#include <iostream>

#include <vector>

#include <memory>

using namespace std;

class Node {

private:

int key;

public:

vector<shared\_ptr<Node>> forward;

Node(int k, int level) : key(k), forward(level + 1, nullptr) {}

int getKey() const {

return key;

}

~Node() {

cout << "key " << key << " destroyed.\n";

}

};

#include <iostream>

#include <memory>

#include <vector>

using namespace std;

class SkipList {

private:

int MAXLVL; // Maximum level of the skip list

float P; // Probability of node promotion to higher levels (typically 0.5)

int level;

shared\_ptr<Node> header;

public:

SkipList(int, float);

bool searchElement(int);

void insertElement(int);

void deleteElement(int);

void displayList();

int randomLevel();

~SkipList() {

cout << "skip list destroyed.\n";

}

};