

**Name:** Phadtare Shivendra Uday

**Class:** BE Div: B

**Roll No:** COB434

**Batch:** B2

---

**Assignment No:2**

**Write a program to implement Huffman Encoding using a greedy strategy.**

**CODE:**

```
#include <iostream>
#include <queue>
#include <vector>
#include <unordered_map>

using namespace std;

struct Node {
char ch;
int freq;
Node *left, *right;

Node(char c, int f) {
    ch = c;
    freq = f;
    left = right = nullptr;
}
};

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

void printCodes(Node* root, string str) {
    if (!root) return;

    if (!root->left && !root->right) {
        cout << root->ch << ": " << str << "\n";
        return;
    }

    printCodes(root->left, str + "0");
    printCodes(root->right, str + "1");
}
```

```

void HuffmanCode(char data[], int freq[], int size) {
    priority_queue<Node*, vector<Node*>, compare> minHeap;

    for (int i = 0; i < size; i++) {
        minHeap.push(new Node(data[i], freq[i]));
    }

    while (minHeap.size() > 1) {
        Node *left = minHeap.top(); minHeap.pop();
        Node *right = minHeap.top(); minHeap.pop();

        Node *top = new Node('$', 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]);

    cout << "Character With their Huffman Codes:\n";
    HuffmanCode(arr, freq, size);

    return 0;
}

```

### Output:

```

(base) sspm@sspm:~$ g++ DAA.cpp (base)
sspm@sspm:~$ ./a.out
Character With their Huffman Codes:
f: 0
c: 100
d: 101
a: 1100
b: 1101
e: 111

```