REPUBLIC OF THE PHILIPPINES

BICOL UNIVERSITY

BICOL UNIVERSITY POLANGUI

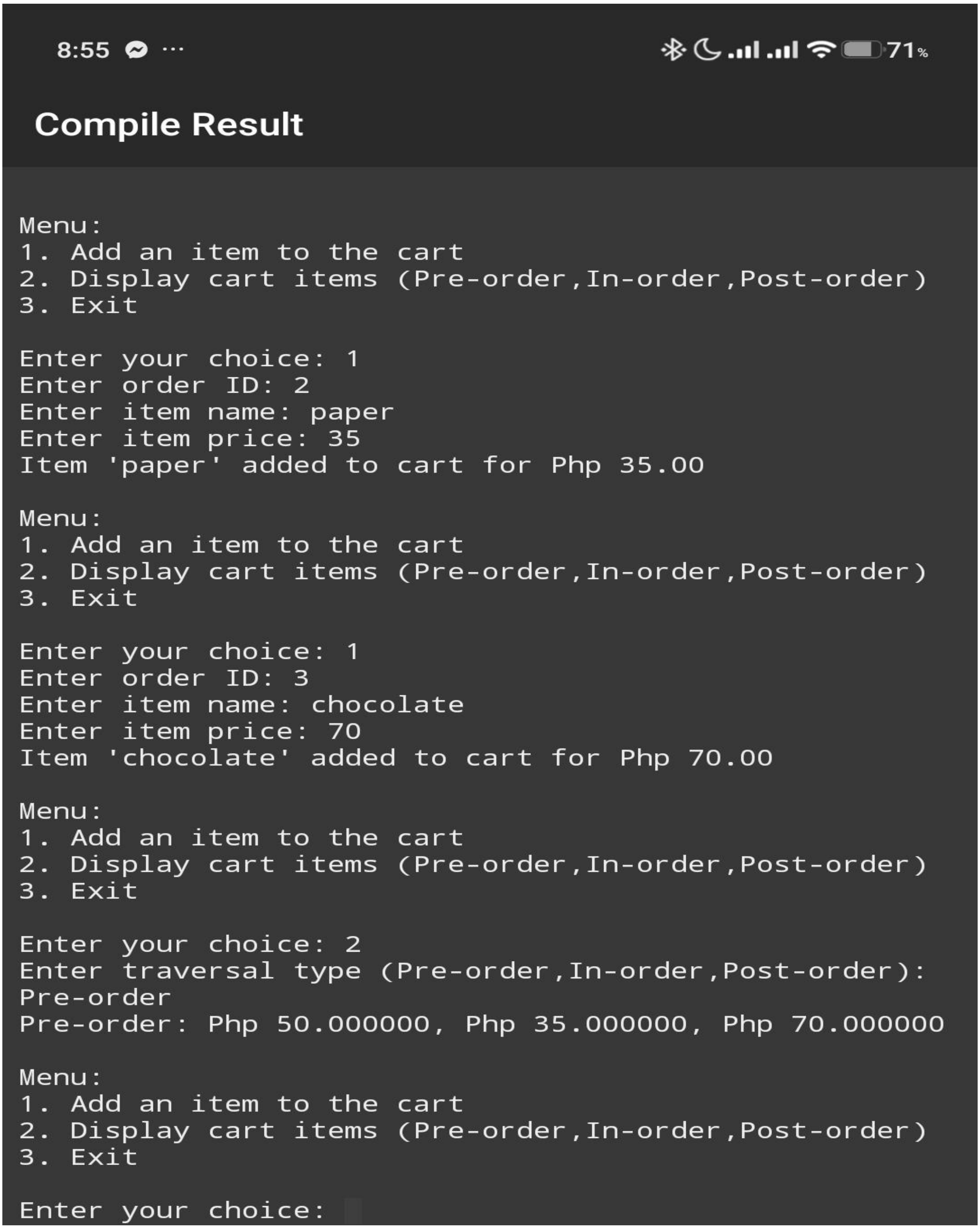
WELCOME TO THE SHOPPING CART SYSTEM CODE:

```
#include <iostream>
#include <iomanip>
#include <string>
Using namespace std;

// Node structure for the Binary Search Tree
Struct Node {
    Int orderId;
    String itemName;
    Double price;
    Node* left;
    Node* right;

    Node(int id, string name, double price) : orderId(id), itemName(name), price(price), left(nullptr), right(nullptr) {}
};

// Binary Search Tree class
Class ShoppingCart {
Private:
    Node* root;
    Int itemCount; // Track the number of items in the cart
    Const int MAX_ITEMS = 5; // Maximum allowed items

    // Helper function to insert an item
    Node* insert(Node* node, int orderId, string itemName, double price) {
        If (node == nullptr) {
            Return new Node(orderId, itemName, price);
        }
        If (price < node->price) {
```

```
      Node->left = insert(node->left, orderId, itemName, price);
    } else {
      Node->right = insert(node->right, orderId, itemName, price);
    }
    Return node;
  }

  // Helper function for pre-order traversal
  Void preOrderTraversal(Node* node, string& result) {
    If (node == nullptr) return;
    Result +=”Php “ + to_string(node->price) + “, “;
    preOrderTraversal(node->left, result);
    preOrderTraversal(node->right, result);
  }

  // Helper function for in-order traversal
  Void inOrderTraversal(Node* node, string& result) {
    If (node == nullptr) return;
    inOrderTraversal(node->left, result);
    result +=”Php “ + to_string(node->price) + “, “;
    inOrderTraversal(node->right, result);
  }

  // Helper function for post-order traversal
  Void postOrderTraversal(Node* node, string& result) {
    If (node == nullptr) return;
    postOrderTraversal(node->left, result);
    postOrderTraversal(node->right, result);
    result +=”Php “ + to_string(node->price) + “, “;
  }

Public:
  ShoppingCart() : root(nullptr), itemCount(0) {}

  // Public method to add an item to the cart
  Void addItem(int orderId, string itemName, double price) {
    If (itemCount >= MAX_ITEMS) {
