#include <algorithm>

#include <cstddef>

#include <utility>

#include <array>

#include <string>

#include <iostream>

template<typename T>

void tree\_sort(T array[], size\_t size) noexcept

{

struct tree\_t {

tree\_t \*left, \*right;

T value;

tree\_t(T &&\_value = T()) {

left = right = nullptr;

value = forward<T>(\_value);

}

T\* operator()(T array[]) {

if (this->left)

array = this->left->operator()(array);

\*array++ = move(this->value);

if (this->right)

array = this->right->operator()(array);

return array;

}

} root(move(array[0]));

for (size\_t idx = 1; idx < size; idx++) {

bool inserted = false;

tree\_t \*p\_root = &root;

while (!inserted) {

if (array[idx] < p\_root->value) {

if (p\_root->left) p\_root = p\_root->left;

else {

p\_root->left = new tree\_t(move(array[idx]));

inserted = true;

}

}

else {

if (p\_root->right) p\_root = p\_root->right;

else {

p\_root->right = new tree\_t(move(array[idx]));

inserted = true;

}

}

}

}

root(array);

}

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Введите числа (10): " << endl;

array <int, 10> arrInt;

for (int i = 0; i < 10; i++) {

cin >> arrInt[i];

}

cout << "Введите строки (10): " << endl;

array <string, 10> arrString;

for (int i = 0; i < 10; i++) {

cin >> arrString[i];

}

for (int i = 0; i < 10; i++) {

cout << arrInt[i] << ", ";

}

cout << endl;

for (int i = 0; i < 10; i++) {

cout << arrString[i] << ", ";

}

cout << endl << endl << endl;

tree\_sort<int>(arrInt.data(), arrString.size());

tree\_sort<string>(arrString.data(), arrString.size());

for (int i = 0; i < 10; i++) {

cout << arrInt[i] << ", ";

}

cout << endl;

for (int i = 0; i < 10; i++) {

cout << arrString[i] << ", ";

}

cout << endl;

system("pause");

return 0;

}

