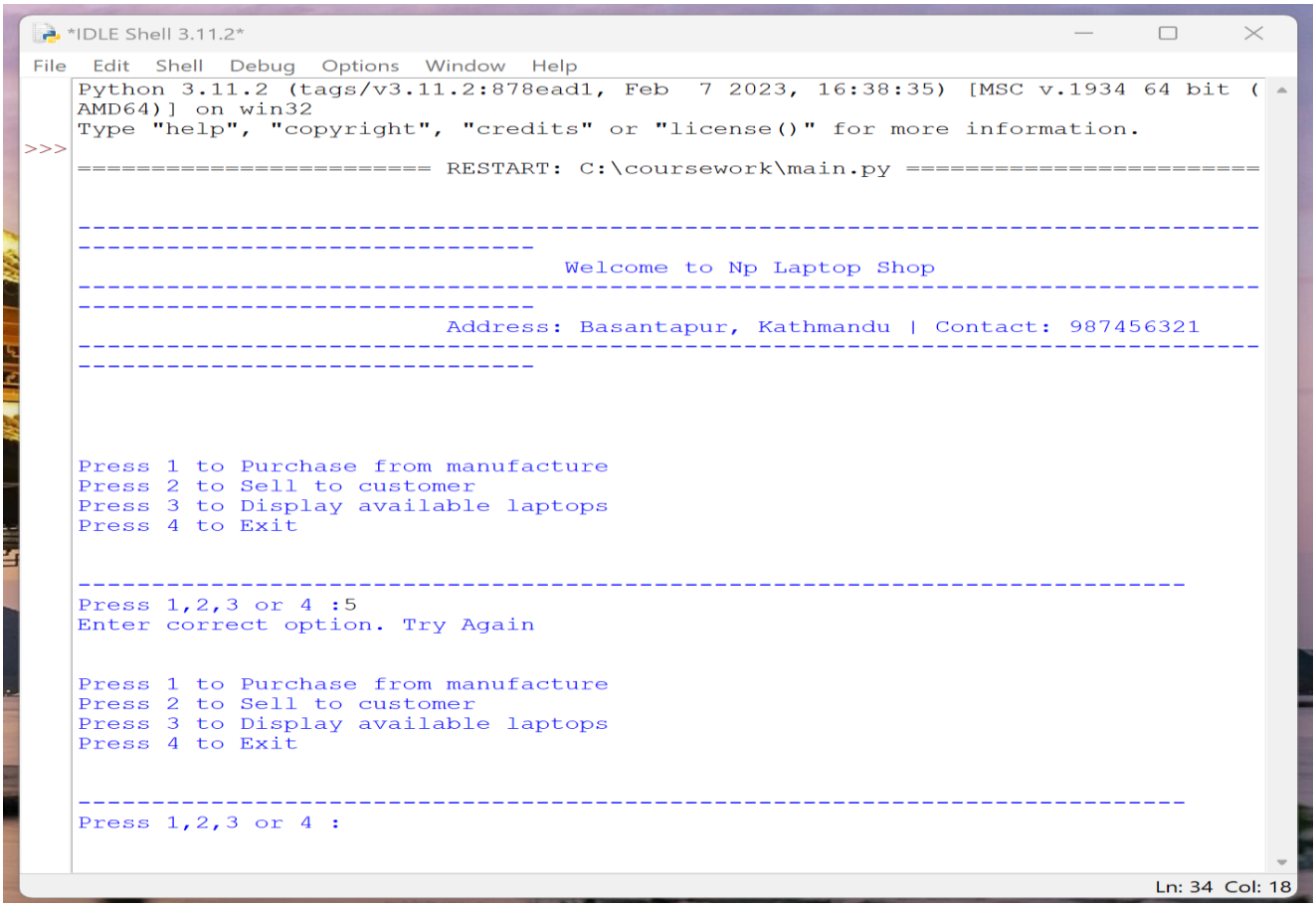


Figure 9:Screenshot of invalid message

5.1.2 Test2

Test 2.	2
Objective	TO test the selection purchase and sell of laptops
Action	• Request user to input laptop id, laptop quantity • If the user inputs negative value while choosing the laptop id, laptop quantity, appropriate message is displayed. • If the user inputs non-existed value While choosing the laptop id, laptop quantity , program ends.
Expected Result	When negative input or non-existed input, the program should end.
Actual Result	When negative input or non-existed input, the program should end.
Conclusion	The test was successful

Table 2: To test the selection purchase and sale of laptops

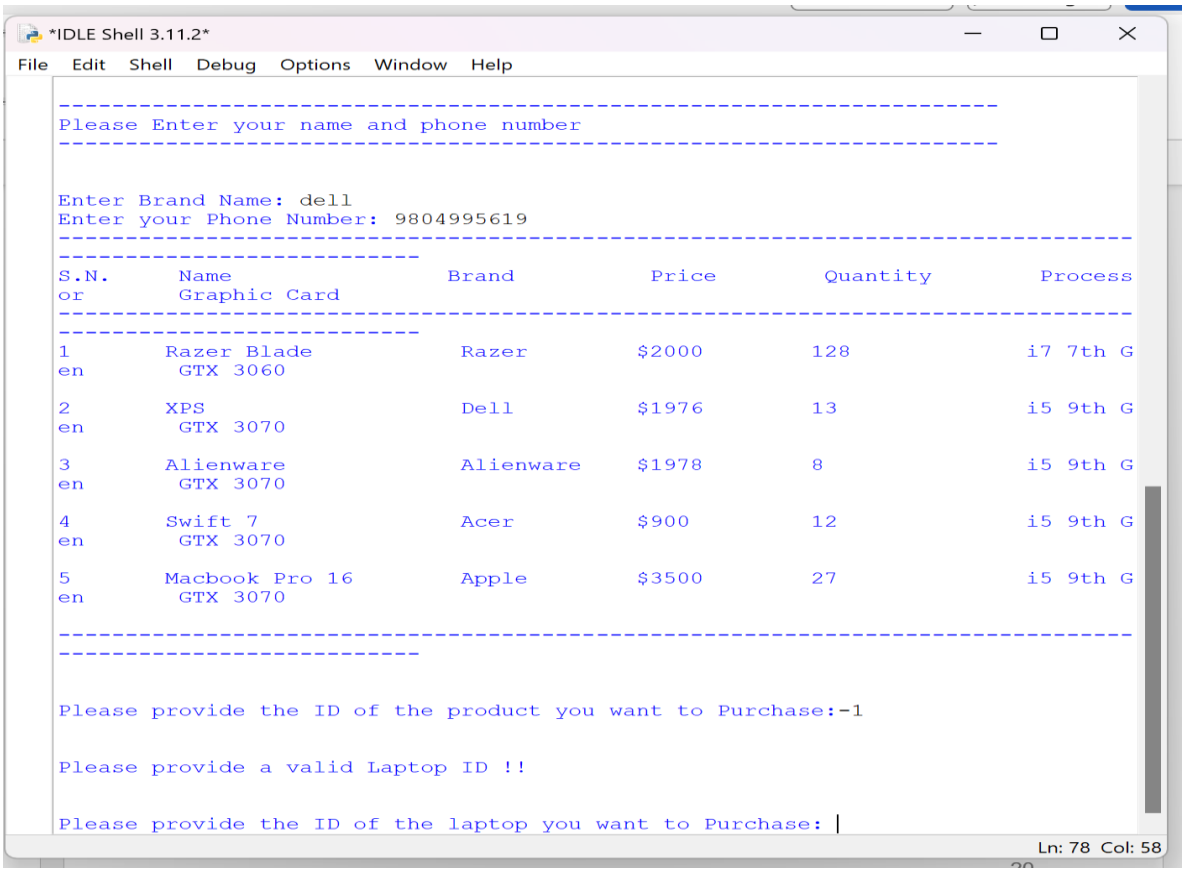


Figure 10:Screenshot of negative integer

5.1.3 Test3

Test	3
Objective	To test the generation of purchase of laptop
Action	• After the user input 1 • Request user to input name, phone number, laptop id and laptop quantity • Ask user if they want to buy more laptops • If the user inputs “Y” or “y” multiple laptops can be bought • If the users inputs “N” or “n”, a bill is printed with invoice
Expected Result	A bill is printed and a invoice is generated in txt file.
Actual Result	A bill is printed and a invoice is generated in txt file
Conclusion	The was successful

