INF 212 Algorithms and Programming II

2020-2021 Spring
Electronic Engineering

Project 1# Sale Program

“S.O.L.D”
Sales Automation Software

School ID	Name	Surname
151024064	Burak	Topçu

PROJECT OBJECTIVE

That project's main objective is create sales automation software. This software will allow sellers to easily manage and store product, customer and sales records and also they can analyze that records, according to that analyze results can detect which product most or less preferred by customers. If a product don't preferred by customers, seller can don't purchase that and save it's store from loss. This program simplifies the sales process and provides the seller to optimize his shop for the customer profile.

PROBLEM

Product and customer information should be able to be defined in the program. Sales transaction must be able made according to defined customer and product information and the record of this transaction must be kept in the program. Variables such as amount, quantity and ID of customer, product and sales transaction information should not be mixed with each other. All these information and transactions should be analyzed and the statistics should be shown to the seller. The seller should be able to easily access the information they want.

ANALYSIS

We defined problem at the previous section, now let's look at it in more detail.
1. Product and Customer Information: That infos should able to define in the program with it's parameters like ID, Name, Price, Location etc…
2. Sales Action and Records: Program should able to sale a defined product to a defined customer and calculate shipping cost relative to customer location, also need to log and store that sale records…
3. Analyze of Records: Program should able to analyze product,customer and sale infos, for calculate statistics like most or less purchased product, most seen customer profile etc…

DATA REQUIREMENTS

Structured Data Type

Product information, Customer information and Purchace information is structured data types.
Product Info Structure consist of: ID, name, price, type.
Customer Info Structure consist of: ID, name, type, coordinates.
Purchace Info Structure consist of: ID, invoice_ID, product_ID , cost, customer_ID

Problem Constants

Problem don't have any constant value.

Problem Inputs

1. Product Info Inputs -> ID,Name,Price,Type
2. Customer Info Inputs ->ID,Name,Type, X Coordinate,Y Coordinate

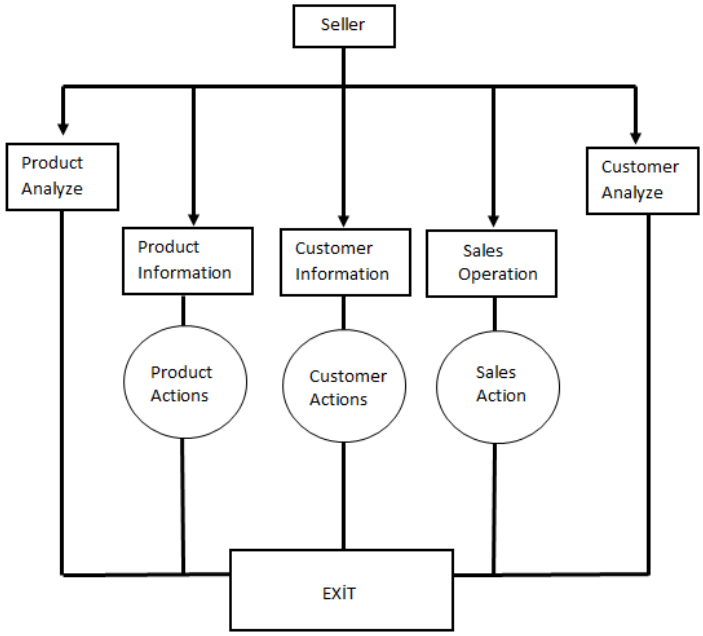
3. Sale Process Inputs:
Quantify: Quantify parameter entering by user during sale process and also can calculate that from Purchace Info/Product Info with Amount/Price function.
product_ID: ID of the product to be sold.
customer_ID: ID of the customer to whom the sale is made.

Problem Output

1. Sale Process Outputs ->ID, invoice_ID,cost
ID: The value assigned to each product sold.
invoice_ID: The value assigned to the invoice record where sales transactions made to the customer are kept.
Cost: Calculation for new sales based on (Quantify*Price) function.
Shipping Cost: Calculating according to customer's distance to store.
Shipping Function: $(X^2+Y^2)^{0.5}*(C)$; C= Shipping price for a unit length.

Customer and Product Categorization:
As a output, user wants to list that infos based on selected specification like Type, ID etc…
Product Analyze Results:
Total purchase quantity and amount of a product
Total purchase amount of a product type
Total purchase amount for all products
Most or less preferred product
Customer Analyze Results:
Products purchased by a customer
Total amount of purchased by a customer
Total amount of products purchased by all customer
Customers shipping fee

DESIGN
ALGORITHM



HEADER FILE(S)

"product.h"
"purchase.h"
"customer.h"

IMPLEMENTATION

I used structure data type with linked list feature to solve problem and provide required specificationsto my program. Problem input parameters and output variables defined in that structures. Structure and parameter/variable names and interface/menu names are meaning full so anyone can easily understand this. Linked list feature provides ease of access to structure for edit/view/insert/delete actions for user. Key point at that structure type is each node contains memory address of next node. I created a main menu and sub menus so that the user can easily switch between sections. In menu creating were used Switch-Case statements method.

My project steps here:

1. Creating data structures and it's elements
2. Creating Functions for add/delete/set actions to structured datas
3. Creating functions for print/list actions
4. Creating main function and menus
5. Importing Test Data
6. Testing and Corrections
7. Finalization

TESTING

I imported product, customer and purchase (sale action) data to my program in main function for testing. In the analysis/print sections, I did the calculations first in the MS Excel program and then in my own program. Then I made sure that I got the right results by comparing. When I got wrong results, I made corrections in the program code and I added if/else statements for unexpected cases.

USER'S GUIDE

User can add custom product, customer and purchace (sale action) data.
Input Parameters: Product(ID, Name,Price,Type)
Customer(ID,Name,Type,X Coordinate,Y Coordinate)
Purchase(~~ID~~, ~~invoice ID~~, customer ID,product_ID, ~~Cost~~)

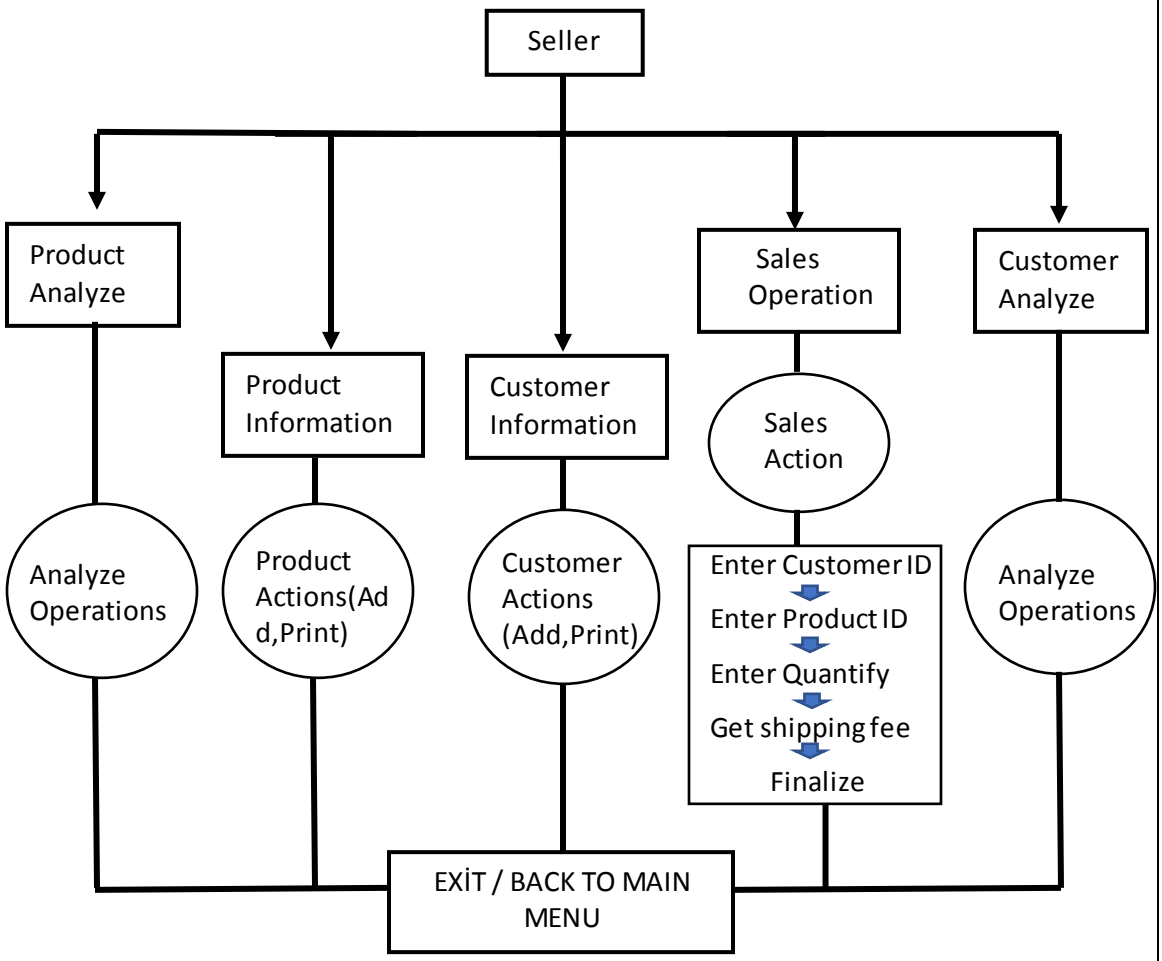
Quantify: Quantify parameter entering by user during sale process.
product_ID: ID of the product to be sold.
customer_ID: ID of the customer to whom the sale is made.
Outputs:
Purchase(ID, invoice ID, ~~customer ID,~~ ,~~product_ID~~ ,Cost)
ID: The value assigned to each product sold.

invoice_ID: The value assigned to the invoice record where sales transactions made to the customer are kept.
Cost: Calculation for new sales based on (Quantify*Price) function.

User can get any customer,product,purchase and invoice data at the program memory. And can analyze that data with statictical calculations like that:
Shipping Cost of Customer, Total Amount of Product and its Quantify, Total Purchase Amount of Customer,Most Sold Product,Less Sold Product etc…

The user selects the desired section in the menus and performs the action he wants to do. It can list all products or a single product on the product menu. It can list all customers or a single customer from the customer menu. It can also list according to the product and customer type.
User can see the statistics I mentioned above in analysis sections.
While performing the sales process, it selects the sale from the menu and performs the sales transaction by entering the customer ID to be sold and the quantify and ID of the products to be sold. Finally, the shipping fee is calculated and the sales process is finalized.

GENERAL FLOWCHART OF THE PROJECT



Design Note: Purchase ID, invoice_ID,product ID, customer_ID assignments done automatically by the program.
For example if last invoice_ID is 3 program will be assign last_ID+1 as a new invoice_ID which is equals to 4.

CONCLUSION AND REMARKS

My program's all functions working properly, I tested with imported test data.
I used switch-case statement for create menu and that's provides user friendly design to program. To do some analysis/print operations, it was necessary to get data from two or three different structures and match it's elements with each other, for this reason I used "for" loop and "if else" statement at many times. I can optimize and simplify my code and improve its performance by writing alternative functions into the header files instead of using loop and statement expressions at so many times.
I learned switch-case statement and linked list feature during development process of project.

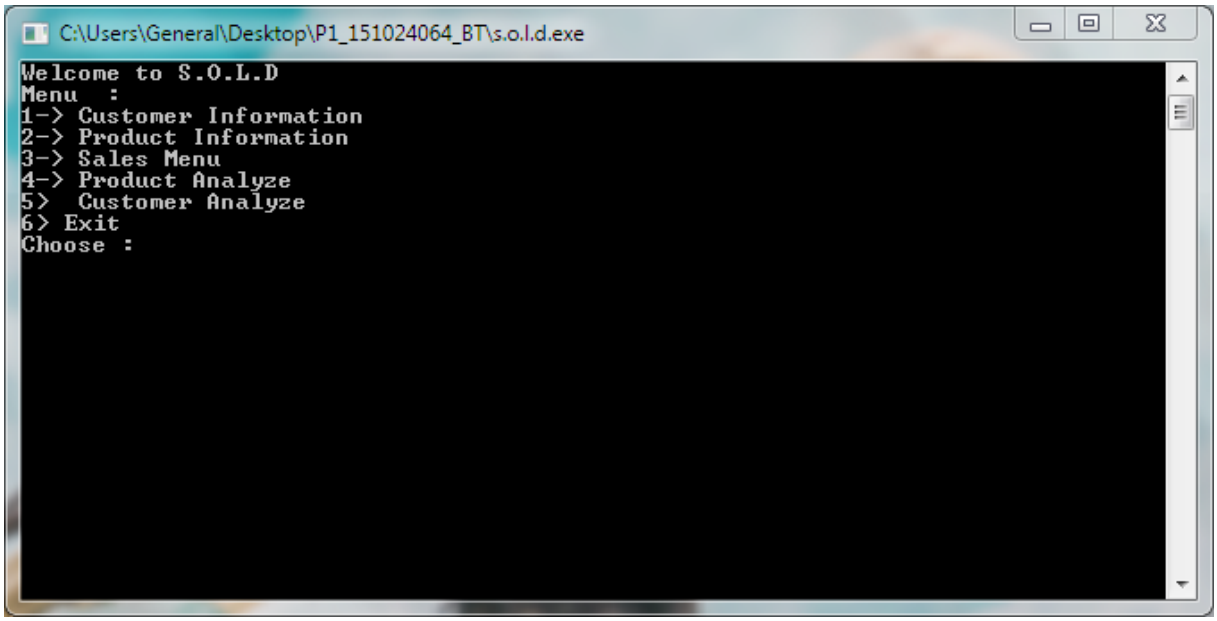
REFERENCES

I did research before start the project for how create menu and how use linked list data structure.
https://www.tutorialspoint.com/data_structures_algorithms/linked_list_program_in_c.htm
https://www.geeksforgeeks.org/menu-driven-program-using-switch-case-c/
And I examined some sales managemenent projects on GitHub for determine main concept of this softwares…
https://github.com/DadiAnas/SalesManagement
https://github.com/rc-bandit4461/shop_management_c
https://github.com/maheshwarkuchana/Car-Sales-Management-System
https://github.com/Vivek2k20/Car-Sales-Management-System
https://github.com/iamAlbano/TAD-Estoque