      Cout << “The cart is full! You can only have up to “ << MAX_ITEMS << “ items.” << endl;
      Return;
    }
    Root = insert(root, orderId, itemName, price);
    itemCount++;
    cout << “Item ‘” << itemName << “’ added to cart for Php “ << fixed << setprecision(2) << price <<
endl;
  }

  // Public method to display items in a specified traversal order
  Void displayItems(string traversalType) {
    If (root == nullptr) {
      Cout << “The cart is empty!” << endl;
      Return;
    }
    String result;
    If (traversalType == “Pre-order”) {
      preOrderTraversal(root, result);
    } else if (traversalType == “In-order”) {
      inOrderTraversal(root, result);
    } else if (traversalType == “Post-order”) {
      postOrderTraversal(root, result);
    } else {
      Cout << “Invalid traversal type!” << endl;
      Return;
    }
    // Remove the trailing “, “ and display the result
    If (!result.empty()) result.pop_back(), result.pop_back();
    Cout << traversalType << “: “ << result << endl;
  }
};

Int main() {
  ShoppingCart cart;
  Int choice, orderId;
  String itemName, traversalType;
  Double price;
  Cout << “Welcome to the Shopping Cart System!” << endl;

  While (true) {
    Cout << “\nMenu:” << endl;
    Cout << “1. Add an item to the cart” << endl;
    Cout << “2. Display cart items (Pre-order,In-order,Post-order)” <<endl;
    Cout << “3. Exit” << endl;
    Cout << “\n”<<”Enter your choice: “;
    Cin >> choice;

    Switch (choice) {
      Case 1:
        Cout << “Enter order ID: “;
        Cin >> orderId;
        Cout << “Enter item name: “;
        Cin.ignore();
        Getline(cin, itemName);
        Cout << “Enter item price: “;
        Cin >> price;
        Cart.addItem(orderId, itemName, price);
        Break;
      Case 2:
        Cout << “Enter traversal type (Pre-order,In-order,Post-order): “<<”\n”;
        Cin.ignore();
        Getline(cin, traversalType);
        Cart.displayItems(traversalType);
        Break;
      Case 3:
        Cout << “Exiting the system. Goodbye!” << endl;
        Return 0;
      Default:
        Cout << “Invalid choice. Please try again.” << endl;
    }
  }
  Return 0;
}
```

