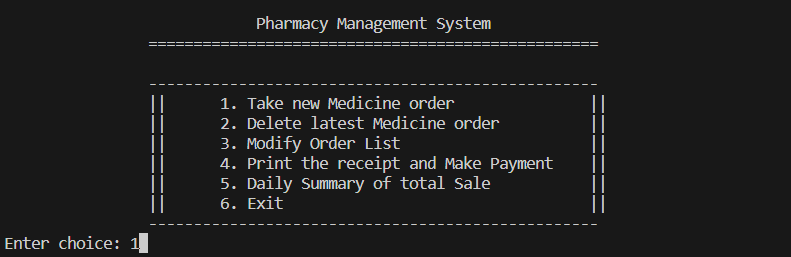
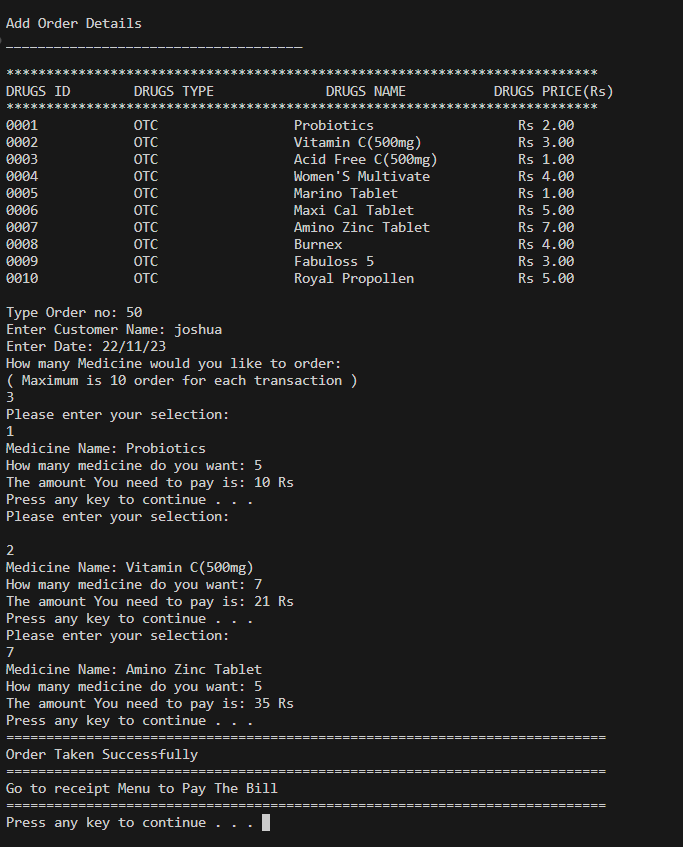
|  |
| --- |
| Program: |
| // Pharmacy Management System  #include <iostream>  #include <stdlib.h>  #include <string>  #include <cctype>  #include <cmath>  #include <cstdio>  #include <fstream>  #include <iomanip>  #define max 10  using namespace std;  class medicineType //base class  {  public:  void take\_order(); //to take\_order  void delete\_order(); //to delete the order  void modify(); //to modify the order  void order\_list(); //to display the order\_list  void daily\_summary(); //to display daily\_summary  void exit(); //function to exit system  medicineType(); //constuctor  };  medicineType::medicineType ()  {  } //constructor for class CarType  struct node  {  int receipt\_number;  string customerName;  string date;  int quantity[10];  string type = {"OTC"};  int x, menu2[10];  double amount[10];  string medicineName[10]={"Probiotics","Vitamin C(500mg)","Acid Free C(500mg)","Women'S Multivate","Marino Tablet","Maxi Cal Tablet",  "Amino Zinc Tablet","Burnex","Fabuloss 5","Royal Propollen"};  double Medicine[10] = {2.00,3.00,1.00,4.00,1.00,5.00,7.00,4.00,3.00,5.00};  double total;  node \*prev;  node \*next;  node \*link;  }  \*q,\*temp; //pointer declaration  node \*start\_ptr = NULL;  node \*head = NULL;  node \*last = NULL;  int main() // Main function  {  medicineType medicine;  int menu;  do  {  system("cls");  cout<<"\t\t\t Pharmacy Management System \n";  cout<<"\t\t==================================================\n\n";  cout<<"\t\t--------------------------------------------------\n";  cout<<"\t\t||\t1. Take new Medicine order \t\t ||\n";  cout<<"\t\t||\t2. Delete latest Medicine order\t\t ||\n";  cout<<"\t\t||\t3. Modify Order List \t\t\t ||\n";  cout<<"\t\t||\t4. Print the receipt and Make Payment \t ||\n";  cout<<"\t\t||\t5. Daily Summary of total Sale \t\t ||\n";  cout<<"\t\t||\t6. Exit\t\t\t\t\t ||\n";  cout<<"\t\t--------------------------------------------------\n";  cout<<"Enter choice: ";  cin>>menu;  switch (menu)  {  case 1:  {  medicine.take\_order();  break;  } //end case 1      case 2:  {  medicine.delete\_order();  system("PAUSE");  break;  } //end case 2    case 3:  {  medicine.modify();  system("PAUSE");  break;  } //end case 3  case 4:  {  medicine.order\_list();  system("PAUSE");  break;  } //end case 4  case 5:  {  medicine.daily\_summary();  system("PAUSE");  break;  } //end case 5  case 6:  {  medicine.exit();  goto a;  break;  } //end case 6    default:  {  cout<<"You entered invalid input\nRe-enter the input\n\n";  break;  }  }  }while(menu!=6);  a://goto  cout<<"Thank you"<<endl;  system ("pause");  return 0;  }  void medicineType::take\_order() //function to take order  {  system("cls");  int i;  int choice, quantity, price,None;  cout <<"\nAdd Order Details\n";  cout <<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n\n";  node \*temp;  temp=new node;  cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";  cout<<"DRUGS ID"<<"\tDRUGS TYPE"<<" \t\tDRUGS NAME"<<" DRUGS PRICE(Rs)\n";  cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";  cout<<"0001"<<"\t"<<"\tOTC"<<"\t\t"<<" Probiotics"<<" Rs 2.00"<<endl;  cout<<"0002"<<"\t"<<"\tOTC"<<"\t\t"<<" Vitamin C(500mg)"<<" Rs 3.00"<<endl;  cout<<"0003"<<"\t"<<"\tOTC"<<"\t\t"<<" Acid Free C(500mg)"<<" Rs 1.00"<<endl;  cout<<"0004"<<"\t"<<"\tOTC"<<"\t\t"<<" Women'S Multivate"<<" Rs 4.00"<<endl;  cout<<"0005"<<"\t"<<"\tOTC"<<"\t\t"<<" Marino Tablet"<<" Rs 1.00"<<endl;  cout<<"0006"<<"\t"<<"\tOTC"<<"\t\t"<<" Maxi Cal Tablet"<<" Rs 5.00"<<endl;  cout<<"0007"<<"\t"<<"\tOTC"<<"\t\t"<<" Amino Zinc Tablet"<<" Rs 7.00"<<endl;  cout<<"0008"<<"\t"<<"\tOTC"<<"\t\t"<<" Burnex"<<" Rs 4.00"<<endl;  cout<<"0009"<<"\t"<<"\tOTC"<<"\t\t"<<" Fabuloss 5"<<" Rs 3.00"<<endl;  