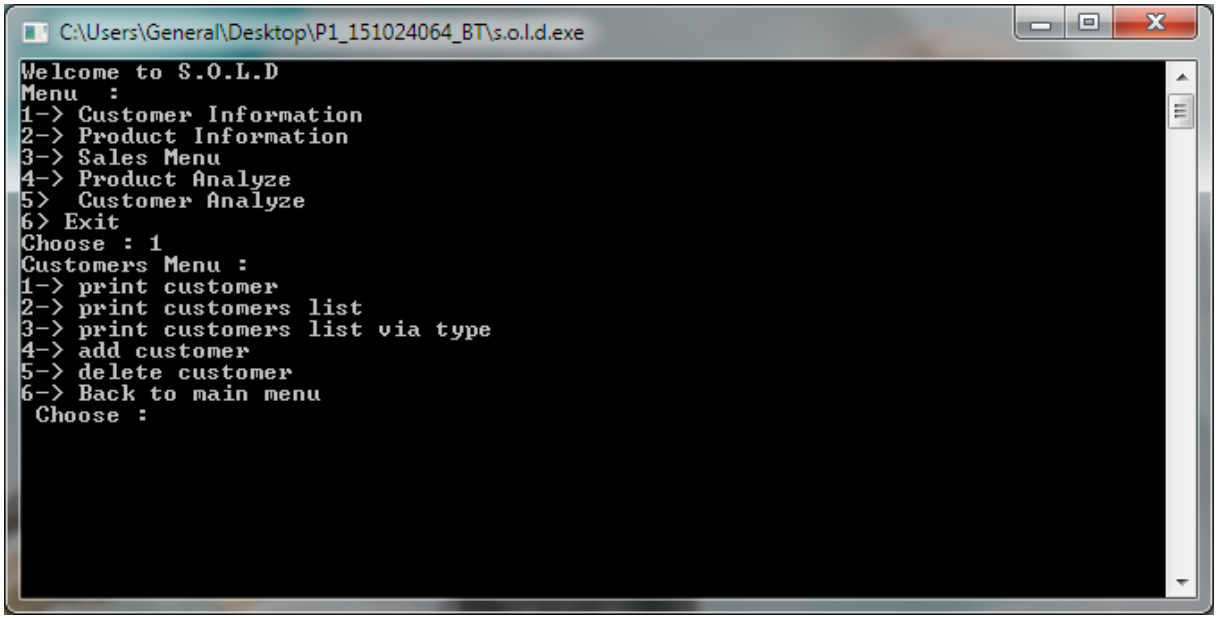
,

Screeenshots

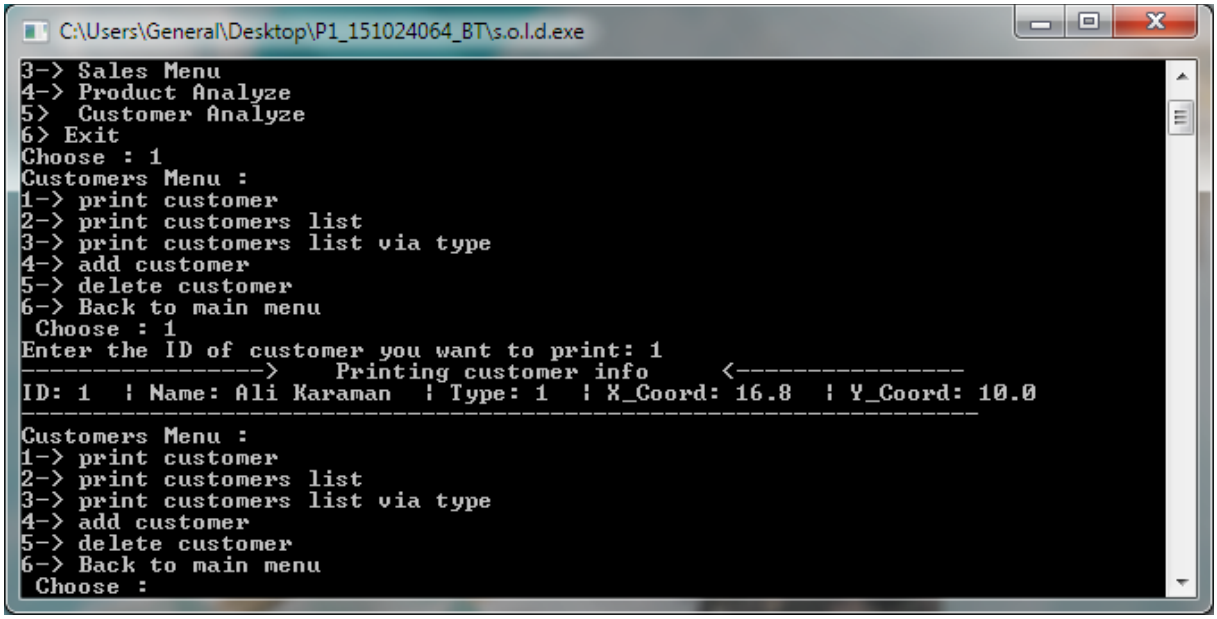
Main Menu



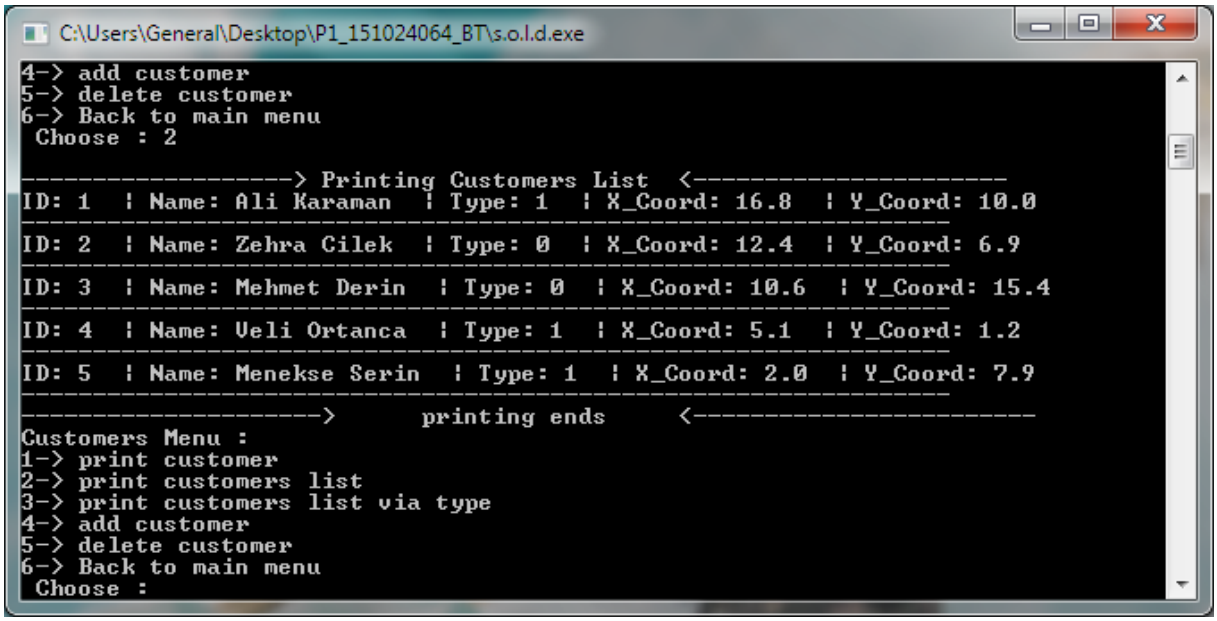
Customer Menu



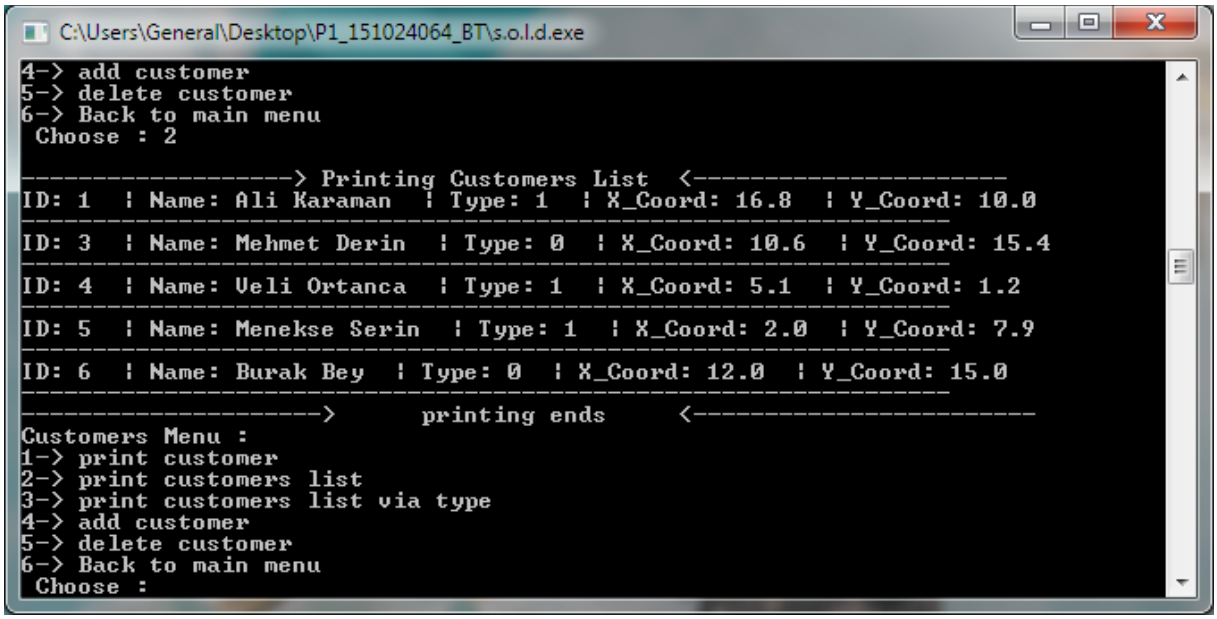