Table 3:To test the generation of purchase laptop

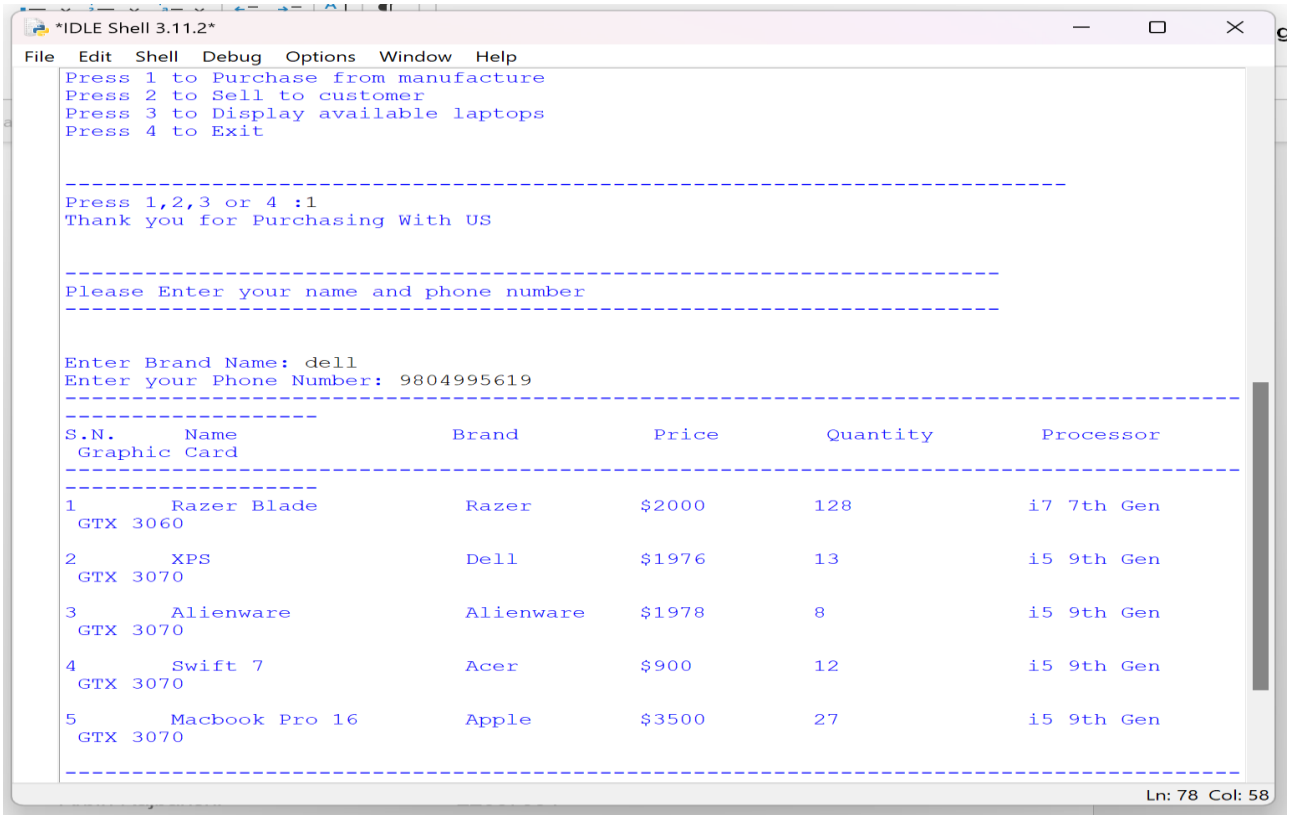
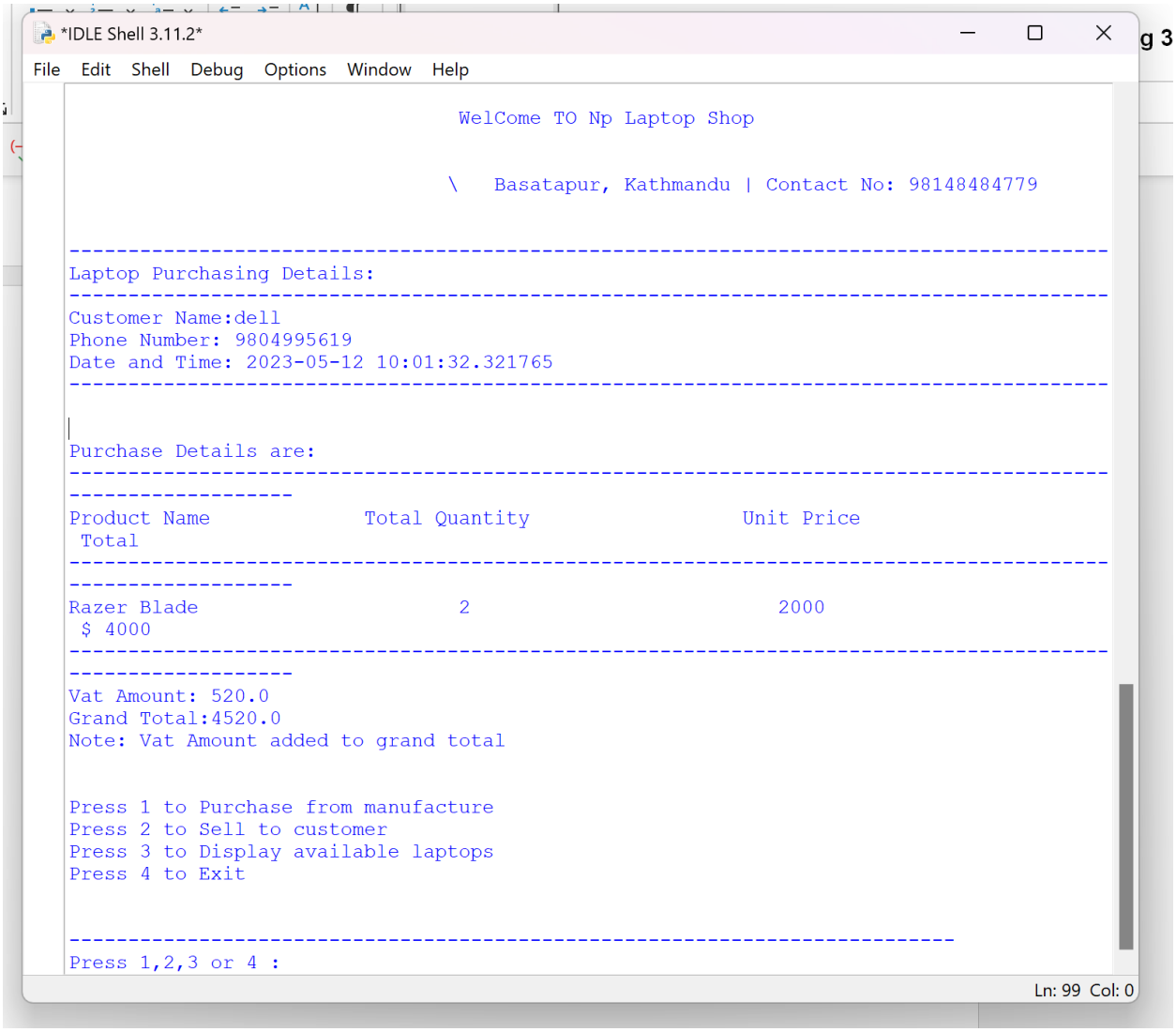


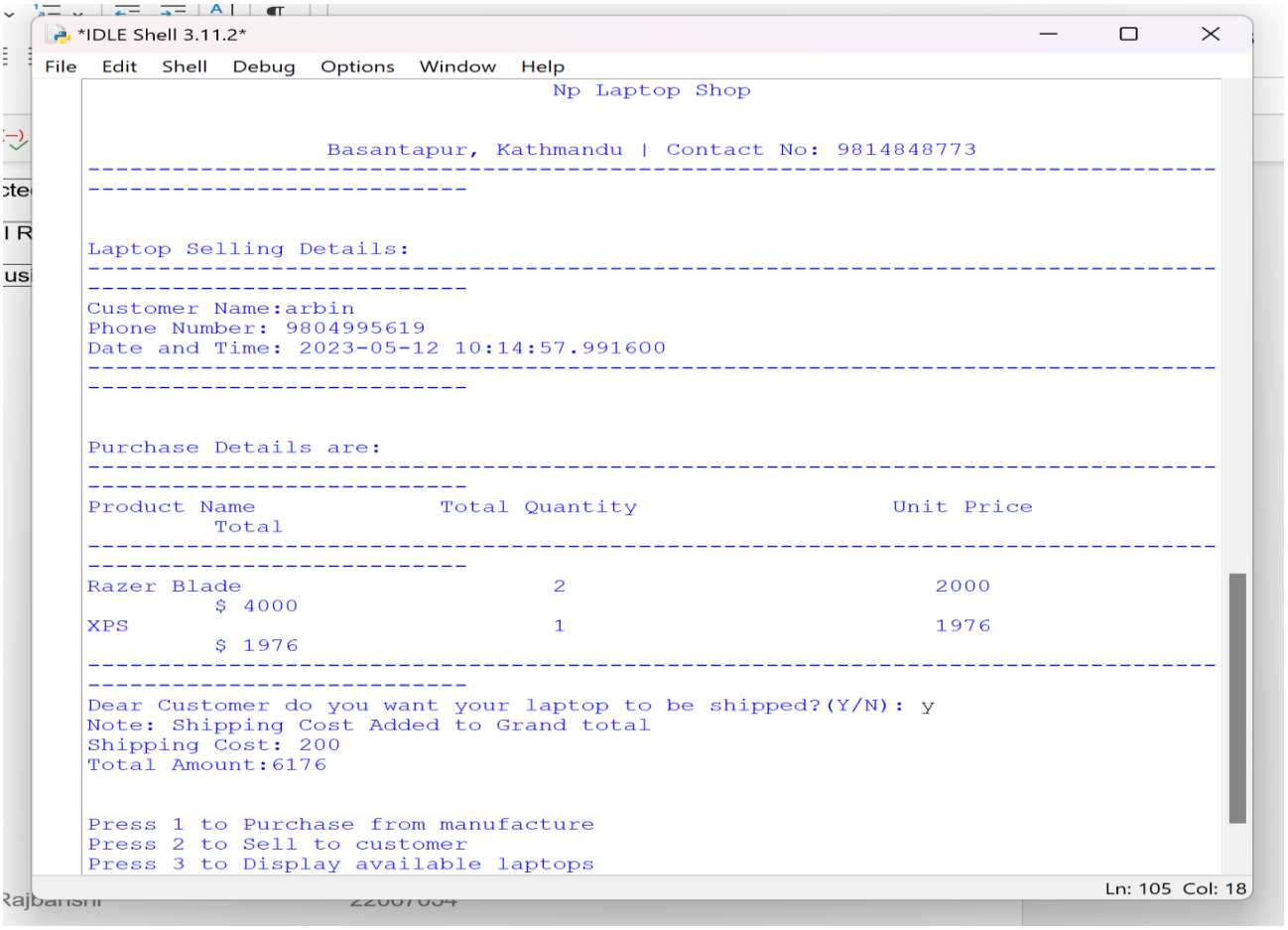
Figure 11:Screenshot of purchase generated


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
WelCome TO Np Laptop Shop
\ Basatapur, Kathmandu | Contact No: 98148484779
Laptop Purchasing Details:
Customer Name:dell
Phone Number: 9804995619
Date and Time: 2023-05-12 10:01:32.321765
Purchase Details are:
Product Name Total Quantity Unit Price
Total
Razer Blade 2 2000
$ 4000
Vat Amount: 520.0
Grand Total:4520.0
Note: Vat Amount added to grand total
Press 1 to Purchase from manufacture
Press 2 to Sell to customer
Press 3 to Display available laptops
Press 4 to Exit
Press 1,2,3 or 4 :
Ln: 99 Col: 0

5.1.4 Test4

Test	4
Objective	To test the generation of sales process of laptop
Action	• After the user input 1 • Request user to input name, phone number, laptop id and laptop quantity • Ask user if they want to sell more laptops • If the user inputs “Y” or “y” multiple laptops can be sold • If the users inputs “N” or “n”, bill is printed with invoice
Expected Result	A bill is printed and a invoice is generated in txt.file
Actual Result	A bill is printed and a invoice is generated in txt.file
Conclusion	The test was successful

