Assignment No. 10

Name – Manish Namdev Barage

PRN – 22520007

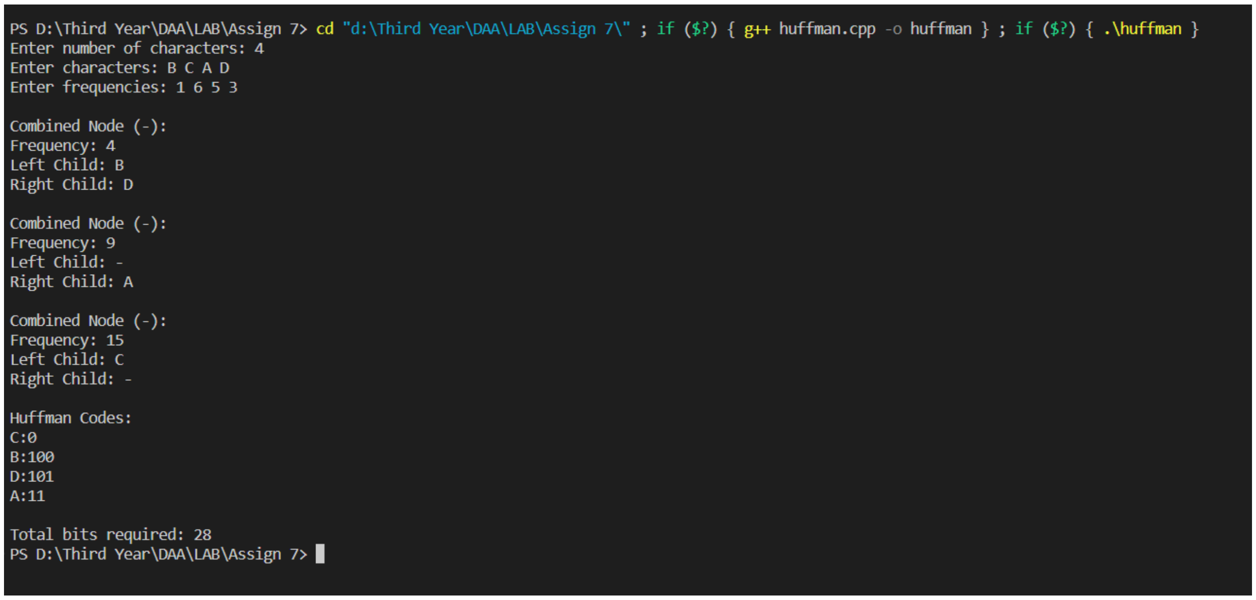
Batch – T7

Problem Statement – To Implement Huffman Coding

Code –

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

struct HuffmanNode {
    int data;
    char c;
    HuffmanNode* left;
    HuffmanNode* right;
};

struct MyComparator {
    bool operator()(HuffmanNode* x, HuffmanNode* y) {
        return x->data > y->data;
    }
};

int calculateTotalBits(HuffmanNode* root, string s) {
    if (root->left == nullptr && root->right == nullptr &&
isalpha(root->c)) {
        return s.length() * root->data;
    }

    int leftBits = calculateTotalBits(root->left, s + "0");
    int rightBits = calculateTotalBits(root->right, s + "1");

    return leftBits + rightBits;
}

void printCode(HuffmanNode* root, string s) {
    if (root->left == nullptr && root->right == nullptr &&
isalpha(root->c)) {
        cout << root->c << ":" << s << endl;
        return;
    }

    printCode(root->left, s + "0");
    printCode(root->right, s + "1");
}
```

```
int main() {
    int n;
    cout << "Enter number of characters: ";
    cin >> n;

    vector<char> charArray(n);
    vector<int> charFreq(n);

    cout << "Enter characters: ";
    for (int i = 0; i < n; i++) {
        cin >> charArray[i];
    }

    cout << "Enter frequencies: ";
    for (int i = 0; i < n; i++) {
        cin >> charFreq[i];
    }

    priority_queue<HuffmanNode*, vector<HuffmanNode*>, MyComparator>
q;

    for (int i = 0; i < n; i++) {
        HuffmanNode* hn = new HuffmanNode();
        hn->c = charArray[i];
        hn->data = charFreq[i];
        hn->left = nullptr;
        hn->right = nullptr;
        q.push(hn);
    }

    HuffmanNode* root = nullptr;
    while (q.size() > 1) {
        HuffmanNode* x = q.top();
        q.pop();
        HuffmanNode* y = q.top();
        q.pop();
        HuffmanNode* f = new HuffmanNode();
        f->data = x->data + y->data;
        f->c = '-';
        f->left = x;
        f->right = y;
        root = f;
        q.push(f);

        // Print step by step
        cout << "\nCombined Node (-):\n";
        cout << "Frequency: " << f->data << endl;
        cout << "Left Child: " << f->left->c << endl;
        cout << "Right Child: " << f->right->c << endl;
    }

    cout << "\nHuffman Codes:\n";
    printCode(root, "");
    int totalBits = calculateTotalBits(root, "");
    cout << "\nTotal bits required: " << totalBits << endl;

    return 0;
}
```



```
PS D:\Third Year\DAA\LAB\Assign 7> cd "d:\Third Year\DAA\LAB\Assign 7\" ; if ($?) { g++ huffman.cpp -o huffman } ; if ($?) { .\huffman }
Enter number of characters: 4
Enter characters: B C A D
Enter frequencies: 1 6 5 3

Combined Node (-):
Frequency: 4
Left Child: B
Right Child: D

Combined Node (-):
Frequency: 9
Left Child: -
Right Child: A

Combined Node (-):
Frequency: 15
Left Child: C
Right Child: -

Huffman Codes:
C:0
B:100
D:101
A:11

Total bits required: 28
PS D:\Third Year\DAA\LAB\Assign 7>
```