Customer Print Example



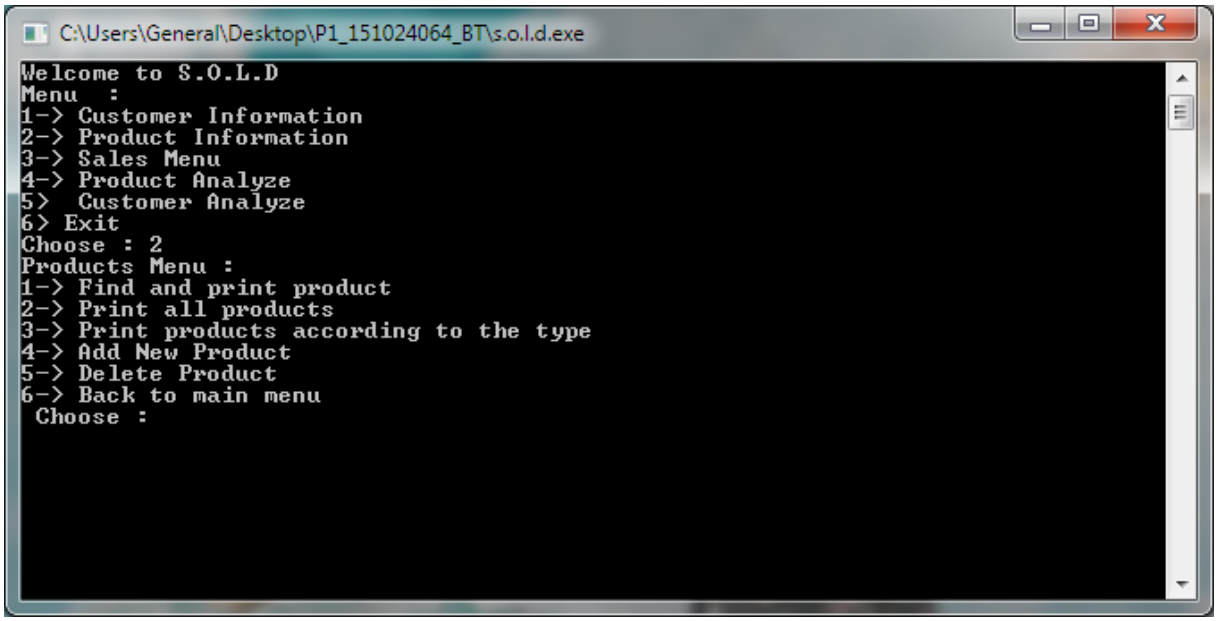
Customer List Print Example



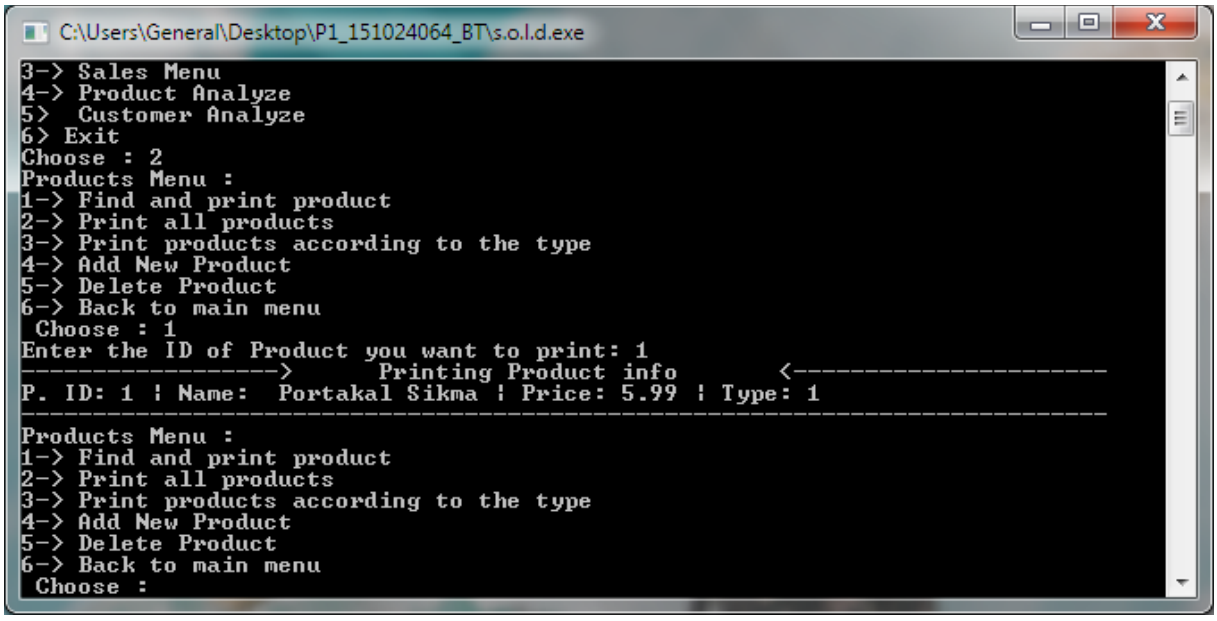
Customer Add/Delete Example (ID 2 deleted ID 6 added)



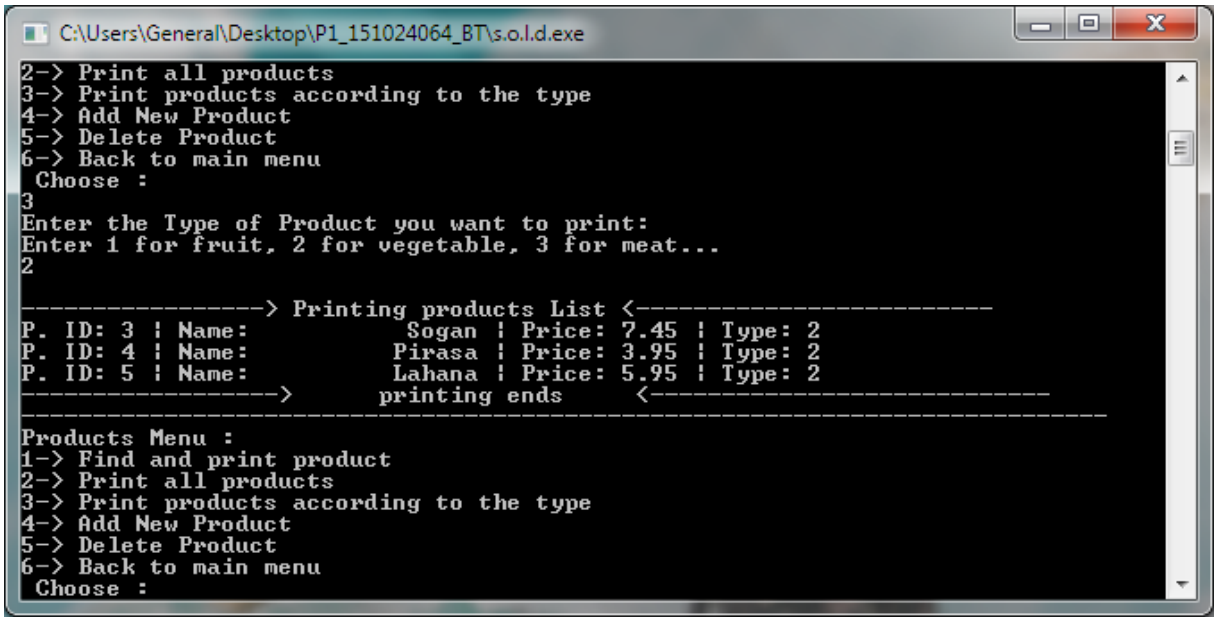
Products Menu



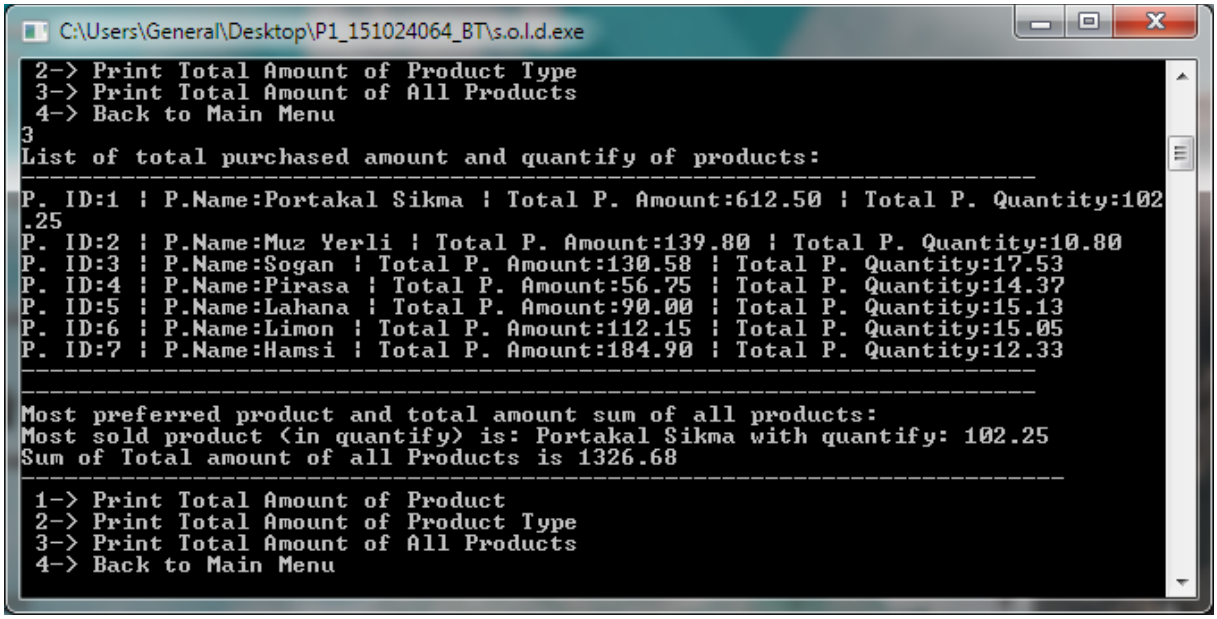
Example Product Print



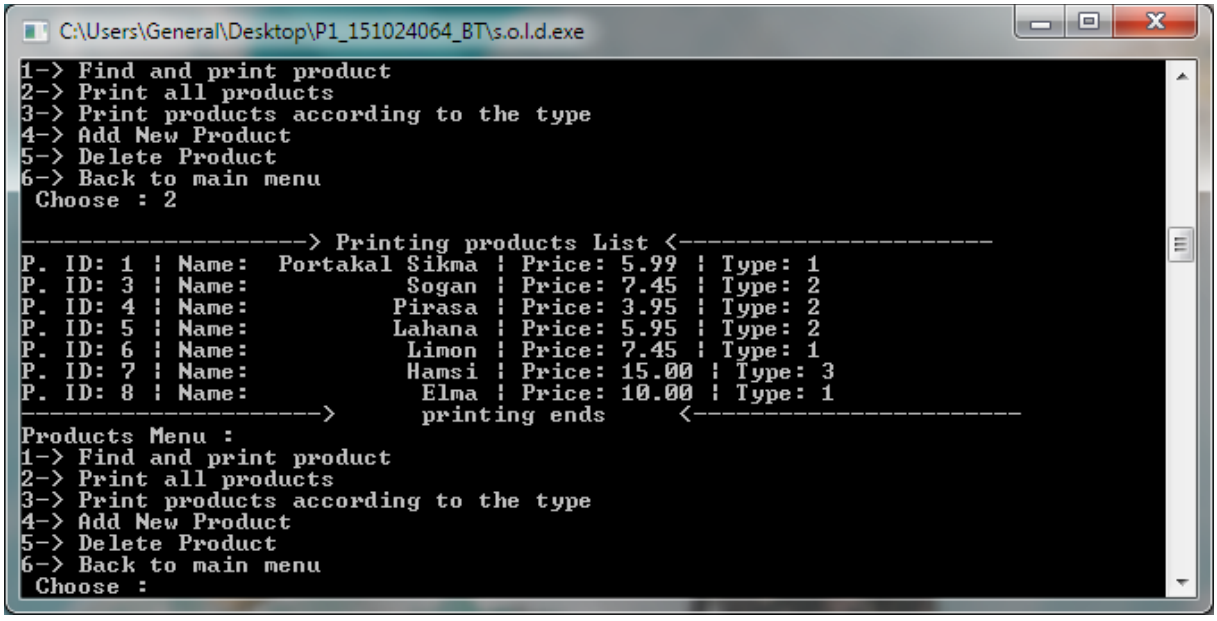
Example Product Print via Type



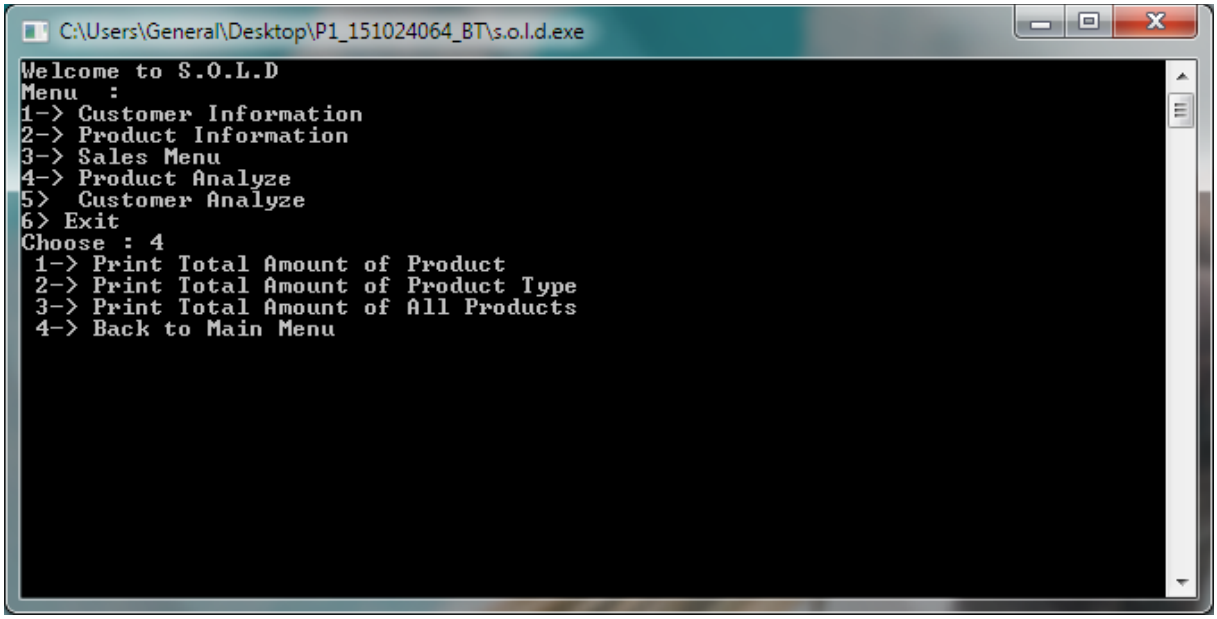
Example Product List Print



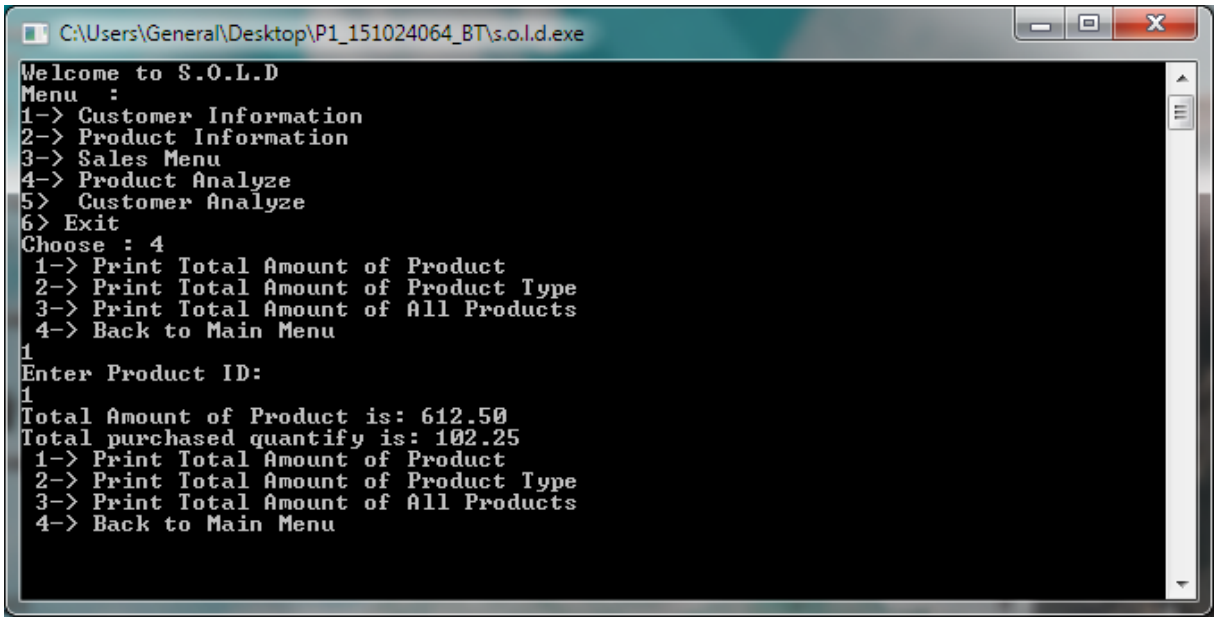
Example Product Add/Delete (ID 2 removed ID 8 added)



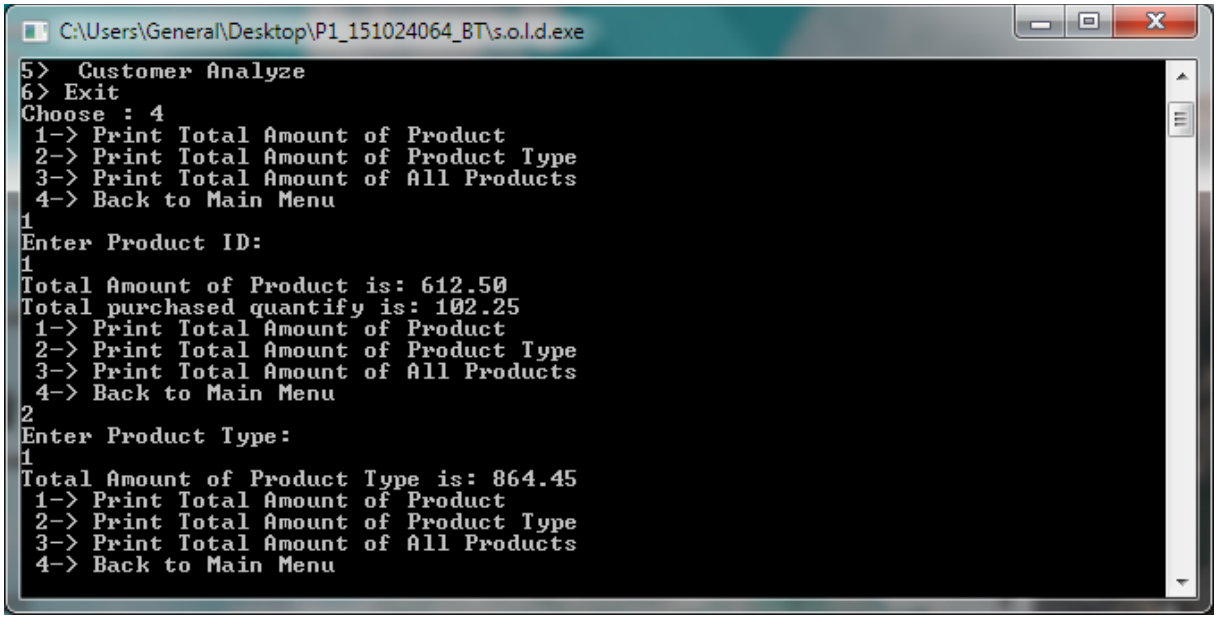
Product Analyze Menu



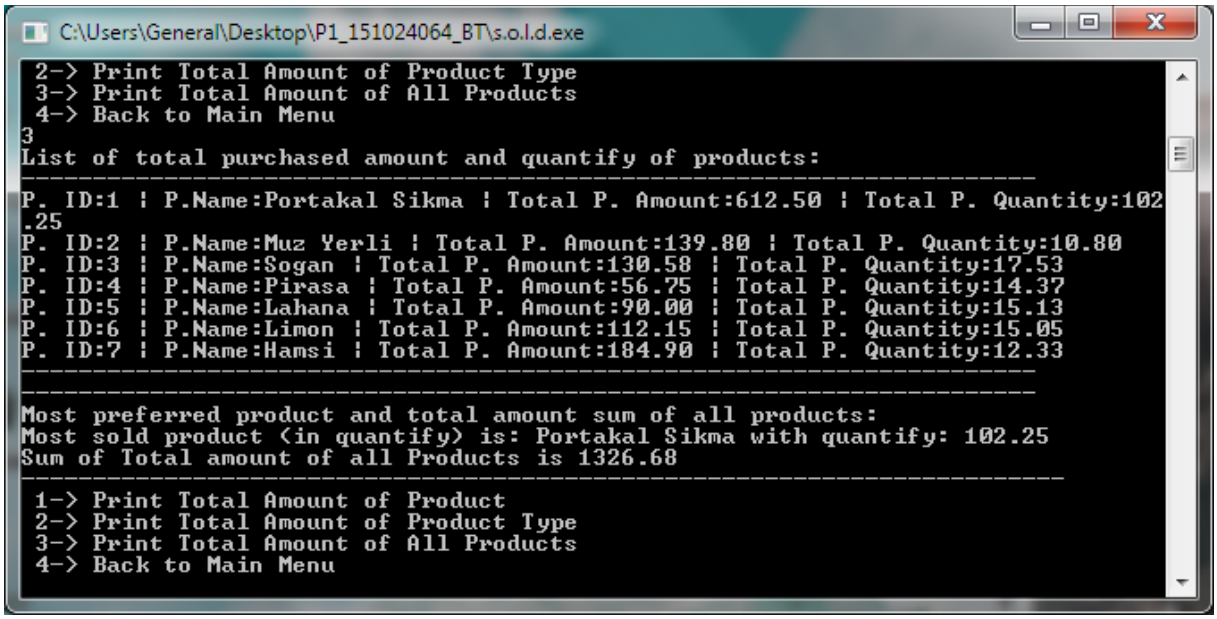
Example Total Amount of Product



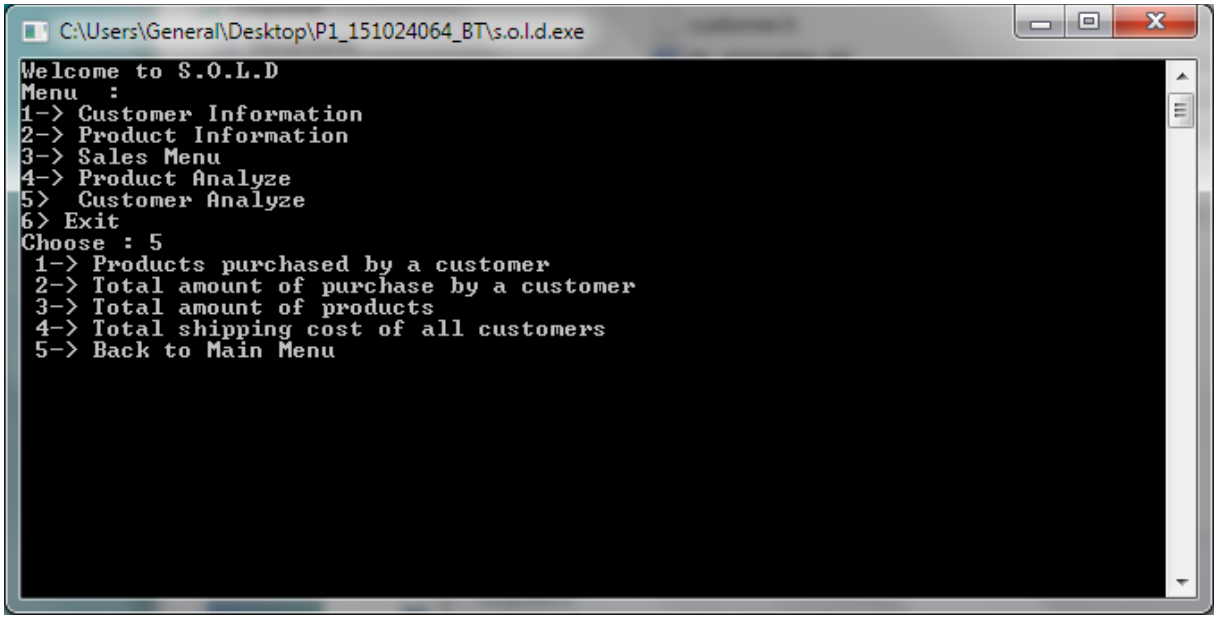
Example Total Amount of Product Type



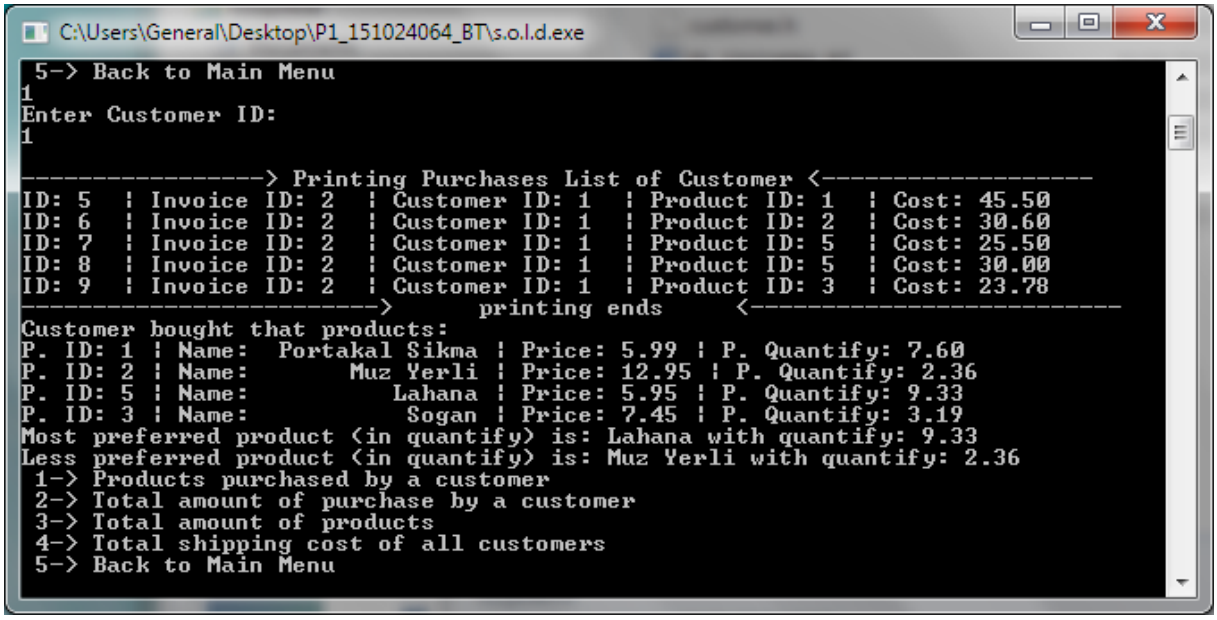
Example Total Amount of All Products