Table 4:To test file generated of sales process of laptop



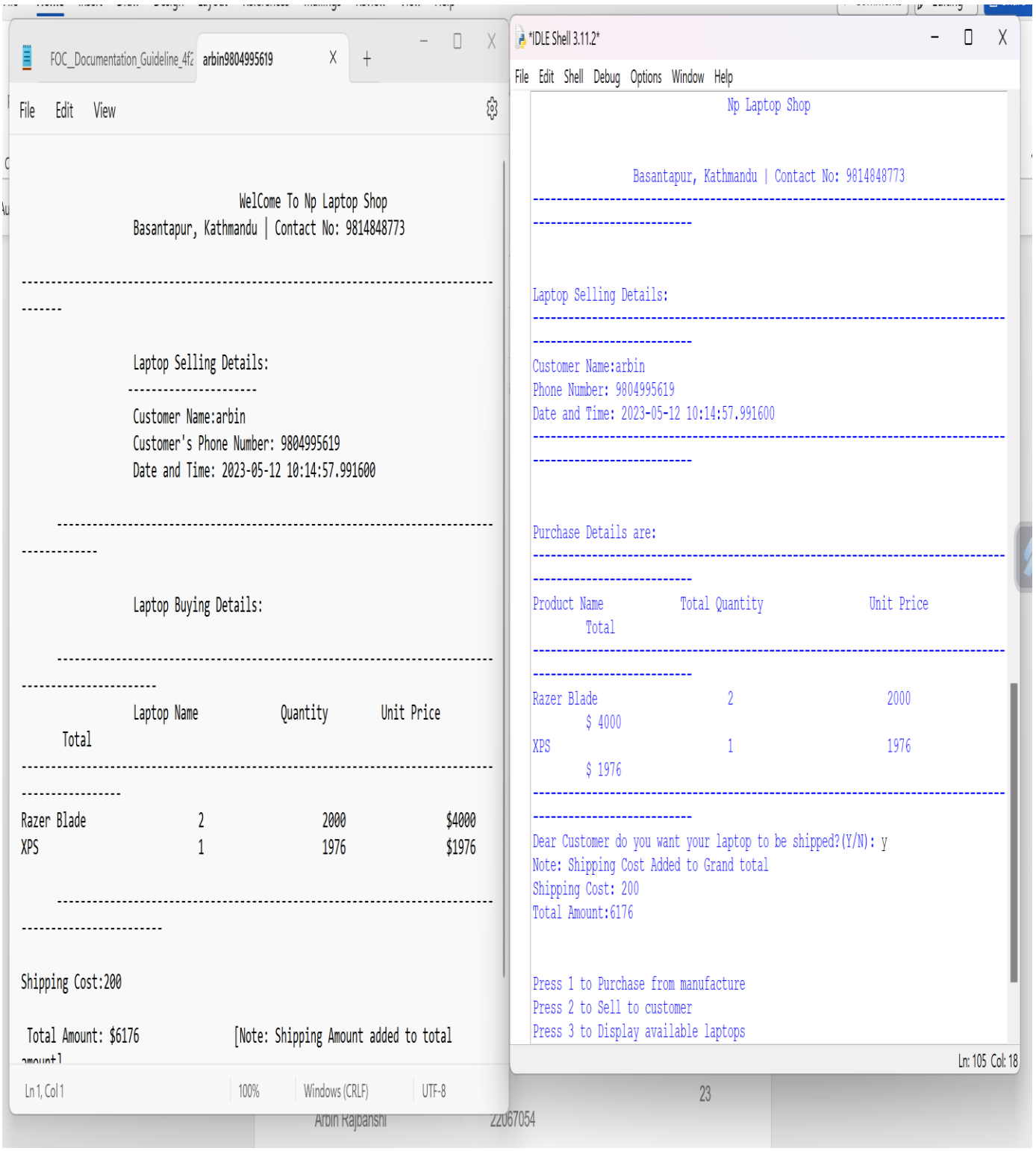


Figure 12: sell proof with screenshot

5.1.5 Test 5

Test	5
Objective	To show the update in stock of laptop
Action	• After user purchase laptops the quantity of laptop increase in laptop stock • After user purchase laptops the quantity of laptop decrease in laptop stock
Expected Result	The stock of laptop is updated
Actual Result	The stock of laptop is updated
Conclusion	The test was successful

Table 5:To show the update in stock of laptop

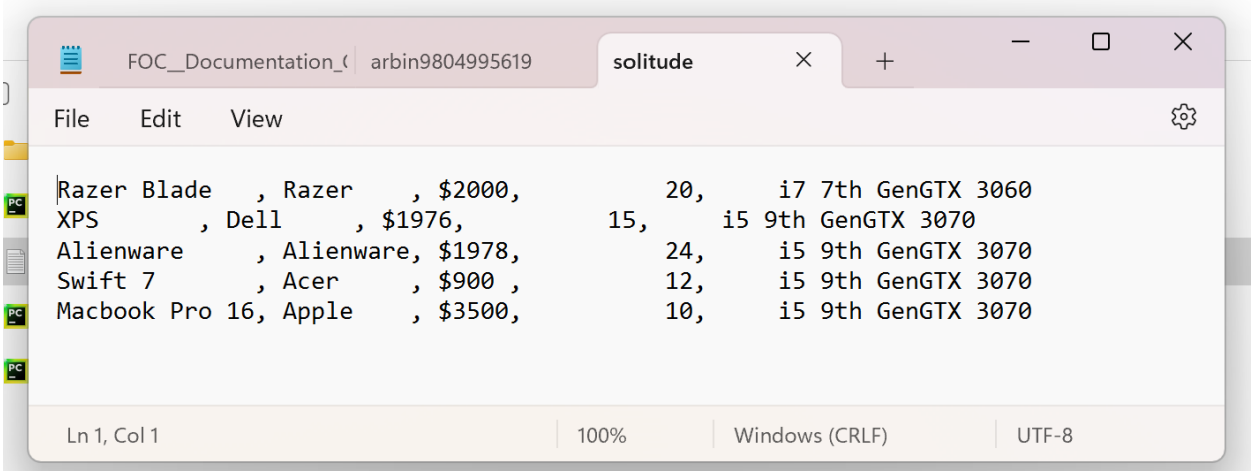
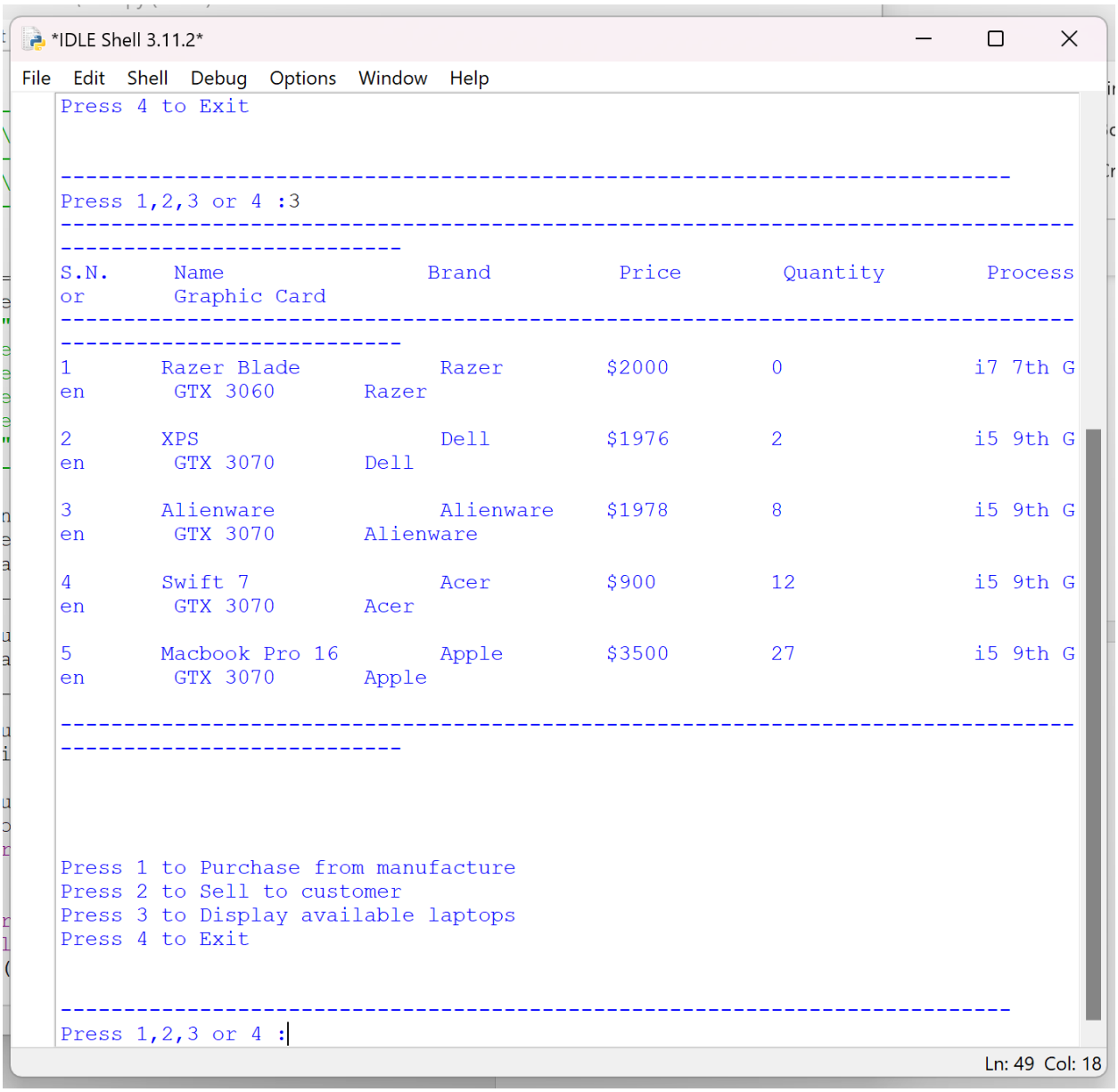


Figure 13: Screenshot of Set Credit Limit Action


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Press 4 to Exit
Press 1,2,3 or 4 :3
S.N. Name Brand Price Quantity Process
or Graphic Card
1 Razer Blade Razer $2000 0 i7 7th G
en GTX 3060 Razer
2 XPS Dell $1976 2 i5 9th G
en GTX 3070 Dell
3 Alienware Alienware $1978 8 i5 9th G
en GTX 3070 Alienware
4 Swift 7 Acer $900 12 i5 9th G
en GTX 3070 Acer
5 Macbook Pro 16 Apple $3500 27 i5 9th G
en GTX 3070 Apple
Press 1 to Purchase from manufacture
Press 2 to Sell to customer
Press 3 to Display available laptops
Press 4 to Exit
Press 1,2,3 or 4 :
Ln: 49 Col: 18

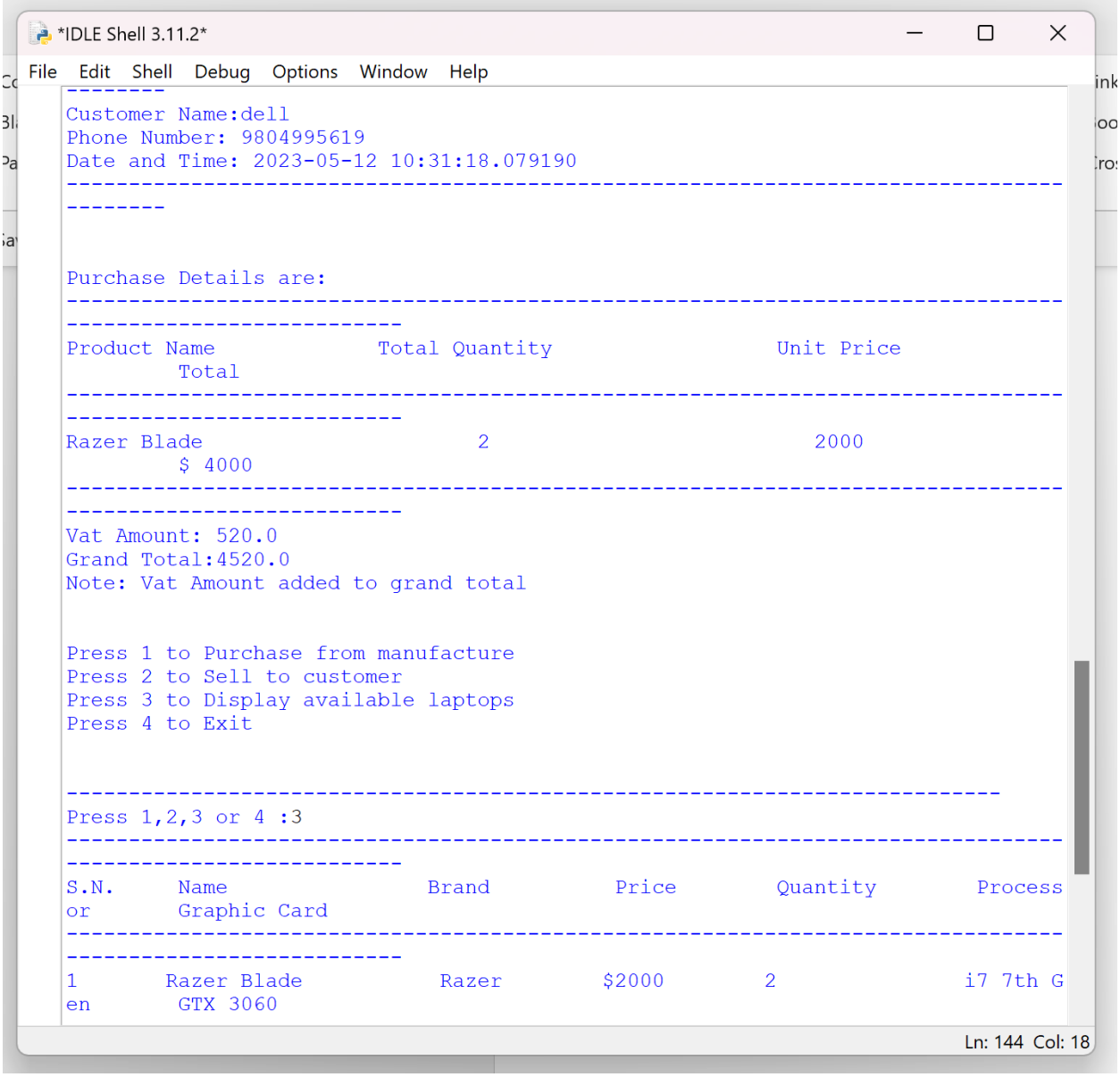
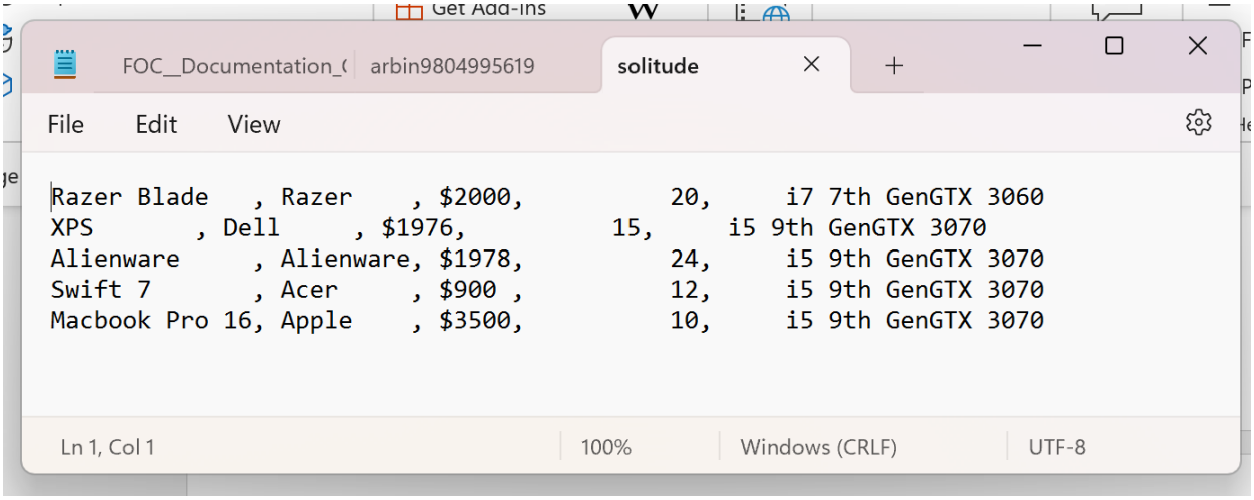


