AIM: PROGRAM FOR HUFFMAN ENCODING WITH GREEDY STRATEGY:

|  |
| --- |
| #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 != '$')  cout << root->data << ": " << str << "\n";  printCodes(root->left, str + "0");  printCodes(root->right, str + "1");  }  void HuffmanCodes(char data[], int freq[], int size)  {  struct MinHeapNode \*left, \*right, \*top;  priority\_queue<MinHeapNode\*, vector<MinHeapNode\*>, compare> minHeap;  for (int i = 0; i < size; ++i)  minHeap.push(new MinHeapNode(data[i], freq[i]));    while (minHeap.size() != 1) {  left = minHeap.top();  minHeap.pop();  right = minHeap.top();  minHeap.pop();    top = new MinHeapNode('$', 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 size = sizeof(arr) / sizeof(arr[0]);  HuffmanCodes(arr, freq, size);  return 0;  } |

Output:

