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

struct node {
    int data;
    node* left;
    node* right;
};

node* createNode(int value) {
    node* newNode = new node;
    newNode->data = value;
    newNode->left = nullptr;
    newNode->right = nullptr;
    return newNode;
}

void insert(node*& root, node* temp) {
    if (root == nullptr) {
        root = temp;
    } else {
        if (root->data > temp->data) {
            if (root->left == nullptr) {
                root->left = temp;
            } else {
                insert(root->left, temp);
            }
        } else if (temp->data > root->data) {
            if (root->right == nullptr) {
                root->right = temp;
            } else {
```

```
            insert(root->right, temp);
        }
    } else {
        cout << "Duplicate element not inserted" << endl;
        delete temp;
    }
  }
}

void display(node* root) {
  if (root != nullptr) {
    display(root->left);
    cout << root->data << " ";
    display(root->right);
  }
}


int minValue(node* root) {
  while (root->left != nullptr) {
    root = root->left;
  }
  return root->data;
}

void swapNodes(node* root) {
  if (root == nullptr) {
    return;
  }

  swap(root->left, root->right);
  swapNodes(root->left);
```

```
    swapNodes(root->right);
}

int main() {
    node* root = nullptr;
    int choice;

    do {
        cout << "\nMenu\n";
        cout << "1) Insert  \n2) Display    \n3) Minimum Data Value  \n4) Swap  \n5) Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: {
                int value;
                cout << "Enter the value to insert: ";
                cin >> value;
                node* newNode = createNode(value);
                insert(root, newNode);
                break;
            }
            case 2: {
                cout << "BST Elements: ";
                display(root);
                cout << endl;
                break;
            }

            case 3: {
                int minVal = minValue(root);
```

```
            cout << "Minimum data value in the tree: " << minVal << endl;
            break;
        }
        case 4: {
            swapNodes(root);
            cout << "Tree nodes swapped." << endl;
            break;
        }
        case 5: {
            cout << "Exiting the program." << endl;
            break;
        }
        default:
            cout << "Invalid choice. Please enter a valid option." << endl;
      }
   } while (choice != 5);

   return 0;
}
```