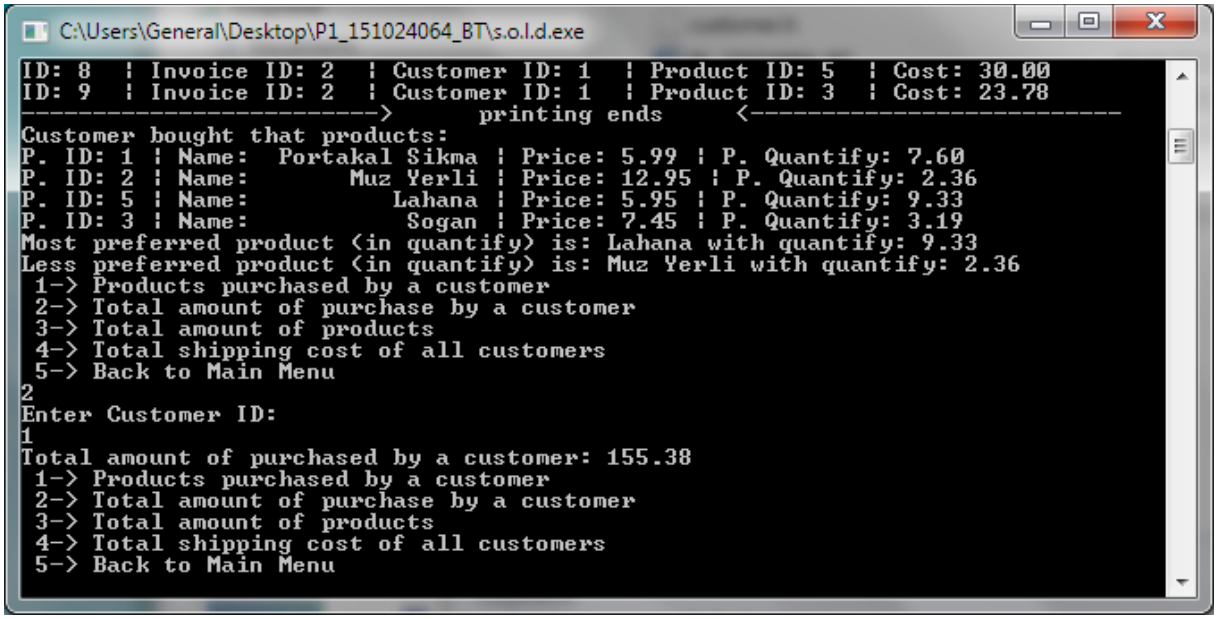
Customer Analyze Menu



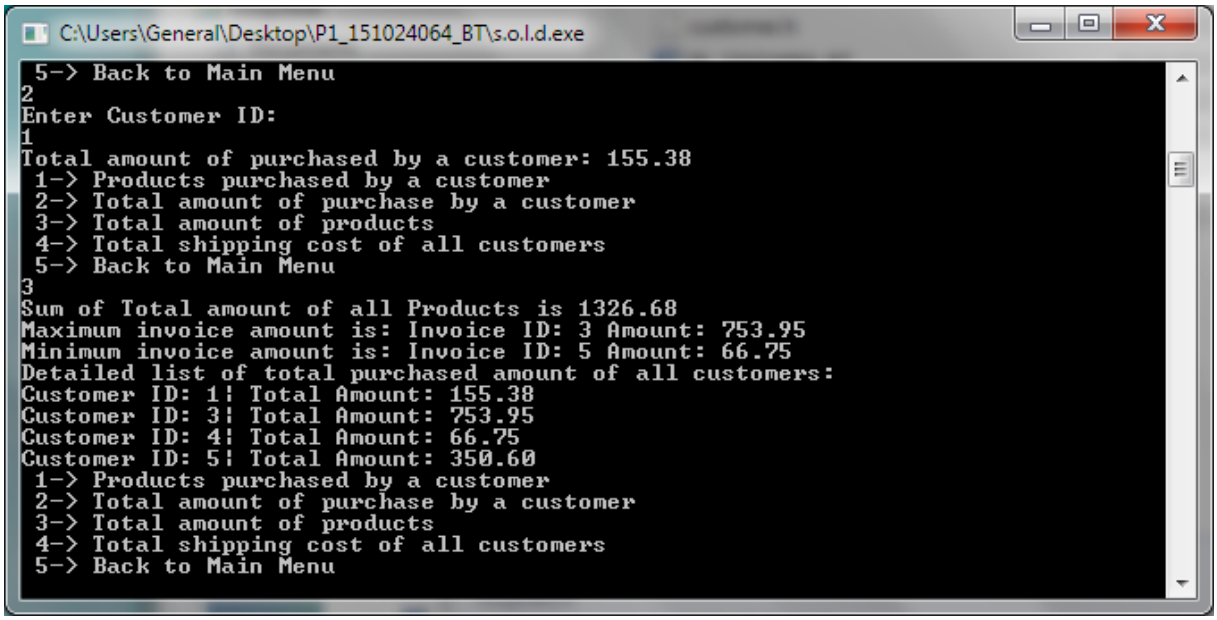
Customer Purchase Listing and Analyze



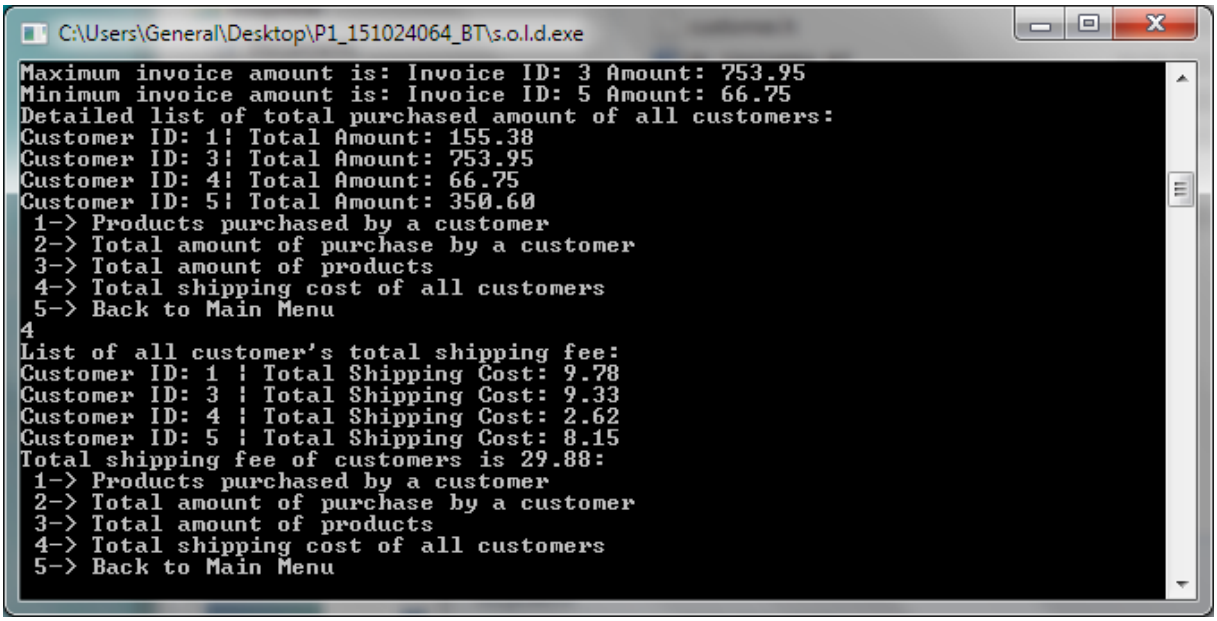
Total Amount of Customer Calculation



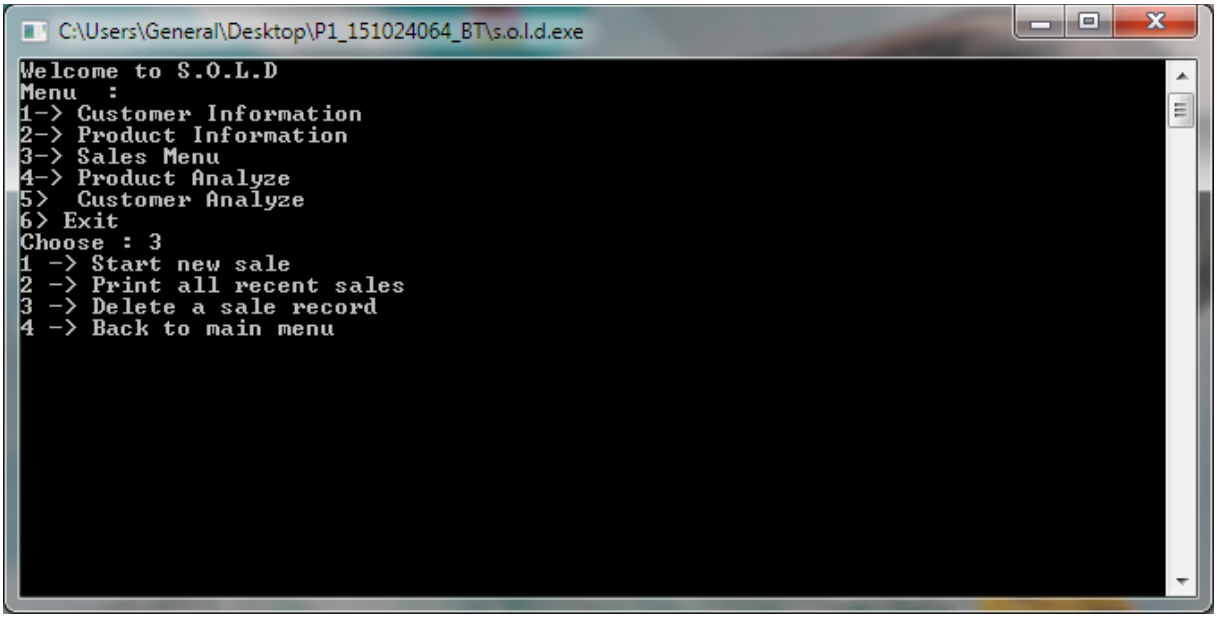
