PRACTICAL-2

AIM: Write a program to generate binary code in case of arithmetic coding.

Code:

```
#include <iostream>
#include <unordered_map>
#include <vector>
using namespace std;
struct node
{    double prob, range_from, range_to;     };
double encoding(unordered_map<char, node> arr, string s)
{
   cout << "\nEncoding\n";
   double low_v = 0.0, high_v = 1.0, diff = 1.0;
   cout << "Symbol\tLow_v\tHigh_v\tdiff\n";
   for (int i = 0; i < s.size(); i++)
   {   high_v = low_v + diff * arr[s[i]].range_to;
      low_v = low_v + diff * arr[s[i]].range_from;
      diff = high_v - low_v;
      cout << s[i] << "\t" << low_v << "\t" << high_v << "\t" << diff << endl;    }
   return low_v;    }
string decoding(unordered_map<char, node> arr, double code_word, int len)
{  cout << "\nDecoding: \n";
   char ch;
   string text = "";
   int j = 0;
   unordered_map<char, node>::iterator it;
   cout << "Code\tOutput\tRange_from\tRange_to\n";
   while (j < len)
   {  cout << code_word << "\t";
      for (it = arr.begin(); it != arr.end(); it++)
      {  char i = (*it).first;
         if (arr[i].range_from <= code_word && code_word < arr[i].range_to)
         {  ch = i;
       code_word = (code_word - arr[i].range_from) / (arr[i].range_to - arr[i].range_from);
            break;          }       }
      cout << ch << "\t" << arr[ch].range_from << "\t\t" << arr[ch].range_to << endl;
      text += ch;
      j++;    }
   return text;     }
int main()
{  int n;
   cout << "Enter number of characters: ";
   cin >> n;
```

```
    unordered_map<char, node> arr;
    vector<char> ar;
    double range_from = 0;
    cout << "Enter probability of each character:\n";
    for (int i = 0; i < n; i++)
    {  char ch;
       cin >> ch;
       ar.push_back(ch);
       cin >> arr[ch].prob;
       arr[ch].range_from = range_from;
       arr[ch].range_to = range_from + arr[ch].prob;
       range_from = arr[ch].range_to;                              }
    cout << "Symbol\tProbability\tRange_from\tRange_to\n";
    cout << "   \n";
    for (int i = 0; i < ar.size(); i++)
    {   char ch = ar[i];
 cout << ch << "\t" << arr[ch].prob << "\t\t" << arr[ch].range_from << "\t\t" <<
arr[ch].range_to << endl;      }
    cout << endl;
    string s;
    cout << "Enter text: ";
    cin >> s;
    double code_word = encoding(arr, s);
    cout << "Code word for " << s << " is: " << code_word << endl;
    string text = decoding(arr, code_word, s.size());
    cout << "Text for " << code_word << " is: " << text << endl;          }
```

Output: