December 2, 2021

Hardik Jain, Sanchit Jain, Palak Jain

Jaypee Institute of Information Technology

NOIDA

Inventory management and billing system

SDF Mini Project

Inventory Management and Billing System

# Details of Team Members:

* Hardik Jain – B62330
* Sanchit Jain – B65560
* Palak Jain – ECN21024

B7 CSE

# Abstract of Project:

In large shops and markets, it could be tedious to keep record of the stock of all items they have.

The function of this program is to help a store/shop to manage the items in their inventory. They can save quantity of each item in their inventory. Update the stock as soon as the sell a product or they order items for the store. They could also calculate the total value of all their products. They can check if something in their inventory is out of stock.

Other main function of the program is to help them create bills for customers. Those bills are saved in text files which could later be used by the store owner/manager as they wish. It also keeps record of the total sales. They can calculate the total profit they made in a particular duration of time.

The backbone of the project is file handling which saves the list of items in a text file. It could be opened in the program and the data could be loaded in multidimensional arrays. After performing the required operation, the data is saved back in the text file.

# Topics of SDF-I used:

* File Handling
* Structures
* 1D Arrays and Multi-Dimensional Arrays
* Basic print, scan functions and mathematical operators
* Functions
* Loops: For loop, while loop, nested loops
* If else, nested if else
* String Methods
* Switch Case
* Pointers

# Design of the project:

The program has a main menu from where user can enter a character choice to continue to do the specified action. The available choices are from A-M, and X to exit the program. These choices are as below:

1. List all inventory items: It prints all the data stored in the text file inventory.txt in a table format readable by the user.
2. Search inventory items: It takes a string keyword from the user as input, and check it against the list of all the items, the items whose name/barcode has the keyword as a substring are printed on the screen.
3. List product details: It takes the barcode of a product as user input and print all the details of the same item.
4. Create new item: It takes all the required details of a new item from the user and stores in the inventory.txt file.
5. Delete an item: It takes the barcode of an item as input and deletes the item whose barcode matches it, and stores the remaining items in the text file. It also decrements the serial numbers of remaining items accordingly.
6. Edit a product detail: Takes the old barcode as user input and then the user can give the item a new name/barcode or enter “-1” if he doesn’t want it to be changed.
7. Update product price: Takes barcode of an item as input and give users the option to update its price and then store the new price in the text file.
8. Update product quantity: Takes barcode and quantity as input, and then the new quantity is stored as the sum of old quantity and the number by which we want to increment it.
9. Update product quantity in bulk: Sometimes when a store receives the delivery of stock, they may want to update the quantity in bulk, so that they don’t have to chose the last option repeatedly, that’s why this option is provided which takes continuous input of barcode and quantity as input from user until they enter -1 to terminate the loop.
10. Create a new bill: Option to create a new bill, it takes details like Customer Name, Address, Contact number as input, then runs a loop to take input of items, their quantity and tax rates. Afterwards it converts all the data into a bill format which is saved in a separate invoice file to be used later such as taking print outs of the bill. The same data is saved in sales.txt file so that the program can access all sales data which could be used to print list of sales in a particular period of time.
11. List all sales: Just like the first option, this one also prints the data stored in sales.txt file in user understandable table format.
12. Search sales within date range: Helps to filter the sales on basis of a time period.
13. Print bill details: This option prints complete bill on the screen after taking the invoice number as input.

# Implementation Details

## C Libraries Used:

* String: For performing string functions like strstr, strcmp, strlen, strcat, strtok etc.
* Stdio: Basic library for performing all input output functions.
* Time: Used for time functions when creating a new bill
* Stdlib.h: Used for functions like atoi, atof etc.
* Conio: Used for the getch() function used in the main function to wait for key press by user to continue ahead.

## Constants Declared:

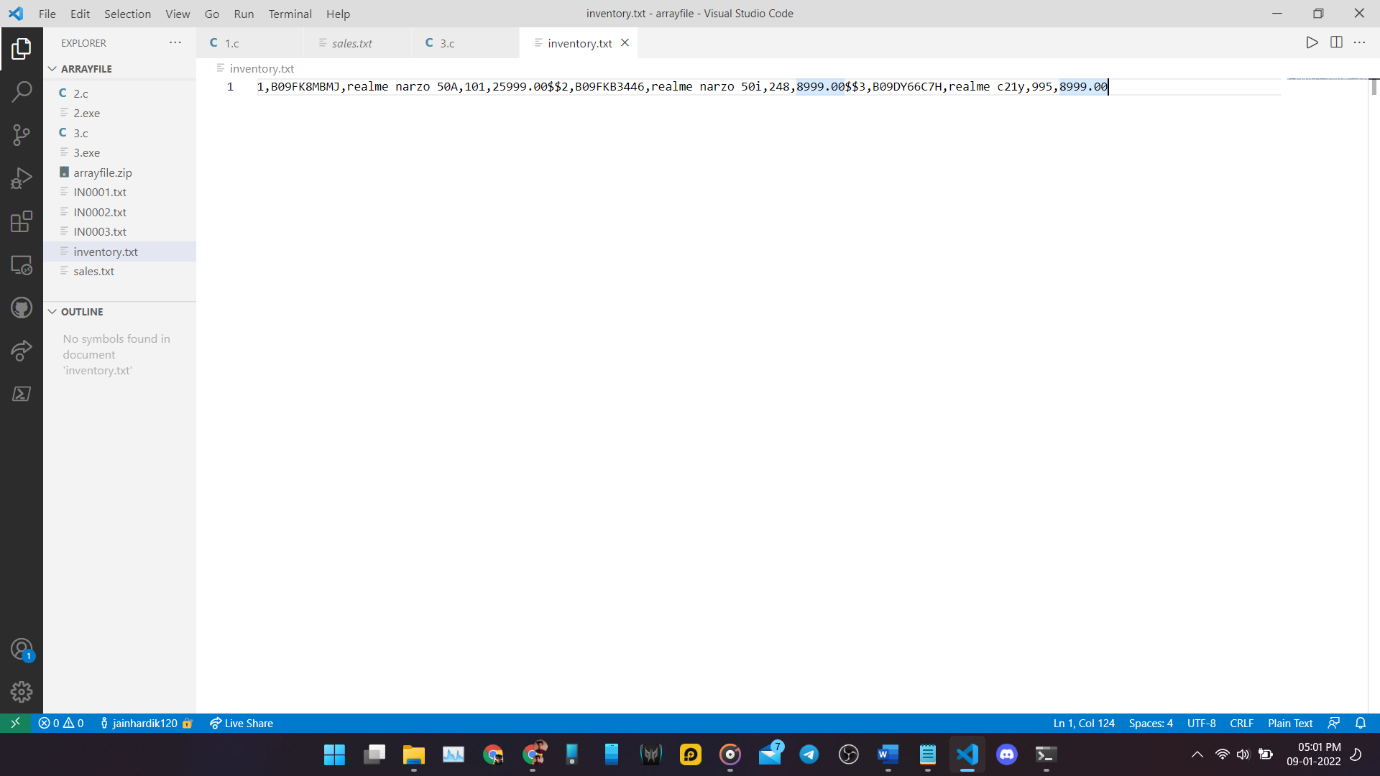
* Text Size and Number of Items/Sales: These all define the maximum limit of data stored in text files and are used to initialize arrays and strings of these size in the program.
* Company Details Constants: These all contains the details of billing company which is printed at the top of a bill.
* Blank Spaces, Linear Line: These are used in the printBillFormat function to align the text at the required position by introducing the required no of spaces, and printing separation line.

## Structures Used:

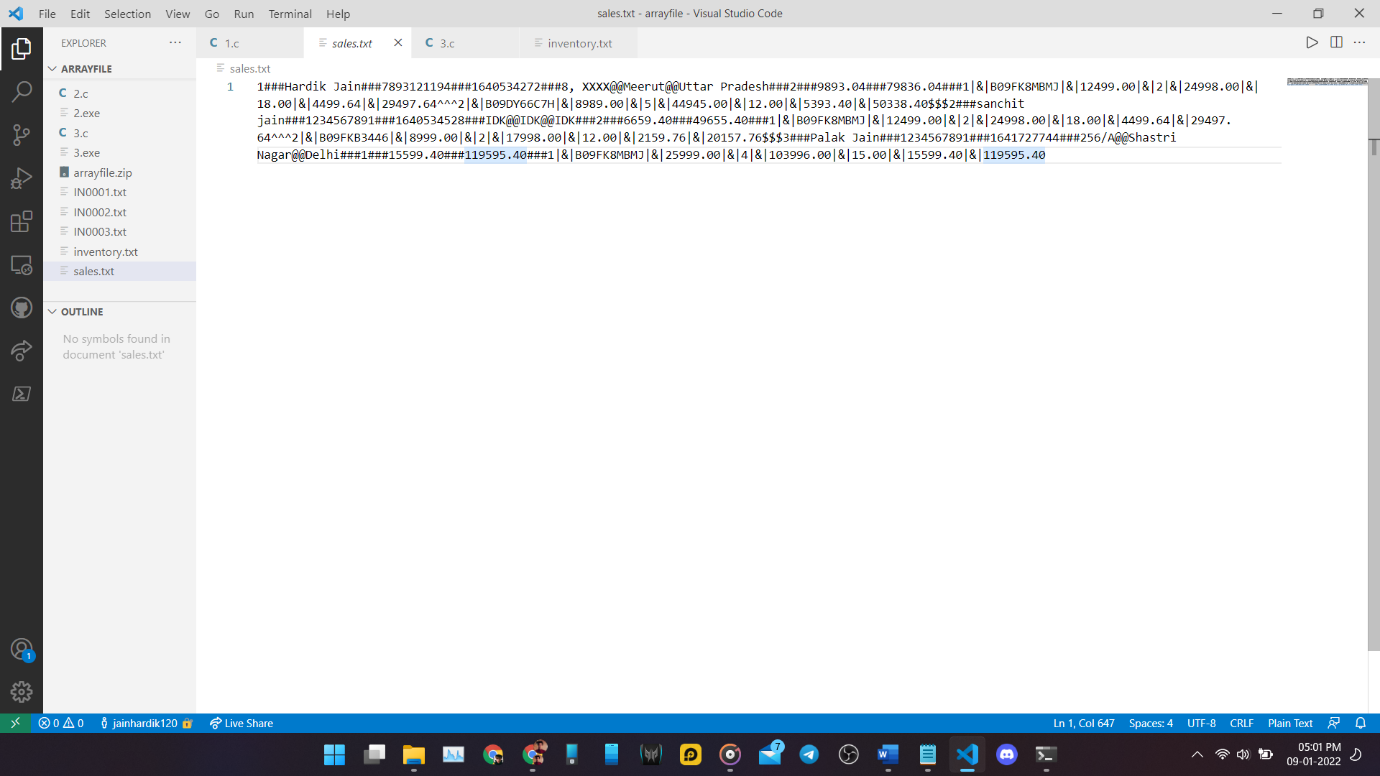
* Inventory item: This structure contains the details of a inventory item such as barcode, name, quantity, price etc.
* Sales Item: This structure is used as a part of the bill structure and contains the details of an item sold such as tax rate, quantity, amount, total amount etc.
* Bill: This structure contains details the customer, invoice number, time and date of purchase, an array of the previously defined sales item structure, and the total taxes and total amount floats.

# Screenshots of the program

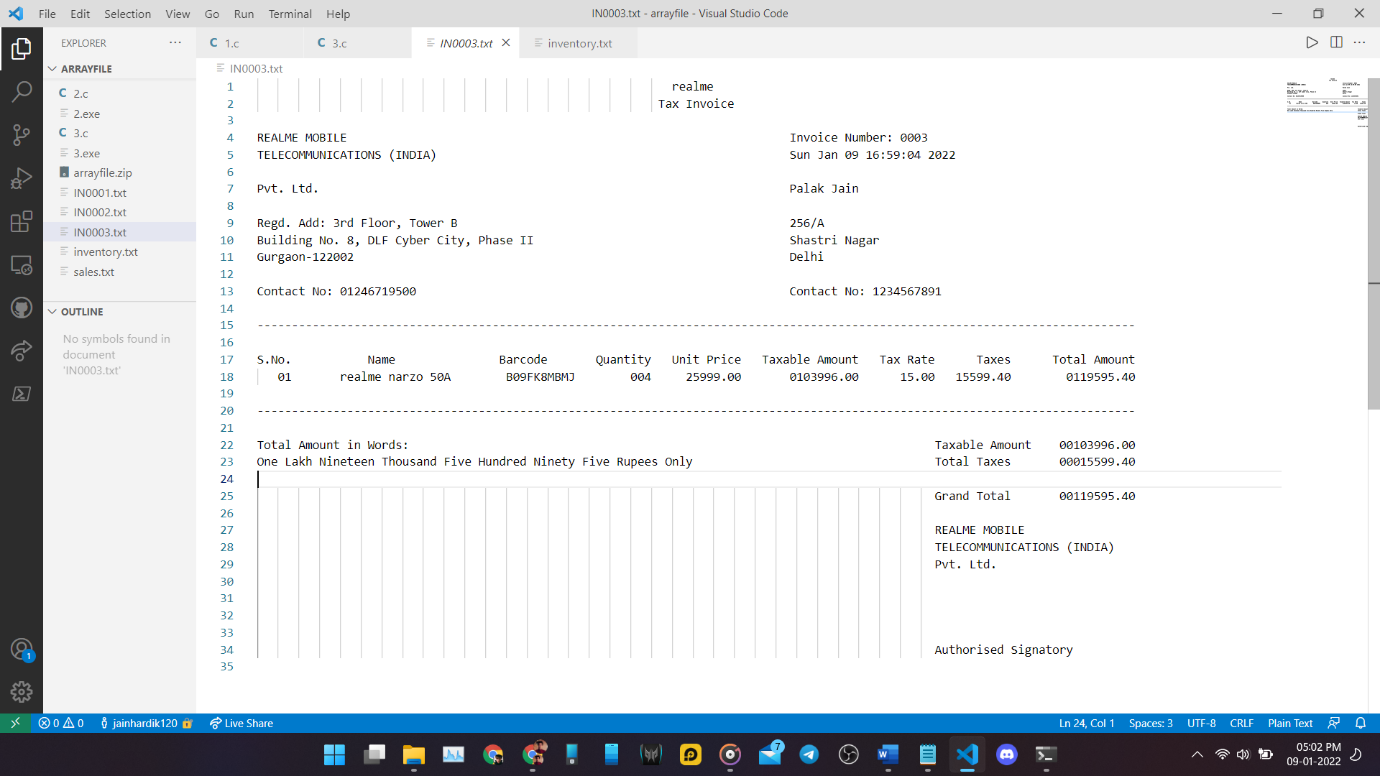
Data stored in inventory.txt file separated by delimiters $$



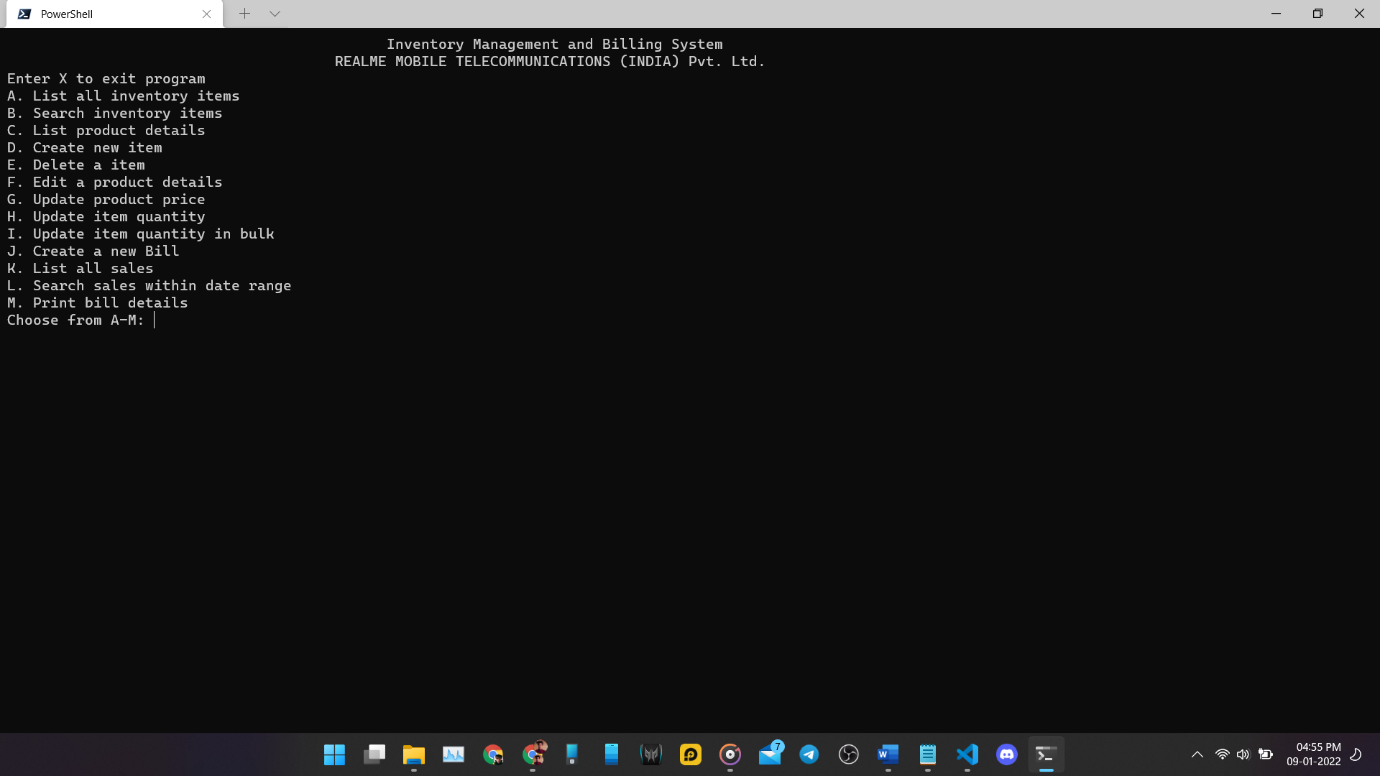
Data stored in sales.txt file separated by delimiters $$$, ###, ^^^, |&|



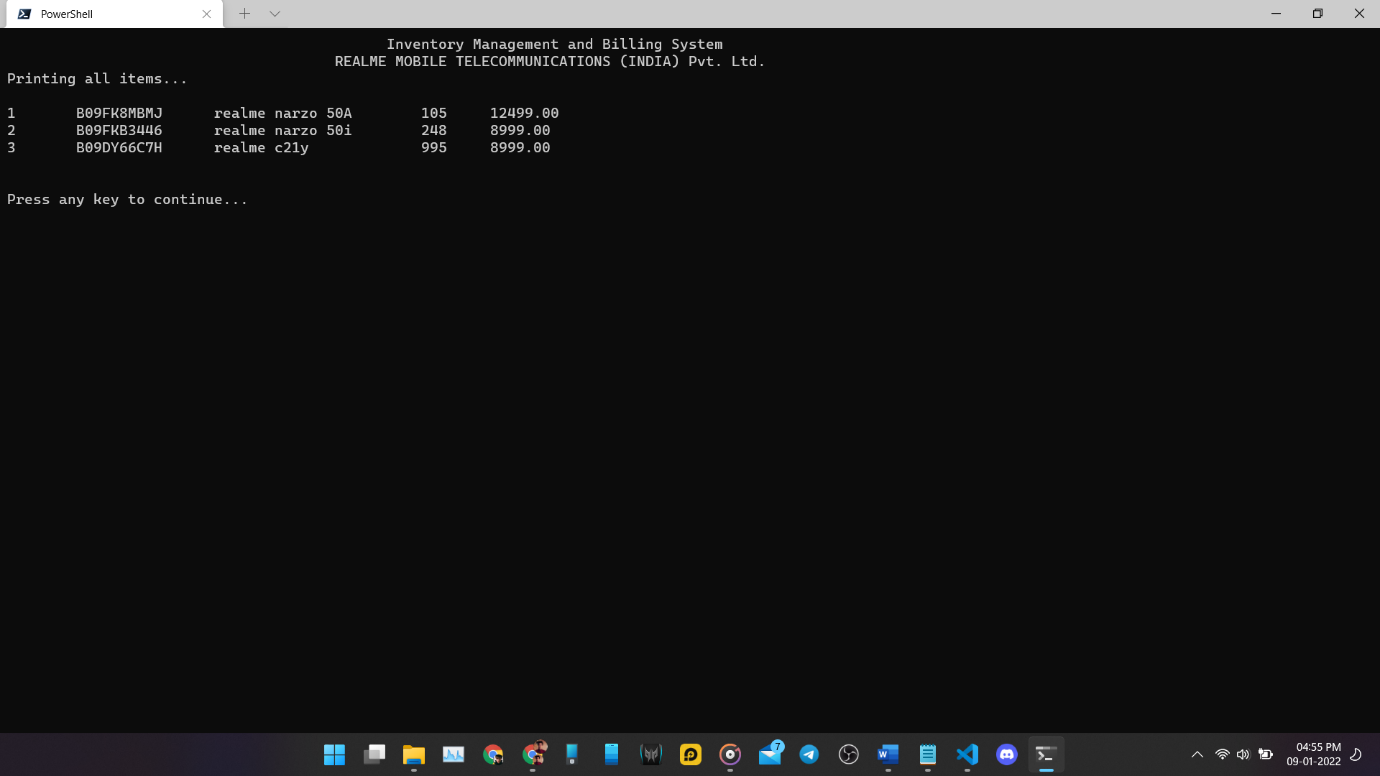
Sample Bill file IN0003.txt



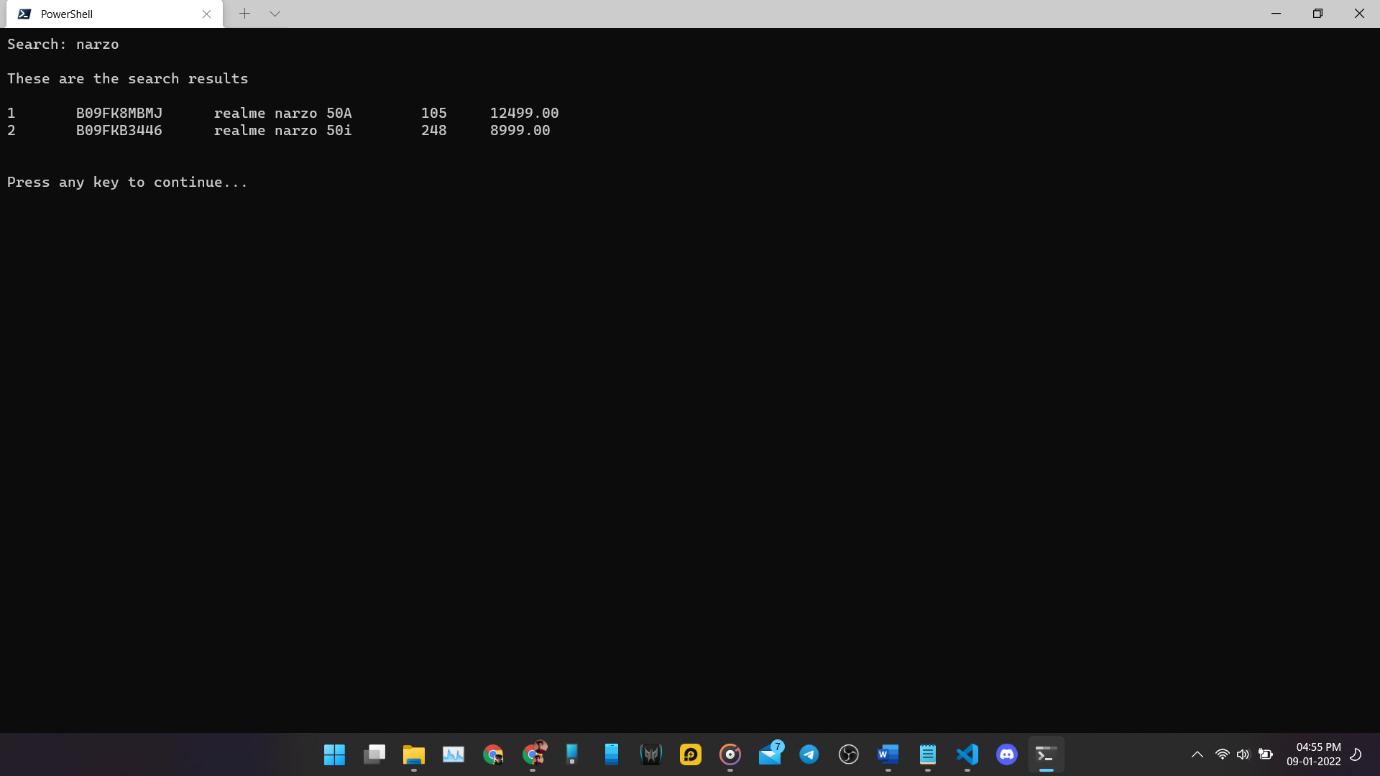
Entry Point of program:



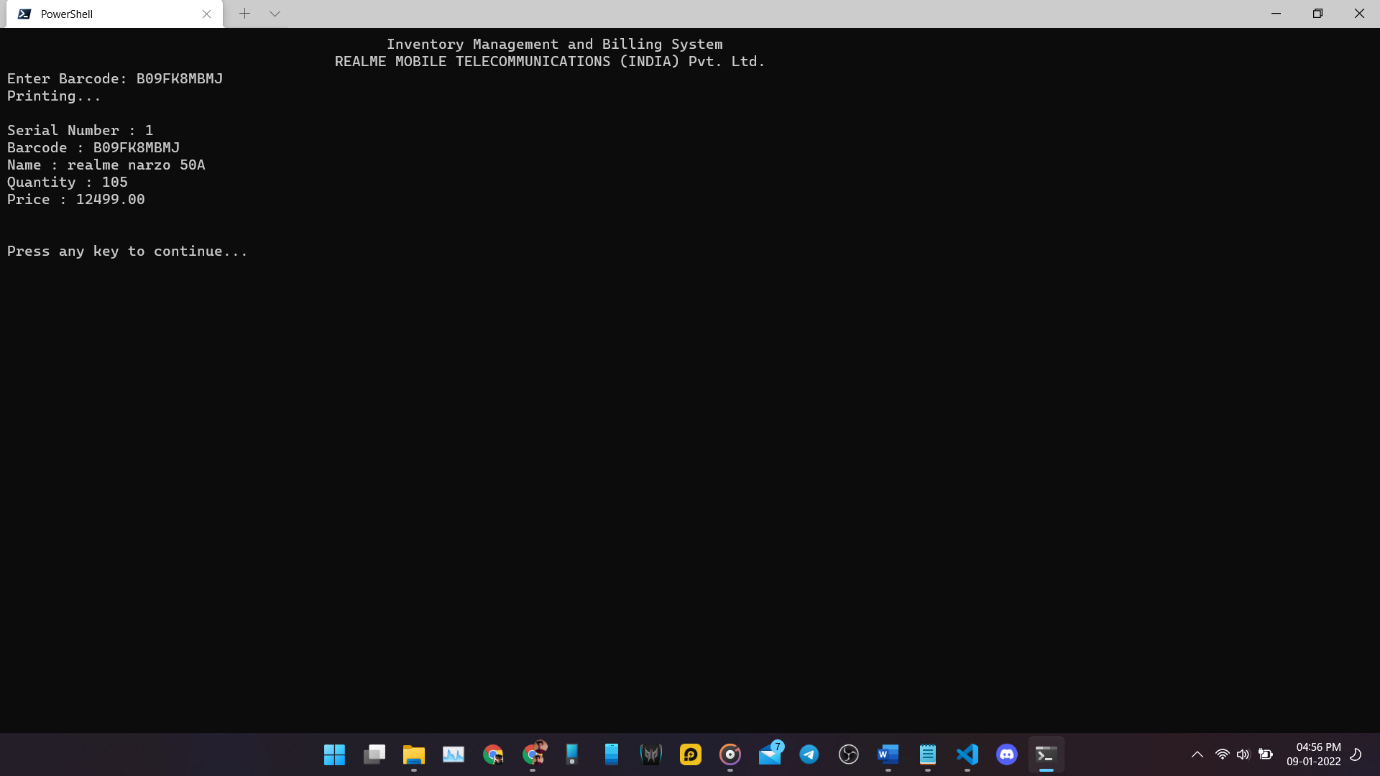
List all items function:



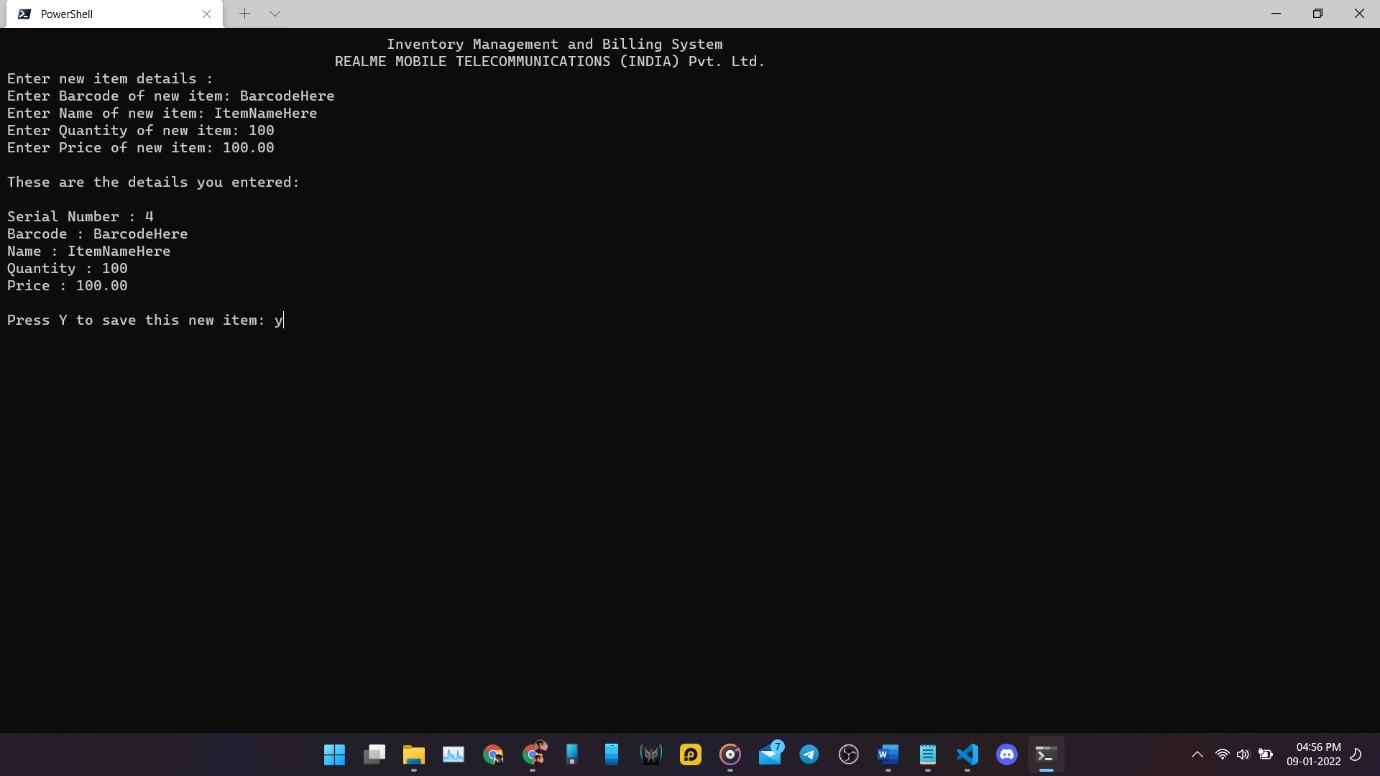
Search Inventory Items function:



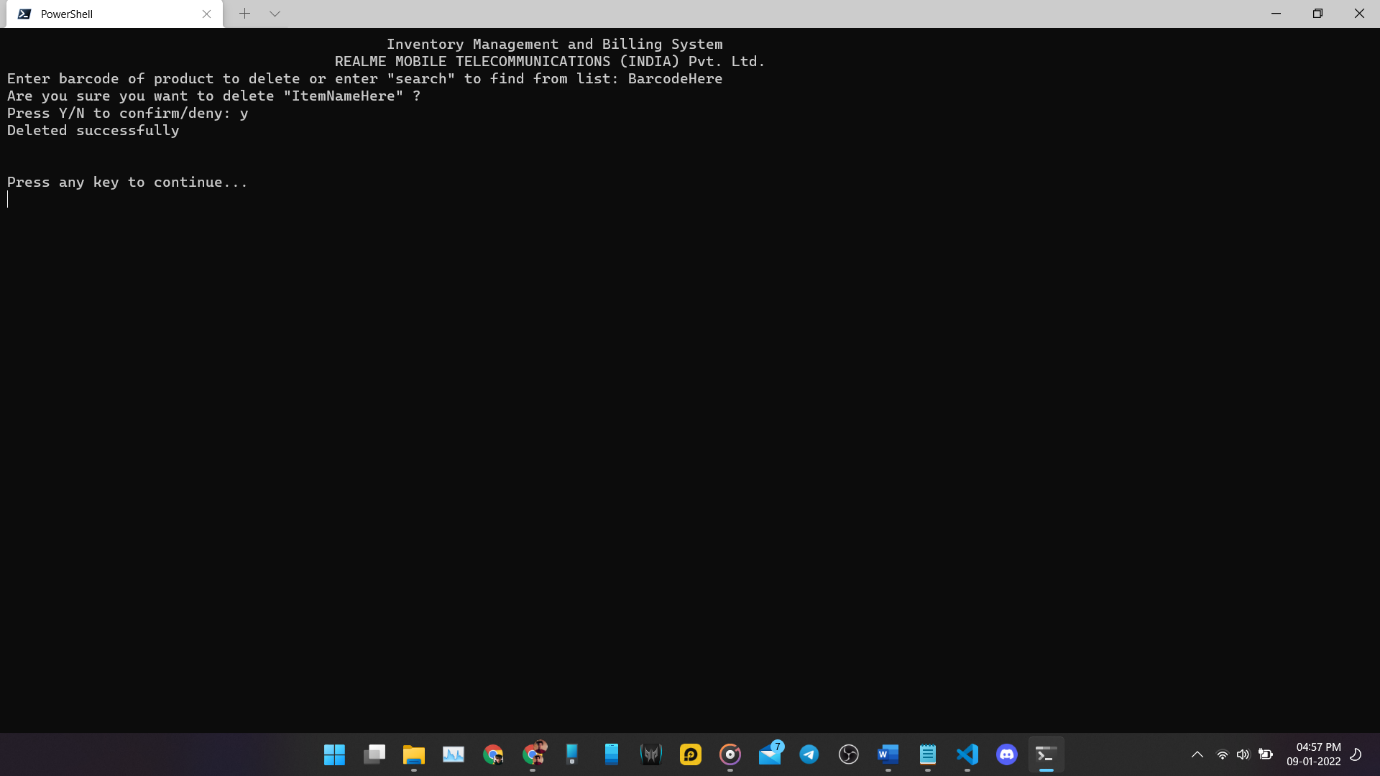
Print item detail’s function:



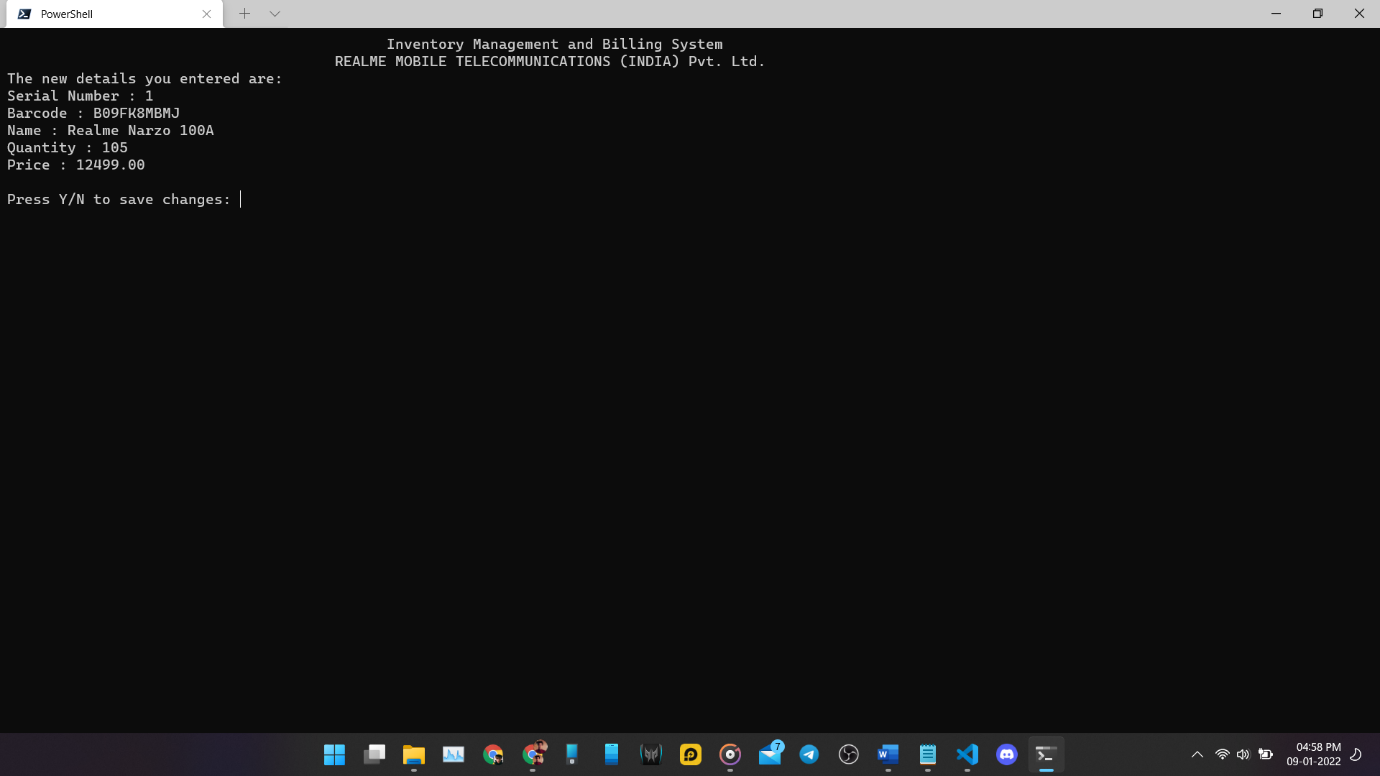
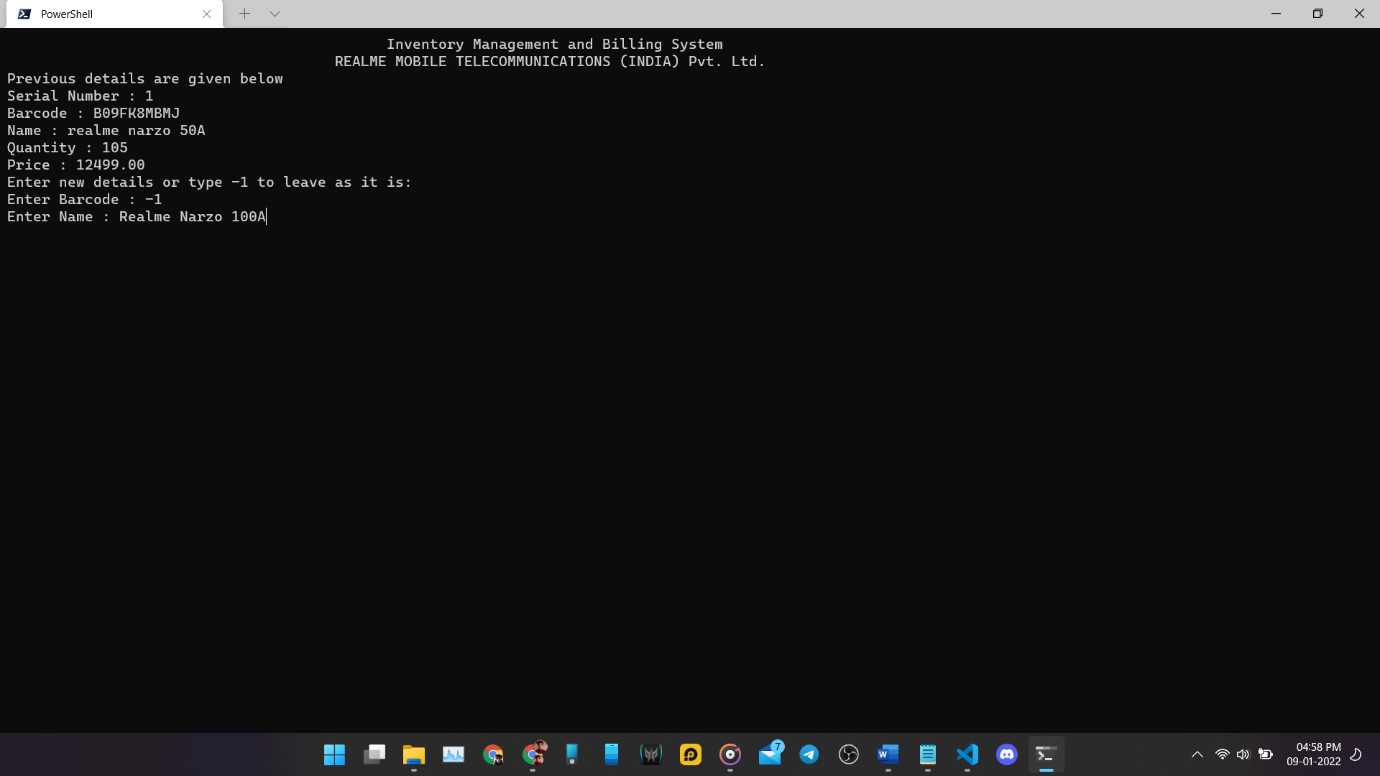
Create new item function:



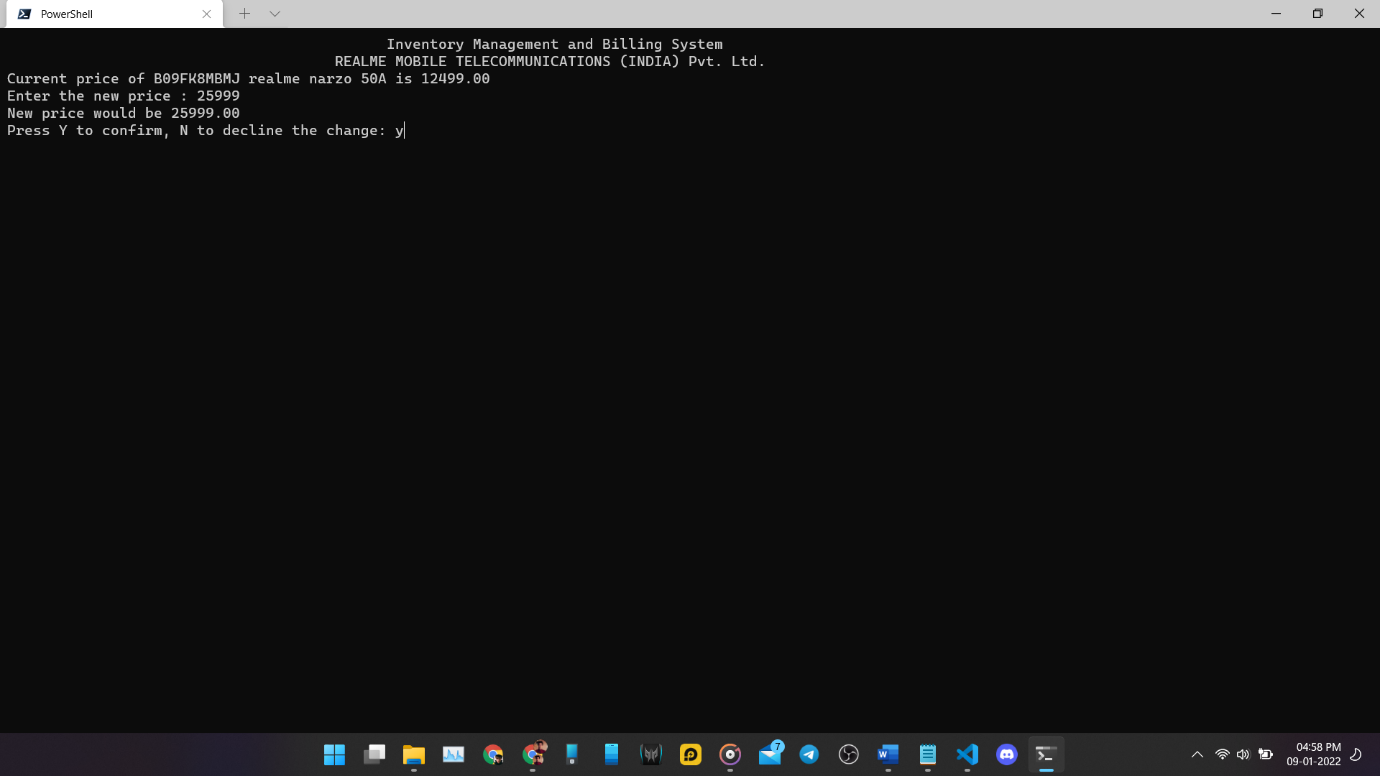
Delete item function:



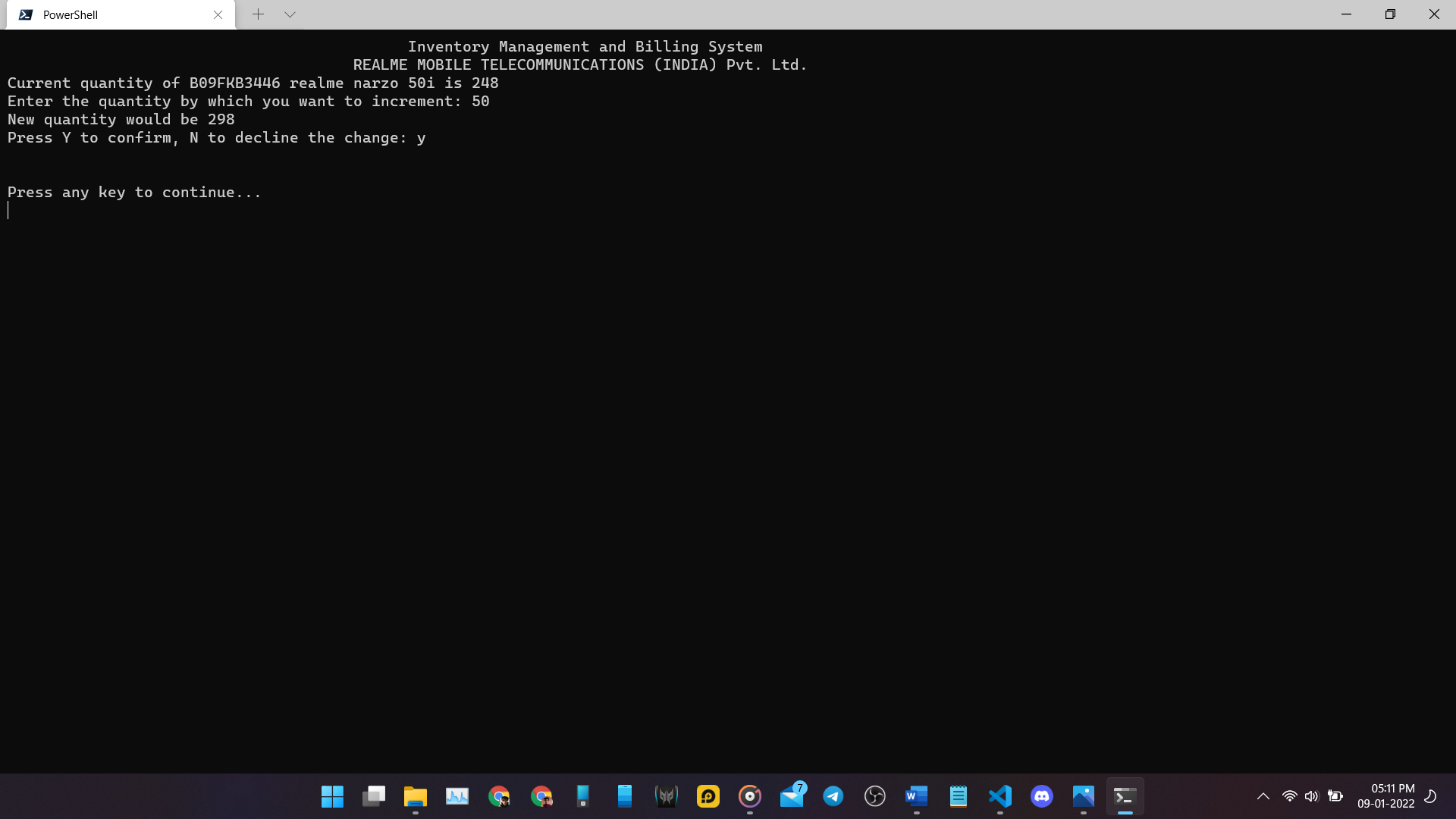
Update product details function:

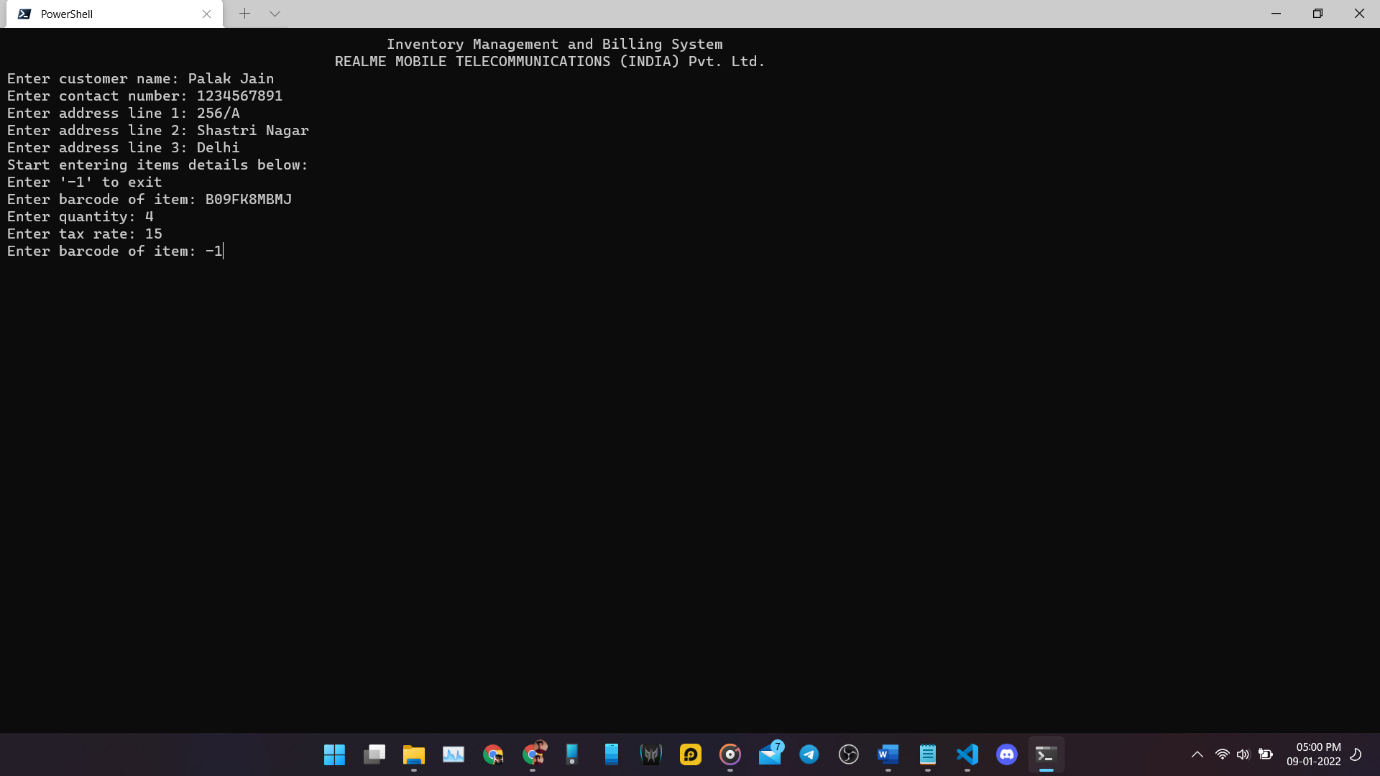


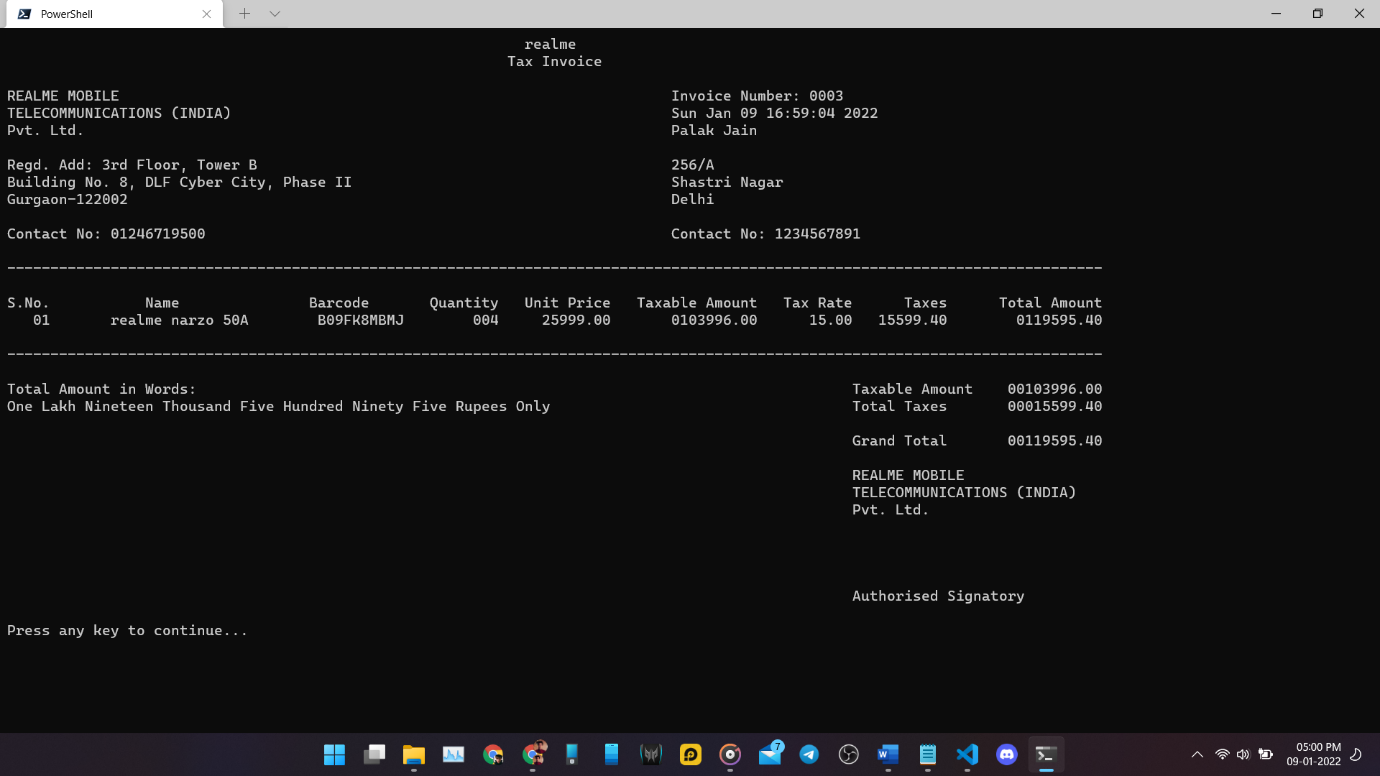
Update Product Price function:

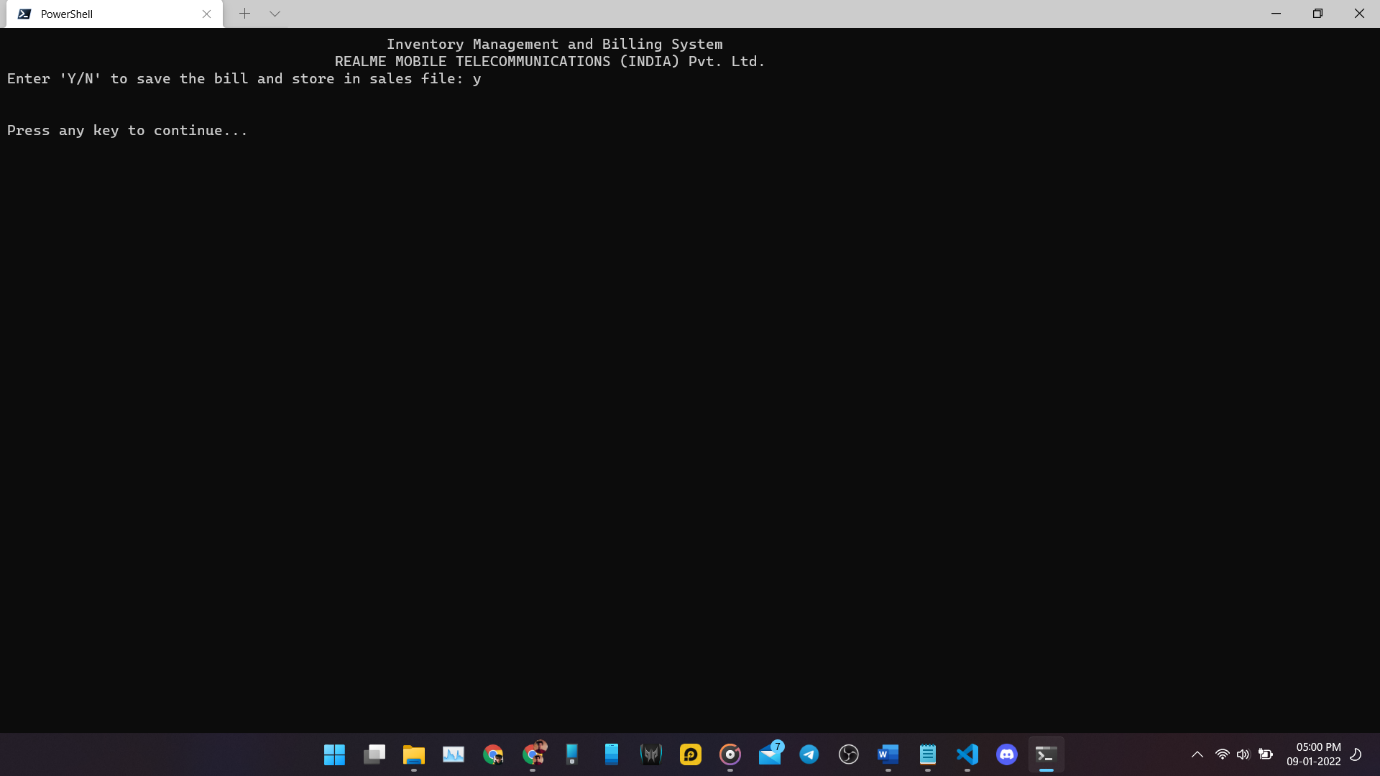


Update product quantity function:

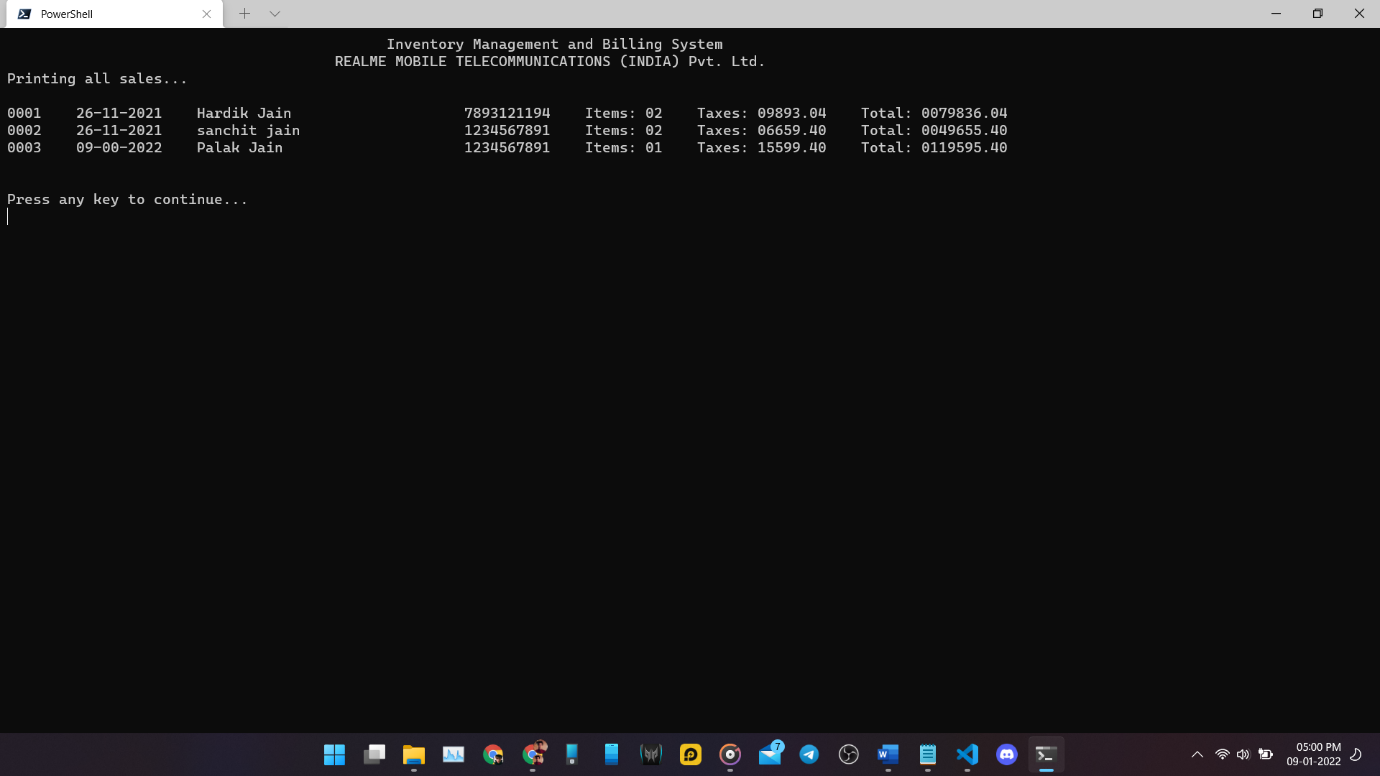


Create a new bill function: 

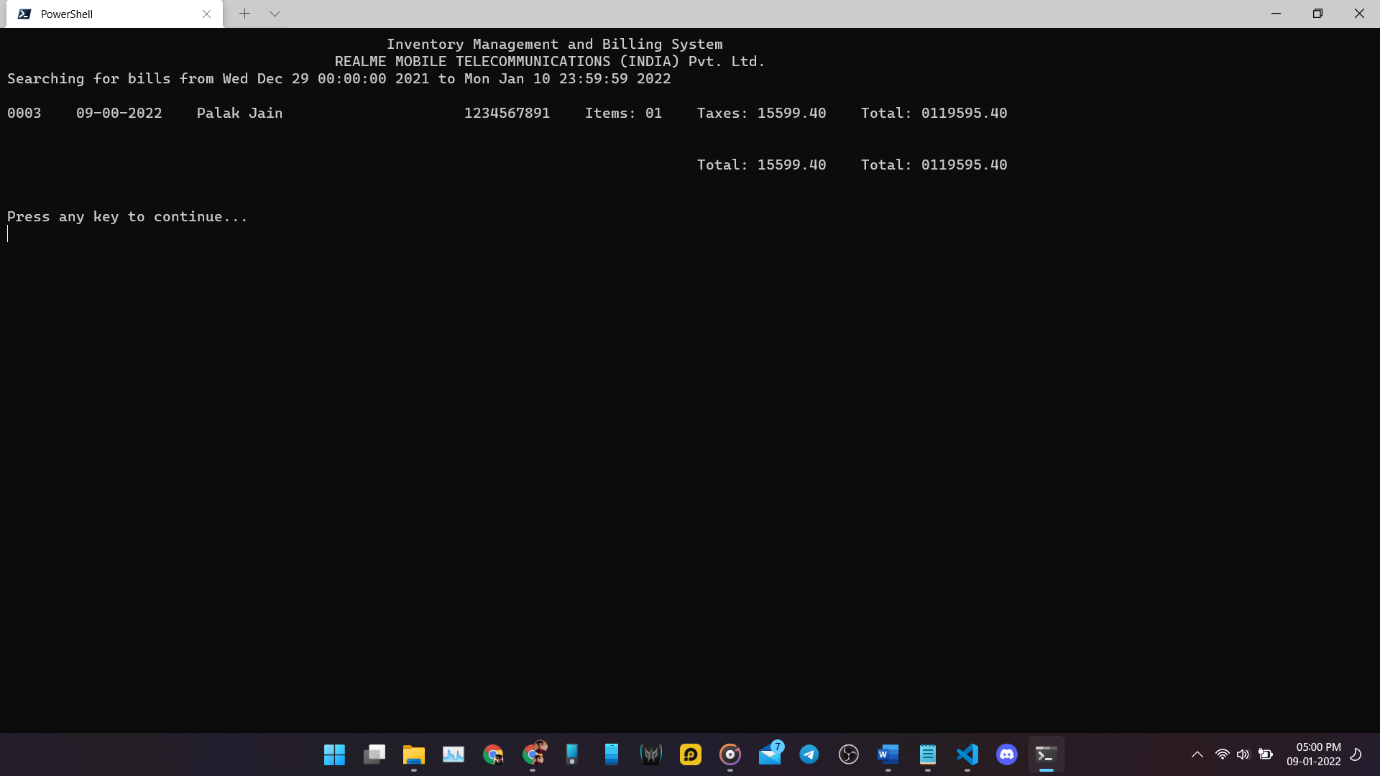




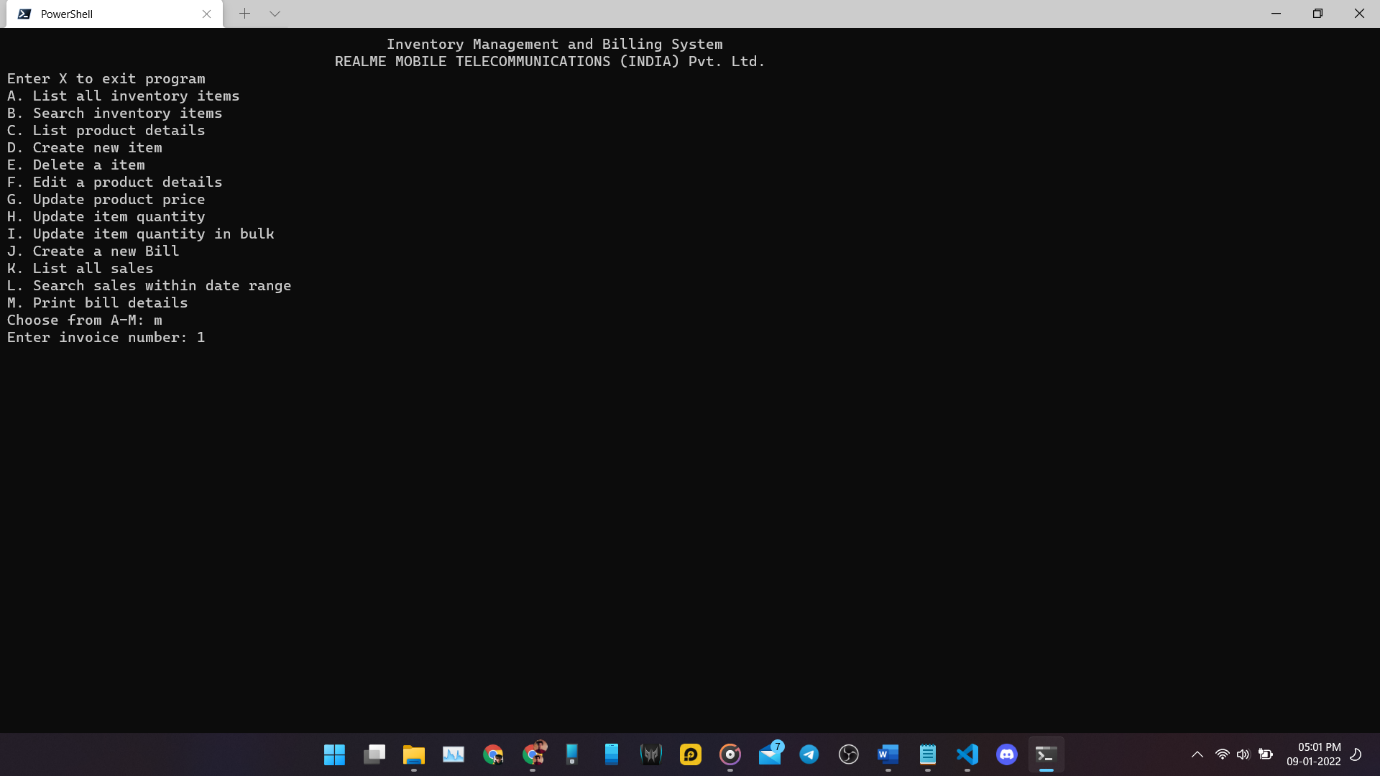
List all sales function:

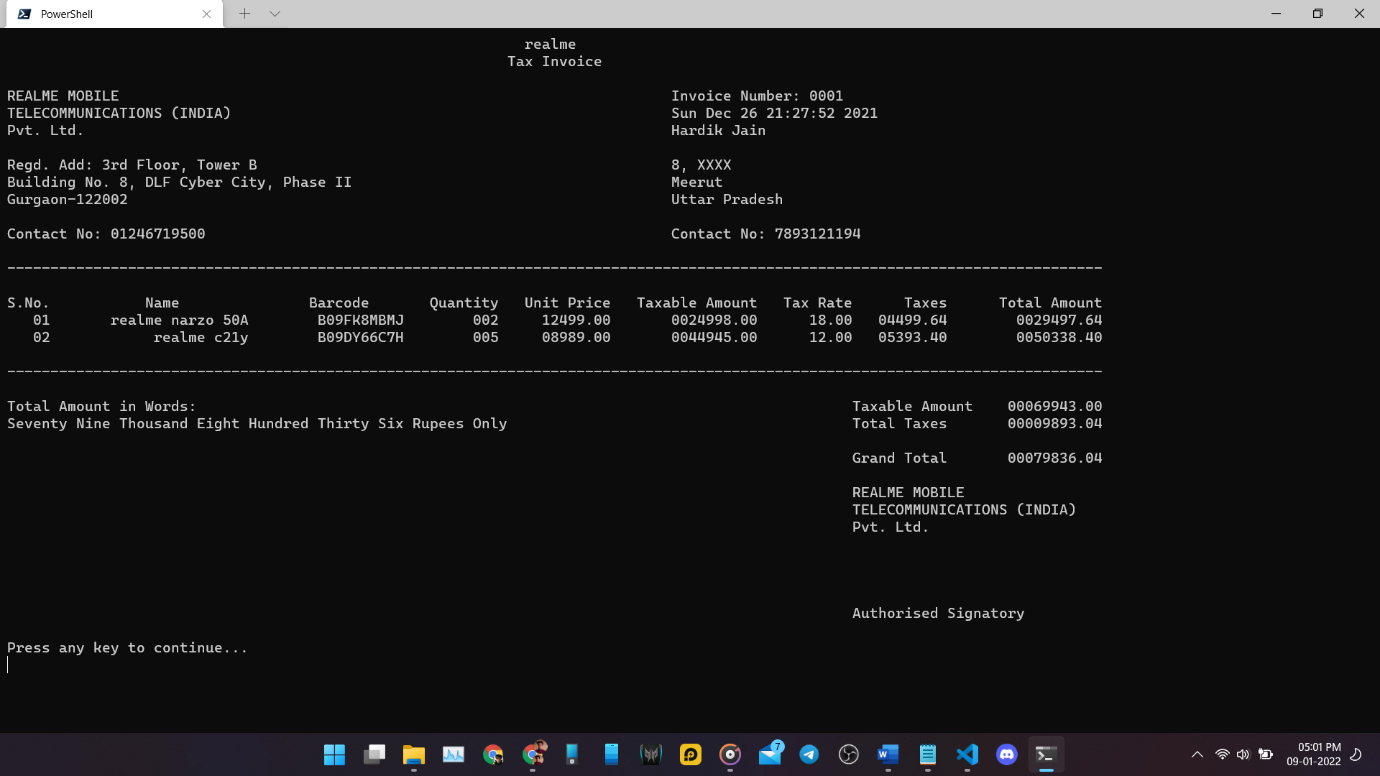


List sales within date range function:



Print previous invoice details function:





# References

* https://www.tutorialspoint.com/cprogramming/c\_constants.htm
* https://www.geeksforgeeks.org/sprintf-in-c/
* https://www.techonthenet.com/c\_language/standard\_library\_functions/stdlib\_h/atof.php
* https://www.tutorialspoint.com/what-are-pointers-to-structures-in-c-language/;
* https://www.epochconverter.com/
* https://zetcode.com/articles/cdatetime/
* https://stackoverflow.com/questions/1442116/how-to-get-the-date-and-time-values-in-a-c-program
* https://www.geeksforgeeks.org/time-h-header-file-in-c-with-examples/
* https://www.geeksforgeeks.org/ctime-function-in-c-c/
* https://www.geeksforgeeks.org/time-function-in-c/
* https://www.geeksforgeeks.org/basics-file-handling-c/
* https://stackoverflow.com/questions/15472299/split-string-into-tokens-and-save-them-in-an-array
* https://stackoverflow.com/questions/19424030/why-gets-function-skips-when-preceded-by-scanfd/
* https://www.geeksforgeeks.org/strcmp-in-c-cpp/
* https://www.geeksforgeeks.org/goto-statement-in-c-cpp/
* https://stackoverflow.com/questions/12740268/how-can-i-restart-while-loop-after-certain-condition-if-satisfied-in-c
* https://www.educative.io/blog/concatenate-string-c
* https://stackoverflow.com/a/32349605
* https://stackoverflow.com/a/153895
* https://www.epochconverter.com/programming/c