Total Amount All of Customers



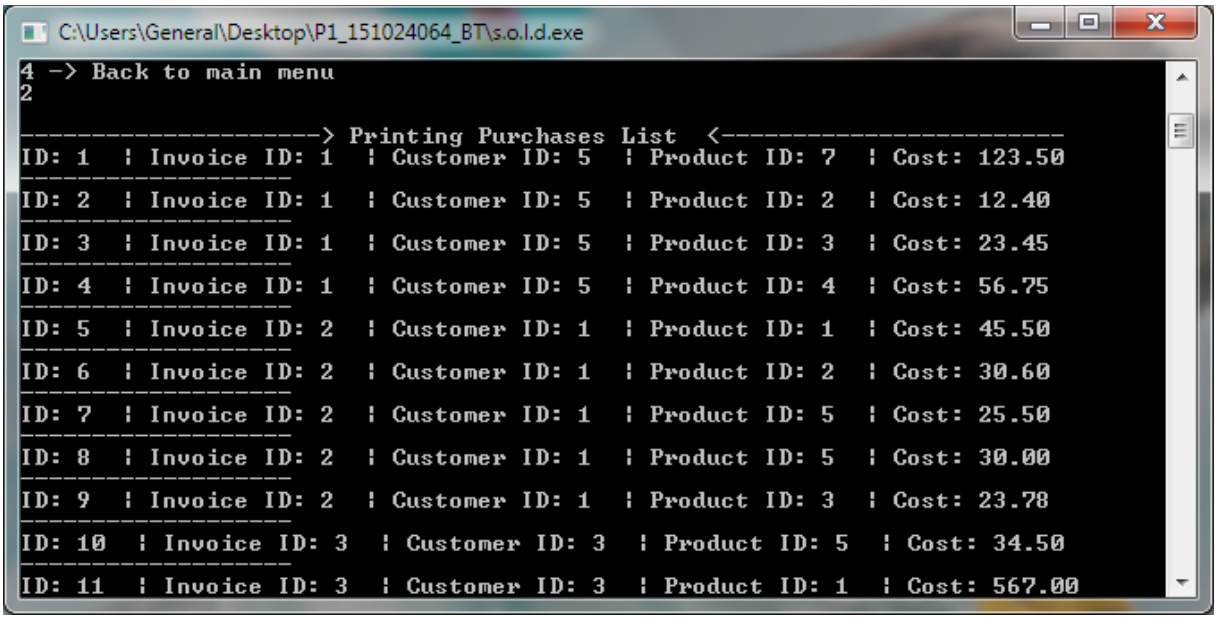
Shipping Free Calculation



Sale Menu



Printing of Purchase Logs



New Sale (Product and Customer Selection)

Quantify entered by user and finalized sale action shipping free and total cost calculated.