Figure 14: Screenshot of Cancel Credit Card Action



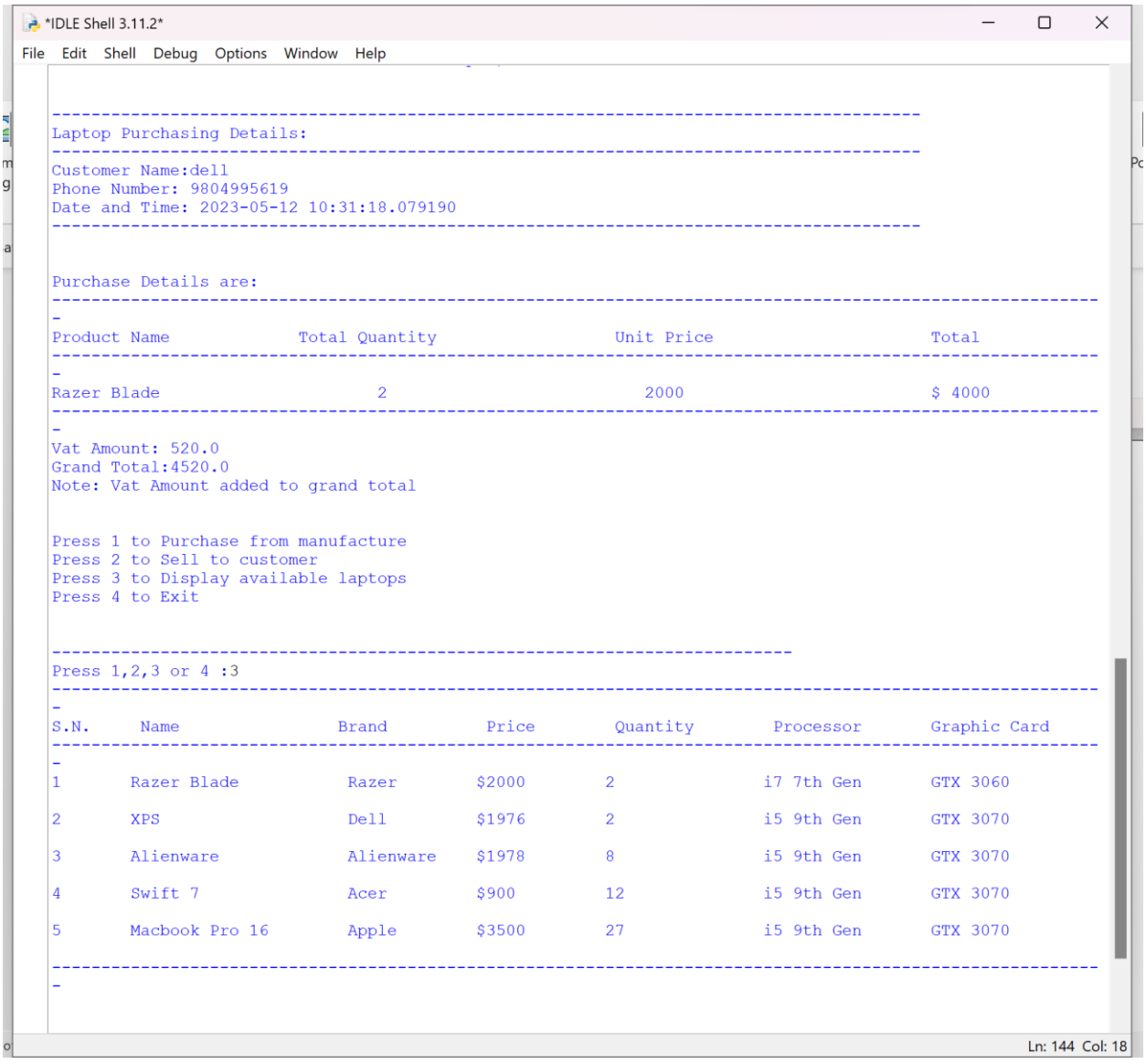


Figure 15: Screenshot of inspection of Debit Card without any values

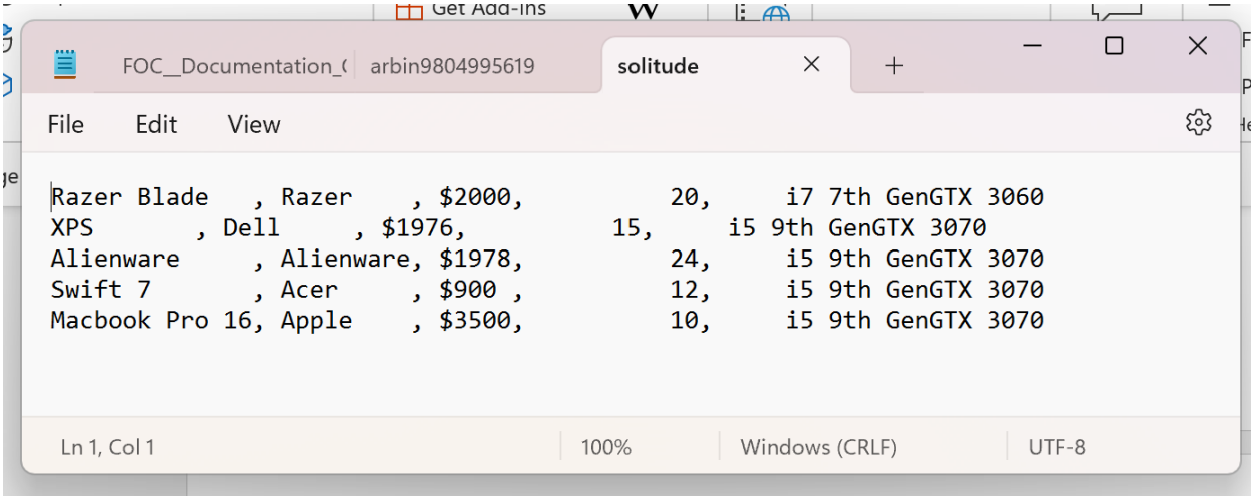


Figure 16: Screenshot of inspection of Debit Card with String values instead of numeric values

5.2 Conclusion

The course work was finished with bunches of exertion and difficulties. Huge loads of research and analysis were directed to traverse each progression of the work. Routine meetups with educators were conducted where reformist work was shown and significant ideas were embraced. With cycle of the above advances over and over, at last, an executable program playing out the allotted task was created. This course work helped a great deal in transforming my stereotype on programmers and the programming world. This assessment has developed exploration and investigation abilities to cut strides for development of a program, choosing the most ideal approach to foster the program, troubleshooting mistakes produced during compilation. The coursework likewise required an appropriately recorded report of the relative multitude of the program developed and ideas used in development of a functional laptop store program. As documentation of any task serves to outline and comprehend the assignment, it was exceptionally valuable and supportive in building our expertise of report creation in a standard practiced globally. Using time productively and multitasking to finish allotted coursework before the cut-off time was perhaps the main lesson gained from this assessment. The course work was a strong errand to be finished under set time. All necessary exploration, examination, arranging, improvement, testing, and reporting the interaction was to be done inside that time span. It was an exceptionally close timetable as course work of other subject were likewise to be finished in those equivalent weeks. There were restless evenings and steady programming musings going through my brain in those weeks. Notwithstanding lingering and overpowering inclination, I had to keep my focus on completions and place procrastination aside to focus only on my work. Looking behind starting here, it causes me to feel great that I have finished the job on schedule. To summarize the 1st coursework of Fundamentals of computing where a library executive’s framework was to be created utilizing Python

programming language, was effectively finished. All things considered, this coursework helps me a lot to understand about laptop shop how it work and function.

