Information Theory and Coding (ITC)

Roll No.-122CS0300
Name- Anmol Agrawal

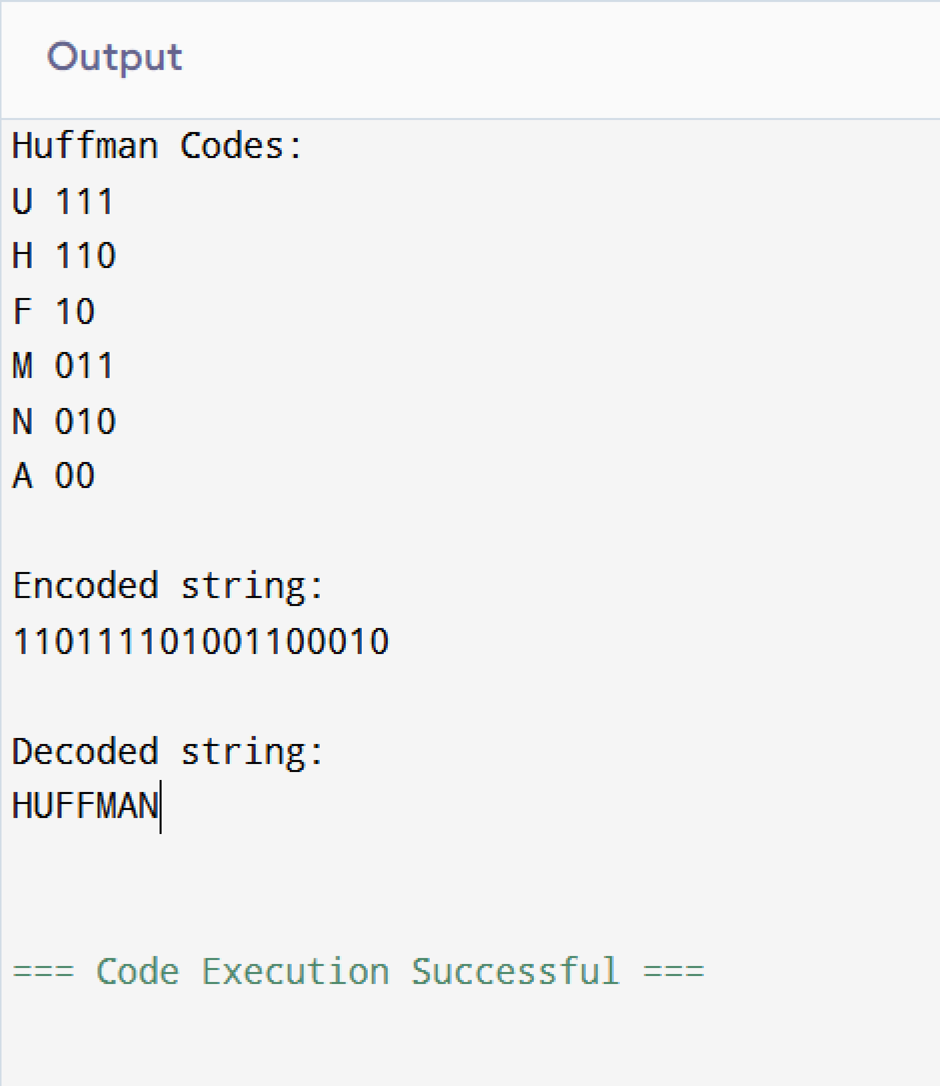
Date:05/02/2025

Classwork

Q.1 Huffman Coding (in cpp)

```
#include <iostream>
#include <queue>
#include <unordered_map>
#include <vector>
using namespace std;

#define ll long long
#define pb push_back
#define mp make_pair
#define vi vector<int>
#define vll vector<ll>
#define pi pair<int, int>
#define pq priority_queue

struct Node {
    char ch;
    int freq;
    Node *left, *right;
    Node(char c, int f) : ch(c), freq(f), left(nullptr), right(nullptr) {}
    Node(char c, int f, Node* l, Node* r) : ch(c), freq(f), left(l), right(r) {}
};

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

class HuffmanCoding {
public:
    unordered_map<char, string> huffmanCode;
```

```
Node* root;

void build(string text) {
    unordered_map<char, int> freq;
    for (char ch : text) freq[ch]++;
    pq<Node*, vector<Node*>, Compare> minHeap;
    for (auto p : freq) minHeap.push(new Node(p.first, p.second));
    while (minHeap.size() > 1) {
        Node *left = minHeap.top(); minHeap.pop();
        Node *right = minHeap.top(); minHeap.pop();
        minHeap.push(new Node('\0', left->freq + right->freq, left, right));
    }
    root = minHeap.top();
    encode(root, "");
}

void encode(Node* node, string str) {
    if (!node) return;
    if (node->ch != '\0') huffmanCode[node->ch] = str;
    encode(node->left, str + "0");
    encode(node->right, str + "1");
}

string getEncodedString(string text) {
    string str = "";
    for (char ch : text) str += huffmanCode[ch];
    return str;
}

string decode(string str) {
    string result = "";
    Node* curr = root;
    for (char bit : str) {
        curr = (bit == '0') ? curr->left : curr->right;
        if (curr->ch != '\0') {
            result += curr->ch;
            curr = root;
        }
    }
    return result;
    }
};

int main() {
    string text = "HUFFMAN";
    HuffmanCoding huff;
    huff.build(text);
    cout << "Huffman Codes:\n";
    for (auto p : huff.huffmanCode) cout << p.first << " " << p.second << "\n";
    string encoded = huff.getEncodedString(text);
    cout << "\nEncoded string:\n" << encoded << "\n";
    cout << "\nDecoded string:\n" << huff.decode(encoded) << "\n";
    return 0;
}
```



```
Huffman Codes:
U 111
H 110
F 10
M 011
N 010
A 00

Encoded string:
110111101001100010

Decoded string:
HUFFMAN


=== Code Execution Successful ===
```

[Note: Code was run on programmiz (online compiler)]