cout<<"0010"<<"\t"<<"\tOTC"<<"\t\t"<<" Royal Propollen"<<" Rs 5.00"<<endl;  cout<<" "<<endl;    temp = new node;  cout<<"Type Order no: ";  cin>> temp->receipt\_number;  cout<<"Enter Customer Name: ";  cin>> temp->customerName;  cout<<"Enter Date: ";  cin>>temp->date;  cout<<"How many Medicine would you like to order:\n( Maximum is 10 order for each transaction ) \n";  cin >> temp->x;  if (temp->x >10)  {  cout << "The Medicine you order is exceed the maximum amount of order !";  //system("pause");  }  else  {  for (i=0; i<temp->x; i++)  {    cout<<"Please enter your selection:\n";  cin>>temp->menu2[i];  cout<<"Medicine Name: "<<temp->medicineName[temp->menu2[i]-1]<<endl;  cout << "How many medicine do you want: ";  cin >> temp->quantity[i];  temp->amount[i] = temp->quantity[i] \* temp->Medicine[temp->menu2[i]-1];  cout << "The amount You need to pay is: " << temp->amount[i]<<" Rs"<<endl;  system("PAUSE");    }  cout<<"===========================================================================\n";  cout << "Order Taken Successfully\n";  cout<<"===========================================================================\n";  cout << "Go to receipt Menu to Pay The Bill\n";  cout<<"===========================================================================\n";  system ("PAUSE");  temp->next=NULL;  if(start\_ptr!=NULL)  {  temp->next=start\_ptr;  }  start\_ptr=temp;  system("cls");  }  }//End function take\_order  void medicineType::order\_list() //Function to display receipt  {  int i, num, num2;  bool found; //variable to search  system("cls");  node \*temp;  temp=start\_ptr;  found = false;    cout<<" Enter the Receipt Number To Print The Receipt\n";  cin>>num2;  cout<<"\n";  cout<<"===========================================================================\n";  cout <<"\t\tHere is the Order list\n";  cout<<"===========================================================================\n";  if(temp == NULL) //Invalid receipt code  {  cout << "\tThere is no Order to show\n\t\t\tSo The List is Empty\n\n\n";  }  while(temp !=NULL && !found)  {  if (temp->receipt\_number==num2)  {  found = true;  }  else  {  temp = temp -> next;  }  if (found) //print the receipt  {  cout <<"receipt Number : "<<temp->receipt\_number;  cout <<"\n";  cout<<"Customer Name: "<<temp->customerName<<endl;    cout<<"Order Date : "<<temp->date<<endl;    cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;    cout << "===============================================================================\n";  cout << "| Medicine Type | Medicine Name | Quantity | Total Price |\n";  cout << "===============================================================================\n";  for (i=0;i<temp->x;i++)  {  cout << temp->type <<"\t\t";  cout<<temp->medicineName[temp->menu2[i]-1]<<"\t\t\t";  cout<<temp->quantity[i] <<"\t\t";  cout<< temp->amount[i]<<" Rs"<<endl;  cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  }    temp->total = temp->amount[0]+temp->amount[1]+temp->amount[2]+temp->amount[3]+temp->amount[4]+temp->amount[5]+temp->amount[6]+temp->amount[7]  +temp->amount[8]+temp->amount[9];  cout<<"Total Bill is : "<<temp->total;  cout<<"\n";  cout<<"Type the exact amount You need to pay:";  cin>>num;  cout <<"\n\n";  cout<<"Payment Done\nThank You\n";  cout <<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  }  }  } //End function order\_list  void medicineType::delete\_order() //function to delete\_order  {  system("cls");  int i, num, count;  cout<<"Enter the order you want to delete \n";  cin>>num;  node \*q;  node \*temp;  bool found;  if(start\_ptr == NULL)  cerr<<"Can not delete from an empty list.