Online Store Management System

A simple command-line application in C++ that simulates an online store with:

- **Admin panel**
    - Add / delete products in a BST catalogue
    - View checkout history
    - Change access PIN
- **User panel**
    - Browse products (in-order traversal of BST)
    - Search product by name
    - Place orders (FIFO order queue)
    - View pending orders
    - Checkout (moves orders into history)
- **Data structures used**
    - **Binary Search Tree** for product catalogue
    - **Queue** (linked list) for user orders
    - **Stack** (linked list) for checkout history

Usage

1. **Main menu**
    1. Login as Admin
    2. Login as User
    3. Exit
2. **Admin workflow**
    - Enter PIN (default: `1234`)
    - Admin Panel:
        1. Add Product
        2. Delete Product
        3. View Checkout History
        4. Change PIN
        5. Exit Admin Panel
3. **User workflow**
    - Enter your name
    - User Panel:
        1. Display Products
        2. Search Product by Name
        3. Place an Order
        4. Display Orders
        5. Checkout
        6. Exit User Panel

Key Classes & Functions

- **ProductBST**
    - insertProduct(id,name,price,qty)
    - deleteProduct(id)
    - findProductByName(name)
    - displayProducts()
- **OrderQueue**
    - enqueue(order)
    - dequeue()
    - displayOrders()
- **CheckoutHistory**
    - addOrderToHistory(order)
    - displayHistory()
- **Admin**
    - validatePin(pin)
    - changePin(newPin)
- **User**
    - placeOrder(bst,name,qty)
    - displayOrder()
    - checkout(admin)

Features & Data Flow

1. **Product Catalogue** lives in a BST sorted by `productId`.
2. **Users** place orders that enqueue into their personal `OrderQueue`.
3. **Checkout**
    - Dequeues each order → adds to `CheckoutHistory` → persists in a stack-like linked list
    - Stock quantity is decremented at order time.
4. **Admin** can review full history of all past checkouts.

Extensibility

- **Persistent storage**: Export/import products, orders, history to files or a DB.
- **Concurrency**: Allow multiple user threads with mutex-protected data structures.
- **GUI/web front-end**: Wrap CLI core in a web service or desktop app.