ITC CODING ASSIGNMENT

NAME- ANMOL AGRAWAL DATE-26/03/25
ROLL NO- 122CS0300

ARITHMETIC ENCODING:

```

```
#include <bits/stdc++.h>
using namespace std;

struct Interval {
    double low, high;
};

// Function to compute the probability of each unique character
map<char, double> computeProbabilities(const string& str) {
    map<char, int> freq;
    for (char c : str){
        freq[c]++;
    }
    map<char, double> prob;
    int len = str.size();
    for (auto [ch, count] : freq) {
        prob[ch] = static_cast<double>(count) / len;
    }
    return prob;
}

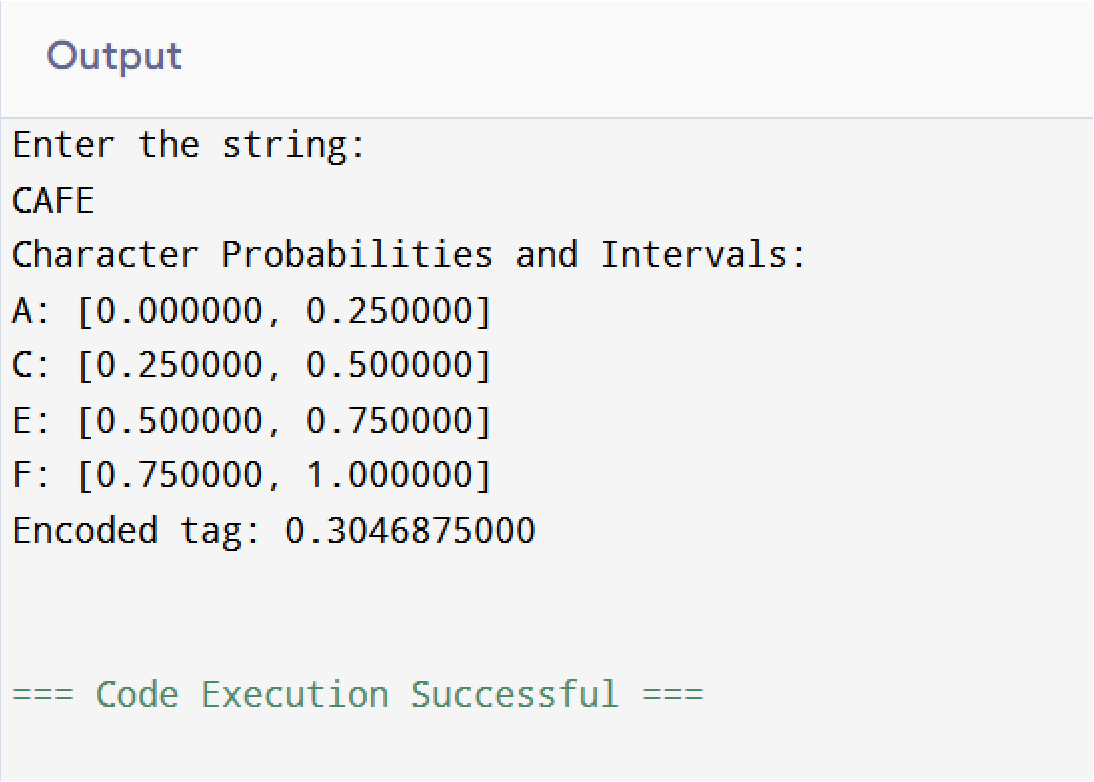
// Function to compute cumulative probabilities and intervals
map<char, Interval> computeIntervals(const map<char, double>&
probabilities) {
```
```

```
    map<char, Interval> intervals;
    double cumulative = 0.0;
    for (auto [ch, prob] : probabilities) {
        intervals[ch] = {cumulative, cumulative + prob};
        cumulative += prob;
    }
    return intervals;
}

// Arithmetic encoding function
double arithmeticEncode(const string& str, const map<char,
Interval>& intervals) {
    double low = 0.0, high = 1.0;
    for (char c : str) {
        double range = high - low;
        high = low + range * intervals.at(c).high;
        low = low + range * intervals.at(c).low;
    }
    return low;
}

int main() {
    string str;
    cout<<"Enter the string:"<<endl;
    cin>>str;
    auto probabilities = computeProbabilities(str);
    auto intervals = computeIntervals(probabilities);