5.3 Appendix

Main.py

```
from operation import buyLaptop

from operation import sellLaptop

from write import W_laptoppurchase, W_laptopsell


def displayLaptops():
    print("-------------------------------------------------------------------------------------------------------")
    print("S.N. \t Name \t\t     Brand \t    Price \t Quantity \t Processor \t Graphic Card")
    print("-------------------------------------------------------------------------------------------------------")
    a = 1
    file = open("solitude.txt", "r")
    for line in file:
        print(a, "\t" + line.replace(",", "\t"))
        a = a + 1
    print("-------------------------------------------------------------------------------------------------------")
    file.close()
    print("\n")


print("\n")
print("-------------------------------------------------------------------------------------------------------")
print("\t \t \t \t Welcome to Np Laptop Shop")
print("-------------------------------------------------------------------------------------------------------")
```

```
print("\t \t \t Address: Basantapur, Kathmandu | Contact: 987456321")
print("-----------------------------------------------------------------------------------------------------------")
print("\n")


continueLoop = True
while continueLoop:
    print("\n")
    print("Press 1 to Purchase from manufacture")
    print("Press 2 to Sell to customer")
    print("Press 3 to Display available laptops")
    print("Press 4 to Exit")
    print("\n")
    print("-------------------------------------------------------------------------")
    try:
        userinput = int(input("Press 1,2,3 or 4 :"))
        if userinput == 1:
            name,phone,date_time,purchased_laptops,VAT,final_total = buyLaptop()
            W_laptoppurchase(name,phone,date_time,purchased_laptops,VAT,final_total)

        elif userinput == 2:
            name,phone,date_time,laptop_sold,cost_of_shipping,final_total = sellLaptop()
            W_laptopsell(name,phone,date_time,laptop_sold,cost_of_shipping,final_total)

        elif userinput == 3:
            displayLaptops()

        elif userinput == 4:
            continueLoop = False
```

```
            print("Thank you for Visiting US")


        else:
            print("Enter correct option. Try Again")
    except ValueError:
        print("Invalid input. Please Enter a number from 1, 2, 3 or 4.")
```

operation.py

```
from read import read_file
from datetime import datetime


def buyLaptop():
    print("Thank you for Purchasing With US")
    print("\n")
    print("-------------------------------------------------------------------")
    print("Please Enter your name and phone number")
    print("-------------------------------------------------------------------")
    print("\n")
    # Input validation for name
    while True:
        try:
            name = input("Enter Brand Name: ")
            if not name.isalpha():
                raise ValueError("Please provide a valid name !!")
            break
        except ValueError as e:
```

```
            print(e)
            print("\n")


    # Input validation for phone number
    while True:
        try:
            phone = input("Enter your Phone Number: ")
            if not phone.isdigit() or len(phone) != 10:
                raise ValueError("Please provide a valid Phone Number !!")
            break
        except ValueError as e:
            print(e)
            print("\n")
    print("-----------------------------------------------------------------------------------------------------
--")
    print("S.N. \t Name \t\t     Brand \t    Price \t Quantity \t Processor \t Graphic Card")
    print("-----------------------------------------------------------------------------------------------------
--")
    a = 1
    file = open("solitude.txt", "r")
    for line in file:
        print(a, "\t" + line.replace(",", "\t"))
        a = a + 1
    print("-----------------------------------------------------------------------------------------------------
--")
    file.close()
    print("\n")


    purchased_laptops = []
```

```
    while True:
        valid_id = int(input("Please provide the ID of the product you want to Purchase:"))
        print("\n")

        # Valid ID
        while valid_id <= 0 or valid_id > len(read_file()):
            print("Please provide a valid Laptop ID !!")
            print("\n")
            valid_id = int(
                input("Please provide the ID of the laptop you want to Purchase: "))

        while True:
            try:
                user_quantity = input("quantity of laptop you wanna order: ")
                if not user_quantity.isdigit():
                    raise ValueError("Please provide quantity of laptop in number!")
                break
            except ValueError as e:
                print(e)
        print("\n")

        # Valid Quantity
        my_dict = read_file()
        get_quantity = my_dict[valid_id][3]

        # Update the text file
        my_dict[valid_id][3] = int(my_dict[valid_id][3]) + int(user_quantity)
        file = open("solitude.txt", "w")
```

```
    for values in my_dict.values():
        file.write(str(values[0])+"," + str(values[1])+"," + str(values[2]) +
                   "," + str(values[3])+"," + str(values[4])+"," + str(values[5]))
        file.write("\n")
    file.close()

    # Purchasing from manufacturer
    product_name = my_dict[valid_id][0]
    quantity_of_user = user_quantity
    price_of_unit = my_dict[valid_id][2]
    price_of_item = my_dict[valid_id][2].replace("$", '')
    final_price = int(price_of_item)*int(quantity_of_user)
    purchased_laptops.append(
        [product_name, quantity_of_user, price_of_item, final_price])

    more_purchase = input("Do you want to Purchase more laptops? (Y/N): ")
    if more_purchase.lower() == 'n':
        break
    elif more_purchase.lower() == 'y':
        continue
    else:
        print("Invalid input. Please enter 'Y' or 'N'.")
        more_purchase = input("Do you want to Purchase more laptops? (Y/N): ")
    if more_purchase.lower() == 'n':
        break
    elif more_purchase.lower() == 'y':
        continue
    else:
```

```
            print("Invalid input. Exiting loop.")
            break


    total = 0
    for i in purchased_laptops:
        total += int(i[3])
        VAT = 0.13*(total)
    final_total = total + VAT
    date_time = datetime.now()
    print("\n")
    print("\t \t \t \t WelCome TO Np Laptop Shop")
    print("\n")
    print("\t \t\t\t\t\   Basatapur, Kathmandu | Contact No: 98148484779 ")
    print("\n")
    print("-------------------------------------------------------------------------------------------")
    print("Laptop Purchasing Details: ")
    print("-------------------------------------------------------------------------------------------")
    print("Customer Name:" + str(name))
    print("Phone Number: " + str(phone))
    print("Date and Time: " + str(date_time))
    print("-------------------------------------------------------------------------------------------")
    print("\n")
    print("Purchase Details are:")
    print("-------------------------------------------------------------------------------------------------------
--")
    print("Product Name \t\t Total Quantity \t\t Unit Price \t\t\t Total")
    print("-------------------------------------------------------------------------------------------------------
--")
    for i in purchased_laptops:
```

```
        print(i[0], "\t\t\t", i[1], "\t\t\t", i[2], "\t\t\t", "$", i[3])
    print("-------------------------------------------------------------------------------------------------------
--")


    print("Vat Amount:",  VAT)
    print("Grand Total:" + str(final_total))
    print("Note: Vat Amount added to grand total")


    return name, phone, date_time, purchased_laptops, VAT, final_total



def sellLaptop():
    print("Thank you for Selling US")
    print("\n")
    print("----------------------------------------------------------------------")
    print("We will need your Name and Phone Number to print bill")
    print("----------------------------------------------------------------------")
    print("\n")
    # Input validation for name
    while True:
        try:
            name = input("Enter your Name: ")
            if not name.isalpha():
                raise ValueError("Please provide a Valid Name !!")
            break
        except ValueError as e:
            print(e)
            print("\n")
    # Input validation for phone number
```

```
    while True:
        try:
            phone = input("Enter Your Phone Number: ")
            if not phone.isdigit() or len(phone) != 10:
                raise ValueError("Please provide a valid Phone Number !!")
            break
        except ValueError as e:
            print(e)
            print("\n")
    print("-------------------------------------------------------------------------------------------------------")
    print("S.N. \t Name \t\t      Brand \t   Price \t Quantity \t Processor \t Graphic Card")
    print("-------------------------------------------------------------------------------------------------------")
    a = 1
    file = open("solitude.txt", "r")
    for line in file:
        print(a, "\t" + line.replace(",", "\t"))
        a = a + 1
    print("-------------------------------------------------------------------------------------------------------")
    file.close()
    print("\n")


    laptop_sold = []


    while True:
        valid_id = int(input( "Please Provide the ID of the laptop you want to sell (Enter 9 to Exit): "))
        if valid_id == 9:
```

```
            break

        # Valid ID

        while valid_id <= 0 or valid_id > len(read_file()):
            print("Please Provide a valid Laptop ID !!")
            print("\n")
            valid_id = int(
                input("Please Provide the ID of the laptop you want to sell: "))

        user_quantity = int(
            input("Please Provide the number of quantity of the laptop you want to sell: "))
        print("\n")

        # Valid Quantity

        Mylap_dict = read_file()
        get_quantity = Mylap_dict[valid_id][3]
        while user_quantity <= 0 or user_quantity > int(get_quantity):
            print("Dear Admin, the quantity you are looking for is out of stock in our shop.
Please look again in the Laptop screen")
            print("\n")
            user_quantity = int(
                input("Please Provide the Number of quantity of the Laptop you want to sell:
"))
        print("\n")

        # Update the text file
```

```
        Mylap_dict[valid_id][3] = int(Mylap_dict[valid_id][3]) - int(user_quantity)

        file = open("solitude.txt", "w")

        for values in Mylap_dict.values():
            file.write(str(values[0])+"," + str(values[1])+"," + str(values[2]) +
                       "," + str(values[3])+"," + str(values[4])+"," + str(values[5]))
            file.write("," + str(values[1].split()[0]))
            file.write("\n")
        file.close()

        # getting user purchased items

        product_name = Mylap_dict[valid_id][0]
        quantity_of_user = user_quantity
        price_of_unit = Mylap_dict[valid_id][2]
        price_of_item = Mylap_dict[valid_id][2].replace("$", '')
        final_price = int(price_of_item)*int(quantity_of_user)

        laptop_sold.append(
            [product_name, quantity_of_user, price_of_item, final_price])

        sell_more = input("Do you want to sell more laptops? (Y/N): ")
        if sell_more.lower() == 'n':
            break
        elif sell_more.lower() == 'y':
            continue
        else:
```

```
            print("Invalid input. Please enter 'Y' or 'N'.")
            sell_more = input("Do you want to Purchase more laptops? (Y/N): ")
        if sell_more.lower() == 'n':
            break
        elif sell_more.lower() == 'y':
            continue
        else:
            print("Invalid input. Exiting loop.")
            break


    date_time = datetime.now()
    # shipping_Cost = input(
    #     "Dear Customer do you want your laptop to be shipped? (Y/N)")


    print("\n")
    print("\t \t \t \t Np Laptop Shop")
    print("\n")
    print("\t \t Basantapur, Kathmandu | Contact No: 9814848773 ")
    print("-------------------------------------------------------------------------------------------------------
--")
    print("\n")
    print("Laptop Selling Details: ")
    print("-------------------------------------------------------------------------------------------------------
--")
    print("Customer Name:" + str(name))
    print("Phone Number: " + str(phone))
    print("Date and Time: " + str(date_time))
    print("-------------------------------------------------------------------------------------------------------
--")
```

```
    print("\n")
    print("Purchase Details are:")
    print("-------------------------------------------------------------------------------------------------------")
    print("Product Name \t\t Total Quantity \t\t Unit Price \t\t\t Total")
    print("-------------------------------------------------------------------------------------------------------")

    for i in laptop_sold:
        print(i[0], "\t\t\t", i[1], "\t\t\t", i[2], "\t\t\t", "$", i[3])
    print("-------------------------------------------------------------------------------------------------------")

    total = 0
    shipping_Cost = 0
    shipping_choice = input(
        "Dear Customer do you want your laptop to be shipped?(Y/N): ")
    if shipping_choice.upper() == "Y":
        shipping_Cost = 200
        print("Note: Shipping Cost Added to Grand total")
        print("Shipping Cost:", shipping_Cost)

    for i in laptop_sold:
        total += int(i[3])
    final_total = total + shipping_Cost
    print("Total Amount:" + str(final_total))

    return name, phone, date_time, laptop_sold, shipping_Cost,final_total
```

