

```

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

class Node {
public:
    char character;
    int frequency;
    Node* left;
    Node* right;

    Node(char c, int f) {
        character = c;
        frequency = f;
        left = nullptr;
        right = nullptr;
    }

    Node(Node* l, Node* r) {
        character = '$';
        frequency = l->frequency + r->frequency;
        left = l;
        right = r;
    }
};

bool compareNodes(const Node* l, const Node* r) {
    return l->frequency > r->frequency;
}

void printHuffmanCodes(Node* root, string str) {
    if (!root) return;

    if (root->character != '$') {
        cout << root->character << ": " << str << endl;
    }

    printHuffmanCodes(root->left, str + "0");
    printHuffmanCodes(root->right, str + "1");
}

```

```

void buildHuffmanTree(vector<char>& characters, vector<int>& frequencies) {
    vector<Node*> nodes;
    for (int i = 0; i < characters.size(); i++) {
        nodes.push_back(new Node(characters[i], frequencies[i]));
    }

    while (nodes.size() > 1) {
        sort(nodes.begin(), nodes.end(), compareNodes);
        Node* left = nodes.back(); nodes.pop_back();
        Node* right = nodes.back(); nodes.pop_back();
        Node* newNode = new Node(left, right);
        nodes.push_back(newNode);
    }

    Node* root = nodes[0];

    // Print the Huffman codes
    printHuffmanCodes(root, "");
}

int main() {
    int n;
    cout << "Enter the number of characters: ";
    cin >> n;

    vector<char> characters(n);
    vector<int> frequencies(n);

    // Take user input for characters and their frequencies
    cout << "Enter the characters: ";
    for (int i = 0; i < n; i++) {
        cin >> characters[i];
    }

    cout << "Enter the frequencies: ";
    for (int i = 0; i < n; i++) {
        cin >> frequencies[i];
    }

    // Build the Huffman Tree and print the codes

```

```
    buildHuffmanTree(characters, frequencies);  
  
    return 0;  
}
```