**ATM Cash Dispenser - README**

**Assigned Task**

The goal of this project was to simulate an ATM cash dispenser using C++ with the following features:

* Allow the user to add and remove currency denominations.
* Enable cash withdrawals using two different strategies:
  + **Greedy Algorithm**: Selects the highest denomination notes first.
  + **Optimal Algorithm**: Selects the combination of notes that minimizes the total number of notes used.
* Implement object-oriented principles including inheritance and polymorphism.
* Use dynamic memory allocation and recursion.

**How the Task Was Completed**

**1. Design and Implementation**

* A base class Dispenser was created to represent a cash dispenser. It contains virtual methods to be overridden by the two dispensing strategies.
* Two derived classes GreedyDispenser and OptimalDispenser implement different withdrawal strategies:
  + GreedyDispenser: Uses a simple loop to subtract the highest possible denomination until the requested amount is met or cannot continue.
  + OptimalDispenser: Uses depth-first search (DFS) and recursion to find the combination with the fewest notes.

**2. Data Structure**

* A struct Denomination holds the value of each note and a pointer to its available count.
* A dynamic array of denominations is maintained to allow adding or removing denominations at runtime.

**3. Memory Management**

* new and delete are used to dynamically allocate and manage memory for denominations and their counts.
* Proper memory cleanup is performed at program exit to avoid memory leaks.

**4. User Interaction**

* A menu-based interface is provided where users can:
  1. Add denominations
  2. Remove denominations
  3. Withdraw cash using the Greedy method
  4. Withdraw cash using the Optimal method
  5. View all current denominations
  6. Exit the program

**5. Testing**

* The program was tested by adding various denominations (e.g., 5000, 2000, 1000) and withdrawing different amounts using both Greedy and Optimal strategies.
* Cases with exact matches, leftover amounts, and insufficient funds were also tested.

**How to Use**

**Menu Options**

ATM Cash Dispenser

1. Add Denomination

2. Remove Denomination

3. Withdraw (Greedy)

4. Withdraw (Optimal)

5. Show Denominations

0. Exit

**Example**

Enter choice: 1

Enter denomination value and count: 2000 5

Added denomination: 2000 x 5

Enter choice: 4

Enter amount to withdraw: 4000

Take 2 x 2000

Withdrawal successful.

**Requirements**

* A working C++ environment (e.g., Dev C++, Code::Blocks, or any C++ compiler)
* Basic C++ knowledge to compile and run the code

**Concepts Demonstrated**

* Object-Oriented Programming: Inheritance and Polymorphism
* Dynamic Memory Allocation
* Recursion and Backtracking (in Optimal withdrawal)
* Greedy and Optimal Algorithms
* User Interaction via Console