read.py

```
def read_file():
    file = open("solitude.txt","r")
    laptop_sn= 1
    MyLap_dict = {}
    for line in file:
        line = line.replace("\n","")
        MyLap_dict[laptop_sn] = (line.split(","))
        laptop_sn += 1
    file.close()
    return MyLap_dict
```

write.py

```
def W_laptopsell(name, phone, date_time, laptop_sold, shipping_Cost, final_total):
    with open(str(name) + str(phone) + ".txt", "w")as file:
        file.write("\n")
        file.write("\t \t \t \t\t WelCome To Np Laptop Shop")
        file.write("\n")
        file.write("\t \t \t Basantapur, Kathmandu | Contact No: 9814848773 ")
        file.write("\n\n")
        file.write(
            "-------------------------------------------------------------------------------------\n")
        file.write("\n")
        file.write("\t\t\t Laptop Selling Details: ")
        file.write("\n")
        file.write("\t\t\t----------------------\n")
        file.write("\t\t\t Customer Name:" + str(name))
```

```
        file.write("\n")
        file.write("\t\t\t Customer's Phone Number: " + str(phone))
        file.write("\n")
        file.write("\t\t\t Date and Time: " + str(date_time))
        file.write("\n")
        file.write(
            "\t\t\t-----------------------------------------------------------------------------------------\n")
        file.write("\n")
        file.write("\t\t\t Laptop Buying Details:")
        file.write("\n")
        file.write(
            "\t\t\t-----------------------------------------------------------------------------------------
\n")
        file.write("\t\t\t Laptop Name \t\t  Quantity \t\t Unit Price \t\t Total")
        file.write("\n")
        file.write(
            "-----------------------------------------------------------------------------------------\n")
        for i in laptop_sold:
            file.write(str(i[0]) + "\t\t\t" + str(i[1]) + "\t\t\t" +
                       str(i[2]) + "\t\t\t" + "$" + str(i[3])+"\n")
        file.write(
            "\t\t-----------------------------------------------------------------------------------\n")
        file.write("\n")
        if shipping_Cost == 200:
            file.write("Shipping Cost:" + "" + str(shipping_Cost) + "\n")
            file.write("\n")
            file.write(" Total Amount: $" + str(final_total) + "\t\t\t"
                       "[Note: Shipping Amount added to total amount]" + "\n")
            file.write("\n")
```

```
        else:
            file.write("Total Amount: $" + str(final_total))


def W_laptoppurchase(name, phone, date_time, laptop_bought, VAT, final_total):
    with open(str(name) + str(phone) + ".txt", "w")as file:
        file.write("\n")
        file.write("\t \t \t \t\t Welcome To Np Laptop Shop")
        file.write("\n")
        file.write("\t \t\t Basatapur, Kathmandu | Phone No: 9814848789")
        file.write("\n")
        file.write(
            "\t\t-------------------------------------------------------------------------------------\n")
        file.write("\t\t Laptop Details: ")
        file.write("\n")
        file.write("\t\t-------------------------\n")
        file.write("\t\t Company Name:" + str(name))
        file.write("\n")
        file.write("\t\t Phone Number: " + str(phone))
        file.write("\n")
        file.write("\t\t Date and Time: " + str(date_time))
        file.write("\n")
        file.write(
            "\t\t -------------------------------------------------------------------------------------\n")
        file.write("\n")
        file.write("\t\t Laptop Buying Details:")
        file.write("\n")
        file.write(
```

```
        "\t\t-------------------------------------------------------------------------------------------------
\n")
    file.write("\t\t Laptop Name \t\t Quantity \t\t Unit Price \t\t Total")
    file.write("\n")
    file.write(
        "\t\t-------------------------------------------------------------------------------------------------
\n")
    for i in laptop_bought:
        file.write(str(i[0]) + "\t\t\t" + str(i[1]) + "\t\t\t" +
                   str(i[2]) + "\t\t\t" + "$" + str(i[3])+"\n")
    file.write(
        "\t\t-------------------------------------------------------------------------------------------------\n")
    file.write("\n")
    if VAT:
        file.write("\t\t Vat Amount:" + "" + str(VAT) + "\n")
        file.write("\n")
        file.write("\t\t Total Amount: $" + str(final_total) + "\n")
        file.write("\n")
        file.write("\t\t Note: Vat Amount added to total amount" + "\n")
        file.write("\n")
    else:
        file.write("Total Amount: $" + str(final_total))
```