Assignment 6

Name: Vishal Sule

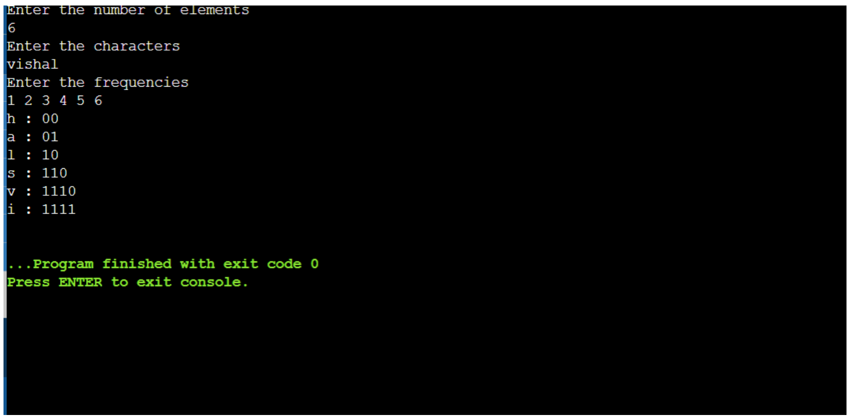
Roll No: 234

PRN: 0120190064

```
//code
#include <iostream>
#include <vector>
#include <queue>
#include <string>

using namespace std;

class Huffman_Codes
{
 struct New_Node
 {
  char data;
  size_t freq;
  New_Node* left;
  New_Node* right;
  New_Node(char data, size_t freq) : data(data),
                              freq(freq),
left(NULL),
right(NULL)
                              {}
 ~New_Node()
 {
   delete left;
   delete right;
 }
 };

 struct compare
 {
  bool operator()(New_Node* l, New_Node* r)
  {
    return (l->freq > r->freq);
  }
};

New_Node* top;

void print_Code(New_Node* root, string str)
{
if(root == NULL)
   return;

 if(root->data == '$')
 {
  print_Code(root->left, str + "0");
  print_Code(root->right, str + "1");
 }

 if(root->data != '$')
 {
   cout << root->data <<" : " << str << "\n";
   print_Code(root->left, str + "0");
   print_Code(root->right, str + "1");
 }
}

public:
  Huffman_Codes() {};
  ~Huffman_Codes()
  {
    delete top;
  }
  void Generate_Huffman_tree(vector<char>& data, vector<size_t>& freq, size_t size)
  {
   New_Node* left;
   New_Node* right;
   priority_queue<New_Node*, vector<New_Node*>, compare > minHeap;


for(size_t i = 0; i < size; ++i)

   {

      minHeap.push(new New_Node(data[i], freq[i]));

   }


while(minHeap.size() != 1)

    {

      left = minHeap.top();

minHeap.pop();


      right = minHeap.top();

minHeap.pop();


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

      top->left  = left;

      top->right = right;

      minHeap.push(top);

     }

    print_Code(minHeap.top(), "");

 }

};


 int main()

 {

  int n, f;

  char ch;

  Huffman_Codes set1;

  vector<char> data;

  vector<size_t> freq;

  cout<<"Enter the number of elements \n";

  cin>>n;

  cout<<"Enter the characters \n";

  for (int i=0;i<n;i++)

  {

      cin>>ch;
```

```
data.insert(data.end(), ch);

 }

 cout<<"Enter the frequencies \n";

 for (int i=0;i<n;i++)

 {

    cin>>f;

freq.insert(freq.end(), f);

 }


 size_t size = data.size();

 set1.Generate_Huffman_tree(data, freq, size);


 return 0;

}
```

// Output



```
Enter the number of elements
6
Enter the characters
vishal
Enter the frequencies
1 2 3 4 5 6
h : 00
a : 01
l : 10
s : 110
v : 1110
i : 1111


...Program finished with exit code 0
Press ENTER to exit console.
```