**Part 1**

#include<iostream>

#include<queue>

#include<vector>

#include<string>

#include<map>

using namespace std;

struct node

{

char data;

int freq;

node\* left, \* right;

node(char data, unsigned freq)

{

left = right = NULL;

this->data = data;

this->freq = freq;

}

};

struct comparison

{

bool operator()(node\* l, node\* r)

{

return (l->freq > r->freq);

}

};

void traverse(struct node\* root, string str)

{

if (!root)

return;

if (root->data != '$')

cout << root->data << ": " << str << "\n";

traverse(root->left, str + "0");

traverse(root->right, str + "1");

}

node\* Huffman(map<char, int> ma)

{

struct node\* left, \* right, \* top;

priority\_queue<node\*, vector<node\*>, comparison> pq;

for (auto it = ma.begin();it != ma.end();it++)

{

pq.push(new node((\*it).first, (\*it).second));

}

while (pq.size() != 1)

{

left = pq.top();

pq.pop();

right = pq.top();

pq.pop();

top = new node('$', left->freq + right->freq);

top->left = left;

top->right = right;

pq.push(top);

}

return pq.top();

}

int main()

{

int freq[26] = { 0 };

string str;

cout << "Enter the string: ";

cin >> str;

map<char, int> ma;

for (int i = 0; i < str.length(); i++)

{

ma[str[i]]++;

}

node\* ptr = Huffman(ma);

traverse(ptr, "");

return 0;

}

**Output**