\n";  else  {  if(start\_ptr->receipt\_number == num)  {  q = start\_ptr;  start\_ptr = start\_ptr->next;  count--;  if(start\_ptr == NULL)  last = NULL;  delete q;  cout<<"The receipt is deleted successfully"<<endl;  }  else  {  found = false;  temp = start\_ptr;  q = start\_ptr->next;    while((!found) && (q != NULL))  {  if(q->receipt\_number != num)  {  temp = q;  q = q-> next;  }  else  found = true;  }  if(found)  {  temp->next = q->next;  count--;  if(last == q)  last = temp;  delete q;  cout<<"The receipt is Deleted Successfully"<<endl;  }  else  cout<<"Item to be deleted is not in the list."<<endl;  }  }  } //End function delete\_order  void medicineType::modify() //function to modify order  {  system("cls");  int i, ch, sid;  bool found;  found = false;  temp = start\_ptr;  cout<<"Enter Receipt Number To Modify: ";  cin>>sid;  if (temp==NULL && sid==0)  {  cout<<"NO RECORD TO MODIFY!"<<endl;  }  else  {  while(temp !=NULL && !found)  {  if (temp->receipt\_number==sid)  {  found = true;  }  else  {  temp = temp -> next;  }  if (found)  {  cout<<"Change Order Number: ";  cin>>temp->receipt\_number;  cout<<"Change Customer Name: ";  cin>>temp->customerName;  cout<<"Change Date : ";  cin>>temp->date;  cout<<"How many New Medicine would you like to Change:\n( Maximum is 10 order for each transaction ) \n";  cout<< " " ;  cin>>temp->x;  if (temp->x >10)  {  cout<<"The Medicine you order is exceed the maximum amount of order !";  system("pause");  }  else{  for (i=0; i<temp->x; i++)  {    cout << "Please enter your selection to Change: "<<endl;  cin>> temp->menu2[i];  cout<< "Change Medicine Name: " <<temp->medicineName[temp->menu2[i]-1]<<endl;  cout << "How many New medicine do you want: ";  cin >> temp->quantity[i];  temp->amount[i] = temp->quantity[i] \* temp->Medicine[temp->menu2[i]-1];  cout << "The amount You need to pay After Modify is: " << temp->amount[i]<<" Rs"<<endl;  system("PAUSE");  }  temp = temp->next;  system("cls");    }  cout<<"RECORD MODIFIED!"<<endl;  }  else  {  if(temp != NULL && temp->receipt\_number != sid)  {  cout<<"Invalid receipt number!"<<endl;  }  }  }  }  } //End modify function  void medicineType::daily\_summary() //Function to display Daily Summary  {  int i,num;  system("cls");  node \*temp ;  temp=start\_ptr;  if(temp == NULL) //Invalid receipt code  {  cout << "\t\t\tThere is no Order to show\n\t\t\tSo The List is Empty\n\n\n";  }  else  {  cout<<"\n";  cout<<"==========================================================================="<<endl;  cout <<" \t\tHere is the Daily Summary of All Orders \n"; //print all receipt  cout<<"==========================================================================="<<endl;  while(temp!=NULL)  {    cout <<"Receipt Number : "<<temp->receipt\_number;  cout <<"\n";  cout<<"Customer Name: "<<temp->customerName<<endl;    cout<<"Order Date : "<<temp->date<<endl;    cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;    cout << "==========================================================================" << endl;  cout << "| Medicine Type | Medicine Name | Quantity | Total Price |" << endl;  cout << "==========================================================================" << endl;  for (i=0;i<temp->x;i++)  {  cout << temp->type <<" \t\t";  cout<<temp->medicineName[temp->menu2[i]-1]<<"\t\t";  cout<<temp->quantity[i] <<"\t\t";  cout<< temp->amount[i]<<" Rs"<<endl;  cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  }    temp->total = temp->amount[0]+temp->amount[1]+temp->amount[2]+temp->amount[3]+temp->amount[4]+temp->amount[5]+temp->amount[6]+temp->amount[7]  +temp->amount[8]+temp->amount[9];  cout<<"Total Bill is : "<<temp->total;    cout <<"\n";  cout <<"\n";  cout <<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";    temp=temp->next;  }  }  } //End daily summary  void medicineType::exit() //Function to exit  {  cout<<"\nYou choose to exit.\n";  } //end function exit  ////////////////////////////THE END OF PROGRAM////////////////////////////////////////// |

