Section C

Given a pre-order traversal sequence of a BST, write a function to reconstruct the original BST. In a pre-order traversal, nodes are visited in the order: root, left, right.

Task: Implement a function `reconstructBST(preorder)` that takes a list representing a pre-order traversal and returns the reconstructed BST.

Example: Given the following pre-order traversal:

- Pre-order: [10, 5, 1, 7, 15, 12, 18]

The function should reconstruct the following BST:

```
   10
   /  \
  5   15
 / \  / \
1  7 12 18
```

```
template<typename T>
typename BST<T>::Node* buildBST(T* preorder, int start, int end) {
    if (start > end) {
        return nullptr;
    }

    T rootVal = preorder[start];
    typename BST<T>::Node* root = new typename BST<T>::Node(rootVal);

    int splitIndex = start + 1;
    while (splitIndex <= end && preorder[splitIndex] < rootVal) {
        splitIndex++;
    }

    root->left = buildBST(preorder, start + 1, splitIndex - 1);
    root->right = buildBST(preorder, splitIndex, end);

    return root;
}

template<typename T>
```

```
typename BST<T>::Node* reconstructBST(T* preorder, int size) {
    return buildBST(preorder, 0, size - 1);
}
```