DSA Lab: 10

Department of Computer Science

Iqra University Islamabad

DSA

Maqsood Ahmed

ID: 3818

Scenario:

You are working in an automated warehouse where products are stored on shelves in increasing order of size. Due to a system upgrade, all the items on Shelf A need to be moved to Shelf C while preserving their order. A temporary Shelf B is available for intermediate storage. The robot moving the items has the following constraints:

- **Only one item can be moved at a time.**
- **A larger item cannot be placed on top of a smaller item.**
- **The robot must follow an optimal sequence to minimize the number of moves.**

Task:

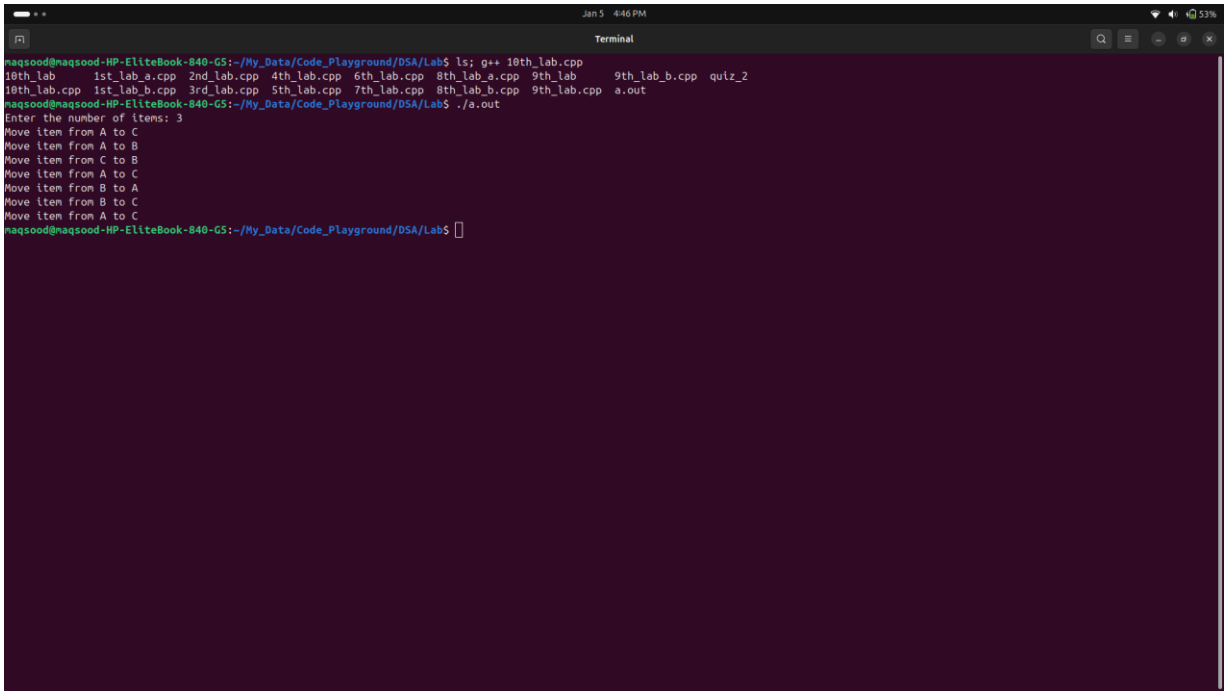
Write a recursive function to determine the sequence of moves required to transfer all items from Shelf A to Shelf C, using Shelf B as a temporary holding space.

Source Code:

```
#include <iostream>
using namespace std;

// Recursive function to solve the problem
void moveItems(int n, char fromShelf, char toShelf, char tempShelf) {
    if (n == 1) {
        cout << "Move item from " << fromShelf << " to " << toShelf << endl;
        return;
    }
    // Step 1: Move n-1 items from 'fromShelf' to 'tempShelf' using 'toShelf' as temporary storage
    moveItems(n - 1, fromShelf, tempShelf, toShelf);

    // Step 2: Move the nth item directly from 'fromShelf' to 'toShelf'
    cout << "Move item from " << fromShelf << " to " << toShelf << endl;

    // Step 3: Move the n-1 items from 'tempShelf' to 'toShelf' using 'fromShelf' as temporary storage
    moveItems(n - 1, tempShelf, toShelf, fromShelf);
}


int main() {
    int n; // Number of items on Shelf A
```

```
    cout << "Enter the number of items: ";
    cin >> n;

    // Call the recursive function
    moveItems(n, 'A', 'C', 'B');

    return 0;
}
```

OUTPUT:



The End