Output:

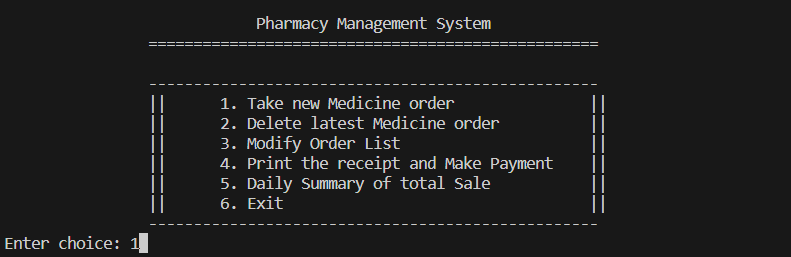
1. Menu(choice:1):



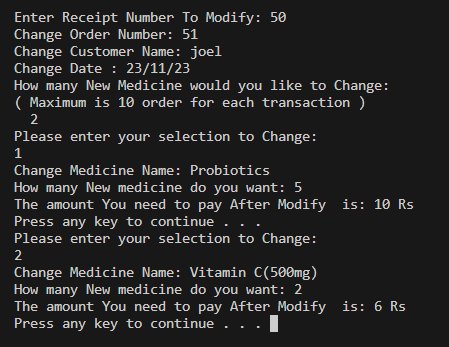
Taking an order:



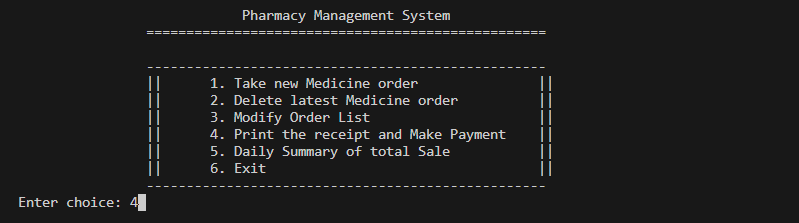
1. Menu(choice:3):



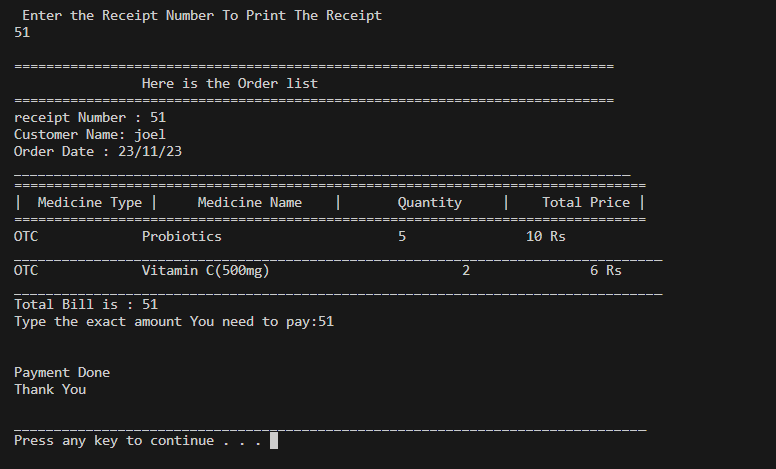
Modifying an order:



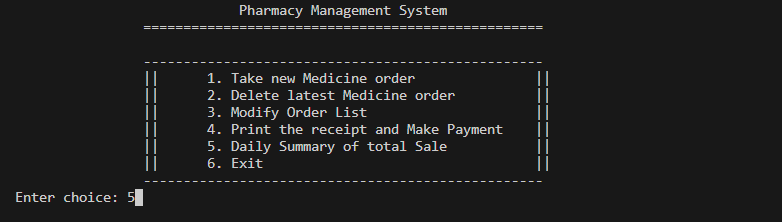
1. Menu(choice:4):



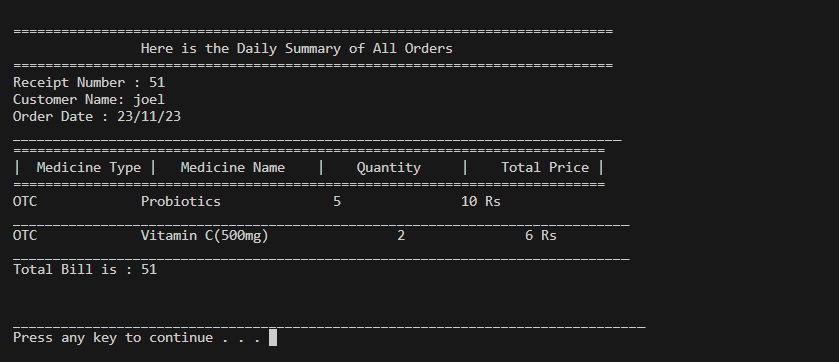
Printing a receipt of the order and making the payment:



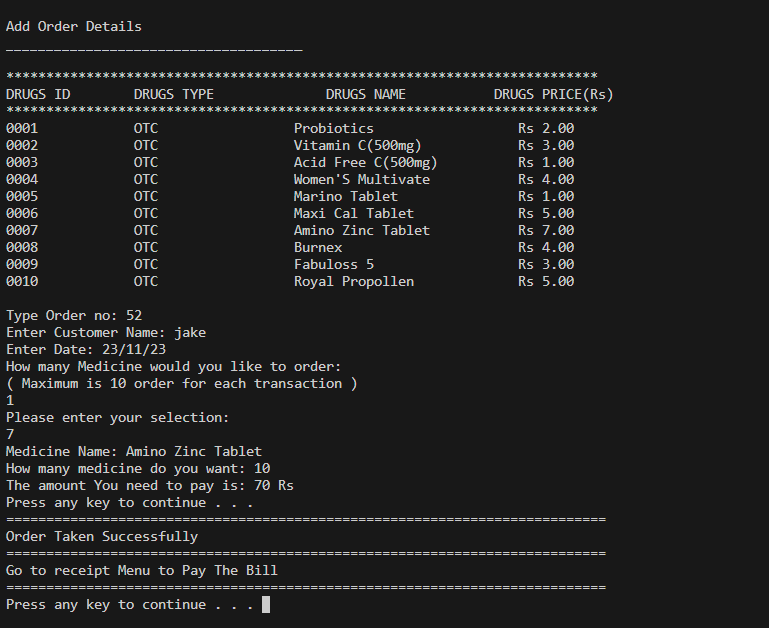
1. Menu(choice:5):

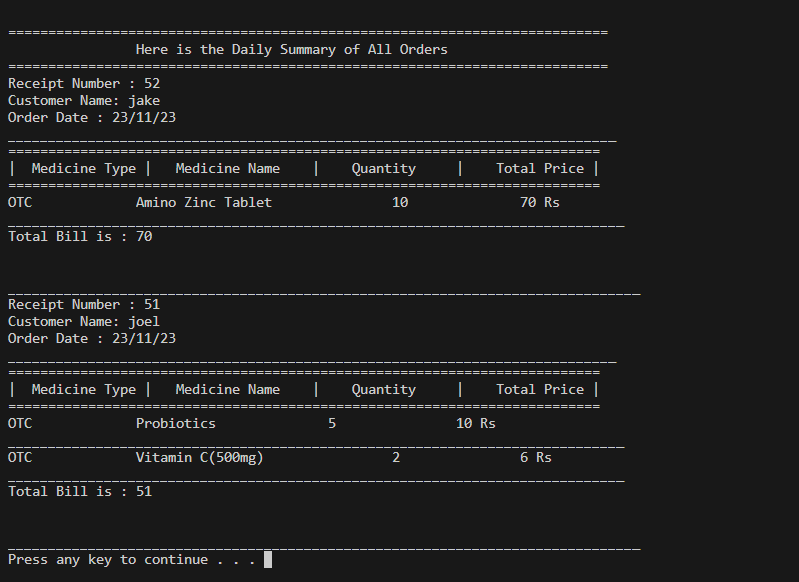


Daily summary of payment(s):

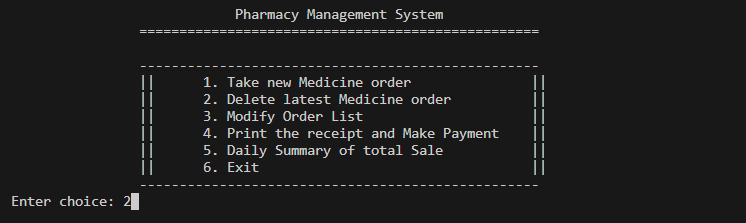


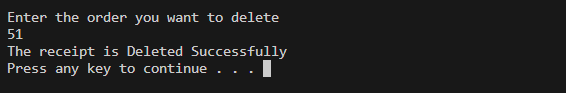
1. Taking another order(choice:1):



Daily summary of payment(s): 

1. Deleting an order(choice:2):





1. Exiting the program:

