

## PRACTICAL-2

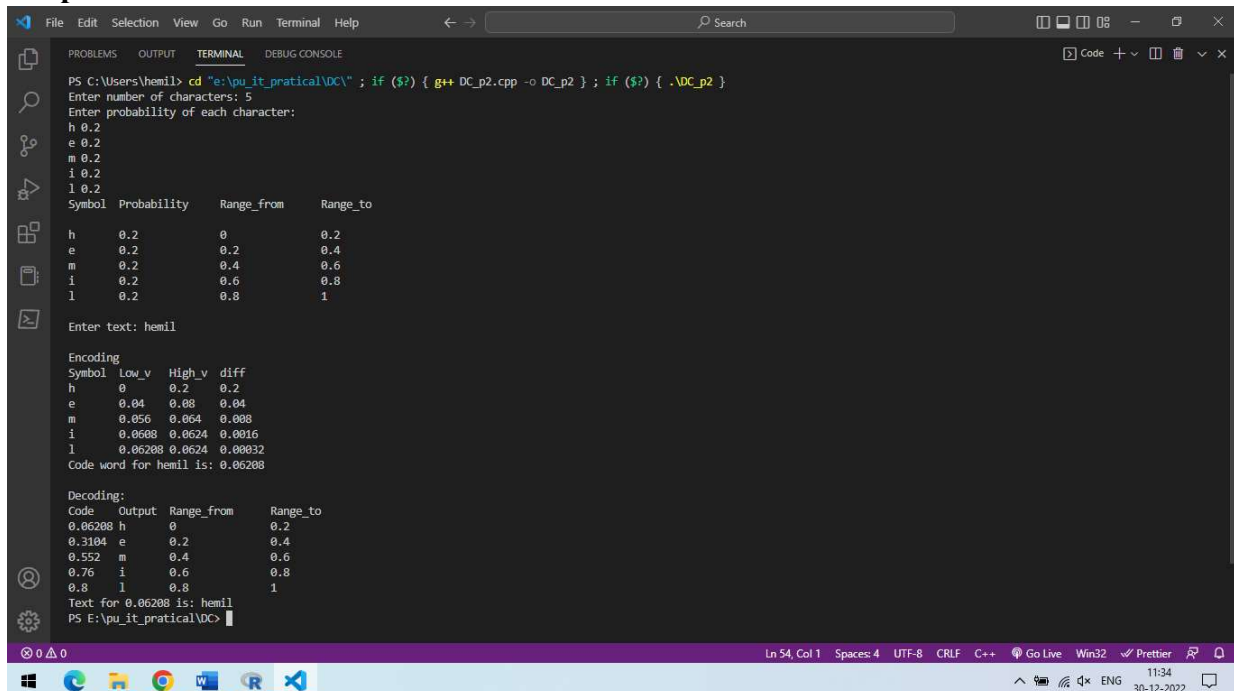
**AIM: Write a program to generate binary code in case of arithmetic coding.**

**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:



```
PS C:\Users\hemil> cd "e:\pu_it_practical\DC\" ; if ($?) { g++ DC_p2.cpp -o DC_p2 } ; if ($?) { .\DC_p2 }
Enter number of characters: 5
Enter probability of each character:
h 0.2
e 0.2
m 0.2
i 0.2
l 0.2
Symbol Probability Range_from Range_to
h 0.2 0 0.2
e 0.2 0.2 0.4
m 0.2 0.4 0.6
i 0.2 0.6 0.8
l 0.2 0.8 1
Enter text: hemil
Encoding
Symbol Low_v High_v diff
h 0 0.2 0.2
e 0.04 0.08 0.04
m 0.056 0.064 0.008
i 0.0608 0.0624 0.0016
l 0.06208 0.0624 0.00032
Code word for hemil is: 0.06208
Decoding:
Code Output Range_from Range_to
0.06208 h 0 0.2
0.3104 e 0.2 0.4
0.552 m 0.4 0.6
0.76 i 0.6 0.8
0.8 l 0.8 1
Text for 0.06208 is: hemil
PS E:\pu_it_practical\DC>
```