    cout << "Character Probabilities and Intervals:" << endl;
    for (auto [ch, interval] : intervals) {
        cout << ch << ": [" << fixed << setprecision(6) <<
interval.low << ", " << interval.high << "]" << endl;
    }
```

````
    double tag = arithmeticEncode(str, intervals);
    cout << "Encoded tag: " << fixed << setprecision(10) << tag
<< endl;

    return 0;
}
````

~~~
```
~~~

->OUTPUT:

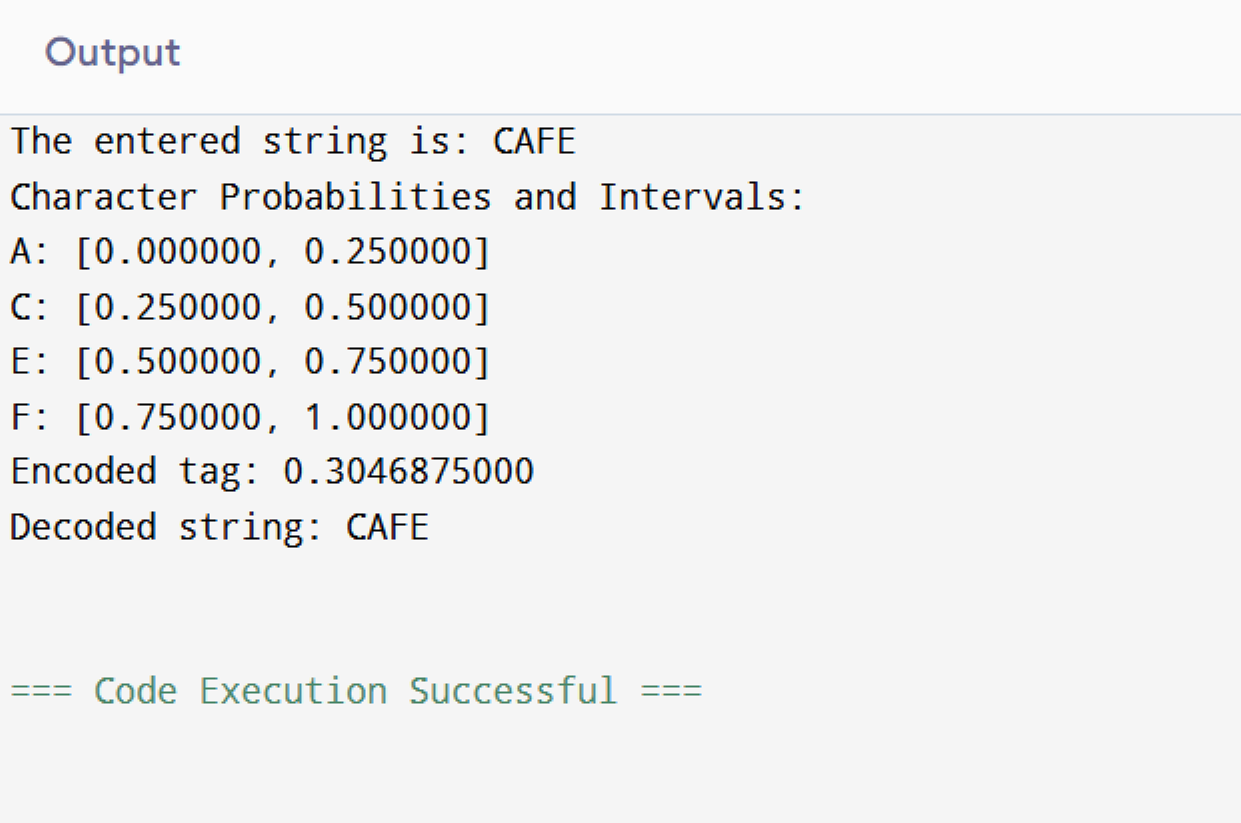


```
Enter the string:
CAFE
Character Probabilities and Intervals:
A: [0.000000, 0.250000]
C: [0.250000, 0.500000]
E: [0.500000, 0.750000]
F: [0.750000, 1.000000]
Encoded tag: 0.3046875000


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

#ARITHMETIC DECODING:

```
//Function to decode the string:
string arithmeticDecode(double tag, const map<char, Interval>&
intervals, int len) {
    string decoded_str;
    for (int i = 0; i < len; i++) {
        for (auto [ch, interval] : intervals) {
            if (tag >= interval.low && tag < interval.high) {
                decoded_str += ch;
                tag = (tag - interval.low) / (interval.high -
interval.low);
                break;
            }
        }
    }
    return decoded_str;
}
```

=>OUTPUT:



```
The entered string is: CAFE
Character Probabilities and Intervals:
A: [0.000000, 0.250000]
C: [0.250000, 0.500000]
E: [0.500000, 0.750000]
F: [0.750000, 1.000000]
Encoded tag: 0.3046875000
Decoded string: CAFE


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