```
#include<bits/stdc++.h>
using namespace std;
struct MinHeapNode{
char data;
unsigned freq;
MinHeapNode *left,*right;
MinHeapNode(char data,unsigned freq)
{
   left=right=NULL;
   this->data=data;
   this->freq=freq;
}
};
struct compare{
bool operator()(MinHeapNode* l,MinHeapNode* r){
  return (l->freq>r->freq);
}
};
void printcodes(struct MinHeapNode* root,string str)
{
  if(!root)
   return;
  if(root->data != '$')
     cout<<root->data<<" : "<<str<<"\n";
  printcodes(root->left,str+"0");
```

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



}
void huffmancodes(char data[],int freq[],int arrsize)
{
   MinHeapNode *left,*right,*top;
   priority_queue<MinHeapNode*,vector<MinHeapNode*>,compare> minHeap;
   for(int i=0;i<arrsize;++i)
      minHeap.push(new MinHeapNode(data[i],freq[i]));


   while(minHeap.size()!=1)
   {
      left=minHeap.top();
      minHeap.pop();
      right=minHeap.top();
      minHeap.pop();
      top=new MinHeapNode('$',left->freq+right->freq);


      top->left=left;
      top->right=right;
        minHeap.push(top);
   }
   printcodes(minHeap.top(),"");
}
```

```
int main()
{
    char arr[]={'a','b','c','d','e','f'};
    int freq[]={5, 9, 12, 13, 16, 45};
    int arrsize=sizeof(arr)/sizeof(arr[0]);
    huffmancodes(arr,freq,arrsize);
}
```