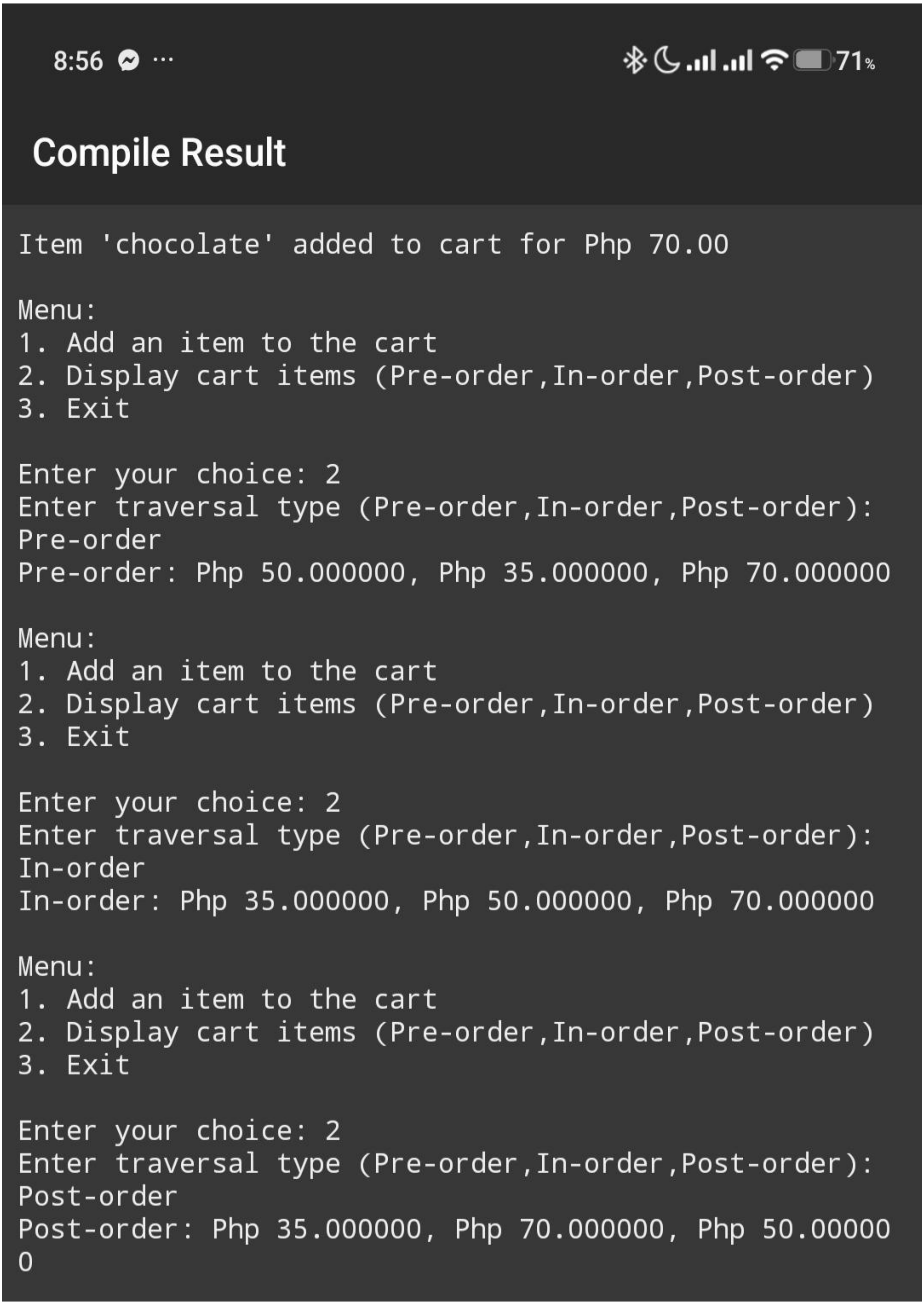
EXAMPLE OUTPUT:

PROCESS:



```
Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 1
Enter order ID: 2
Enter item name: paper
Enter item price: 35
Item 'paper' added to cart for Php 35.00

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 1
Enter order ID: 3
Enter item name: chocolate
Enter item price: 70
Item 'chocolate' added to cart for Php 70.00

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 2
Enter traversal type (Pre-order,In-order,Post-order):
Pre-order
Pre-order: Php 50.000000, Php 35.000000, Php 70.000000

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice:
```

DISPLAYING TRAVERSALS:



```
Item 'chocolate' added to cart for Php 70.00

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 2
Enter traversal type (Pre-order,In-order,Post-order):
Pre-order
Pre-order: Php 50.000000, Php 35.000000, Php 70.000000

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 2
Enter traversal type (Pre-order,In-order,Post-order):
In-order
In-order: Php 35.000000, Php 50.000000, Php 70.000000

Menu:
1. Add an item to the cart
2. Display cart items (Pre-order,In-order,Post-order)
3. Exit

Enter your choice: 2
Enter traversal type (Pre-order,In-order,Post-order):
Post-order
Post-order: Php 35.000000, Php 70.000000, Php 50.00000
0
```