#include<vector>

#include<string>

class BinarySearchTree

{

private:

struct tree\_node

{

tree\_node\* left\_node;

tree\_node\* right\_node;

std::string question;

int number;

};

tree\_node\* root;

public:

BinarySearchTree()

{

root = nullptr;

}

void create\_animal\_tree( std::string file\_name );

tree\_node\*create\_tree\_node( int number, std::string question );

void connector( tree\_node\*&tem\_tree\_node, tree\_node \*&new\_tree\_node );

void root\_checker( tree\_node \*&new\_tree\_node );

void tree\_display( tree\_node\*&tem\_tree\_node );

void choice( tree\_node\*& next\_node, tree\_node\*& previews\_node,

int &loop\_guard );

int get\_answer();

void start\_to\_play( std::string file\_name );

void add\_node( tree\_node \*previous\_node );

void read\_in\_file( tree\_node \*tem\_tree\_node, std::fstream &out\_file );

void delete\_whole\_tree( tree\_node \*& tem\_tree\_node );

};