```cpp
#include "balanced_tree.h"
#include "iostream"

using namespace std;


void main() {
	setlocale(0, "Rus");
	Tree* userTree = nullptr;

	while (true) {
		int user_choice = 0;

		cout << "Enter chosen action: \n"
			<< "1. Create tree\n"
			<< "2. Add element into a tree\n"
			<< "3. Show tree\n"
			<< "4. Show reversed tree\n"
			<< "5. Show tree height\n"
			<< "6. Show tree length\n"
			<< "7. Get average key value\n"
			<< "8. Delete tree\n"
			<< "9. Exit \n";
		cin >> user_choice;
		switch(user_choice){
			case 1: {
				int size = 0;
				cout << "\nEnter amount of elements: \n";
				cin >> size;
				if (size < 1) "\nIt is not a size\n";
				else {
					userTree = new Tree(new node(rand() % 1000 +
100));
					userTree->createTree(userTree->getRoot(),
size);
				}
				break;
			}
			case 2: {
				if (userTree != nullptr) {
					userTree->add(userTree->getRoot());
				}
				else {
					cout << "\nTree doesn't exist";
				}
				break;
			}
			case 3: {
				if (userTree != nullptr) {
					cout << "\n\n";
					userTree->print(userTree->getRoot(), 0);
					cout << "\n\n";
				}
				else {
					cout << "\nTree doesn't exist";
				}
				break;
			}
```

```
                    case 4: {
                          if (userTree != nullptr) {
                                cout << "\n\n";
                                userTree->reverseTreeRightToLeft(userTree-
>getRoot());
                                userTree->print(userTree->getRoot(),0);
                                userTree->reverseTreeRightToLeft(userTree-
>getRoot());
                                cout << "\n\n";
                          }
                          else {
                                cout << "\nTree doesn't exist";
                          }
                          break;
                    }
                    case 5: {
                          if (userTree != nullptr) {
                                cout << "\nTree height: " << userTree-
>countTreeHeight(userTree->getRoot()) << endl;
                          }
                          else {
                                cout << "\nTree doesn't exist";
                          }
                          break;
                    }
                    case 6: {
                          if (userTree != nullptr) {
                                cout << "\nTree length: " << userTree-
>getTreeLength(userTree->getRoot()) << endl;
                          }
                          else {
                                cout << "\nTree doesn't exist";
                          }
                          break;
                    }
                    case 7: {
                          if (userTree != nullptr) {
                                cout << "\nAverage: " << userTree-
>getAverage() << endl;
                          }
                          else {
                                cout << "\nTree doesn't exist";
                          }
                          break;
                    }
                    case 8: {
                          if (userTree != nullptr) {
                                cout << "\nSuccessfully deleted" << endl;
                                userTree->~Tree();
                                userTree = nullptr;
                          }
                          else {
                                cout << "\nTree doesn't exist";
                          }
                          break;
                    }
                    case 9: {
                          return;
```

```
                    }
                }
            }

        }
```