CODE:-

```
import math
def huffman(eg):
    assert(sum(eg.values())) == 1.0
    if (len(eg)==2):
        return dict(zip(eg.keys(), ['0', '1']))
    p_prime = eg.copy()
    a1, a2 = lowest_prob_pair(eg)
    p1, p2 = p_prime.pop(a1), p_prime.pop(a2)
    p_prime[a1 + a2] = round(p1 + p2,3)

    c = huffman(p_prime)
    ca1a2 = c.pop(a1 + a2)
    c[a1], c[a2] = ca1a2 + '0', ca1a2 + '1
    return c

def lowest_prob_pair(eg):
    assert(len(eg) >= 2)

    sorted_p = sorted(eg.items(), key=lambda x:x[1],reverse=True)
    return sorted_p[-2][0], sorted_p[-1][0]

def descendingorder(eg):
    sorted_p = sorted(eg.items(), key=lambda x: x[1], reverse=True)
    return sorted_p

def entropy(eg):
    x = []
    for k, v in eg.items():
        eg[k] = float(v)
        x.append(eg[k] * math.log2(1 / eg[k]))
        entropy = sum(x)
    return round(entropy,3)

def codelength(code,prob):
    x=[]
    y=[]
    z=[]
    for k, v in code.items():
        code[k] = len(v)
        x.append(code[k])
    for k, v in prob.items():
        prob[k] = float(v)
        y.append(prob[k])
    for i in range(0,len(code)):
        z.append(x[i]*y[i])
    length = sum(z)
    return float(round(length,3))

def code_rate(u,v):
    eff = u/v
    return round(eff,3)
n =int(input('Enter the numbers of the symbols:-'))
a = dict(input("Enter symbols and Probabilities: ").split() for _ in range(n))
for k, v in a.items():
    a[k] = float(v)
print('our Input sequence is',a)
s = dict(descendingorder(a))
final = huffman(a)
print('The descending order of sequence = ',descendingorder(a))
print('The huffman code of the sequence is:-',huffman(a))
x = entropy(a)
print('The entropy = ',x)
y = codelength(final,s)
print('The Average code length =',y)
rate = code_rate(x,y)
print('The code rate is',rate)
print('The efficiency is', rate*100,'%')
print('The code redundancy is',round((1-rate)*100,3),'%')
print('Made by Varad Patil 120A2036')
```

OUTPUT:-

```
Enter the numbers of the symbols:-3
Enter symbols and Probabilities: a 0.3
Enter symbols and Probabilities: b 0.1
Enter symbols and Probabilities: c 0.6
our Input sequence is {'a': 0.3, 'b': 0.1, 'c': 0.6}
The descending order of sequence =  [('c', 0.6), ('a', 0.3), ('b', 0.1)]
The huffman code of the sequence is:- {'c': '0', 'a': '10', 'b': '11'}
The entropy =  1.295
The Average code length = 1.4
The code rate is 0.925
The efficiency is 92.5 %
The code redundancy is 7.5 %
Made by Varad Patil 120A2036
```