```
/*
 * Problem Statement :- Write a program to implement Huffman Encoding using a greedy
 *                                    strategy.
 *
 * Time Complexity  : O(nlogn)    (n => number of unique characters in string)
 * Space Complexity : O(n)
 */

#include<bits/stdc++.h>
using namespace std;

class Node
{
      private:
            char data;
            int freq;
            Node *lchild, *rchild;
      public:
            Node(char d, int f=-1)
            {
                  data = d;
                  freq = f;
                  lchild = rchild = NULL;
            }
            Node(Node *lc, Node *rc)
            {
                  data = 0;
                  freq = lc->freq + rc->freq;
                  lchild = lc;
                  rchild = rc;
            }
            void traverse(string code="") const
          {
                  if(lchild != NULL)
                  {
                        lchild->traverse(code+'0');
                        rchild->traverse(code+'1');
                  }
                  else
                  {

      cout<<"\n\t"<<setw(10)<<data<<setw(10)<<freq<<setw(15)<<code<<endl;
                  }
          }
            bool operator<(const Node &a) const
            {
                  return freq > a.freq;
            }
};

void huffman_encoding(string str)
{
      priority_queue<Node> pq;
      vector<int> frequency(256, 0);

      for(unsigned int i=0; i<str.size(); i++)
```

```
	{
		frequency[str[i]]++;
	}
	for(int i=0; i<256; i++)
	{
		if(frequency[i] > 0)
		{
			pq.push(Node(i, frequency[i]));
		}
	}
	while(pq.size() >1)
	{
		Node *lc = new Node(pq.top());
		pq.pop();
		Node *rc = new Node(pq.top());
		pq.pop();
		pq.push(Node(lc, rc));
	}
	cout<<"\n\t The Huffman Code "<<endl;
	cout<<"\n\t"<<setw(10)<<"Data\t"<<setw(10)<<"Frequency"<<setw(15)<<"Huffman
Code"<<endl;
	pq.top().traverse();
}

int main()
{
	string str;

	cout<<"\n\t Enter String to find Huffman Codes : ";
	cin>>str;

	huffman_encoding(str);
}
```