

P[10] = 0.664833 norm: 9388.84
P[11] = 0.462257
P[12] = 0.390924
P[13] = 0.351915
P[14] = 0.326709
P[15] = 0.308876
P[25] = 0.244955
P[50] = 0.21658
P[100] = 0.205577
P[1000] = 0.196941
P[10010] = 0.196133
B(10,9.6) = 0.196044

ρ [10] = 0.96
 ρ [11] = 0.919658
 ρ [12] = 0.897452
 ρ [13] = 0.882037
 ρ [14] = 0.870388
 ρ [15] = 0.86116
 ρ [25] = 0.818906
 ρ [50] = 0.793667
 ρ [100] = 0.782382
 ρ [1000] = 0.772827
 ρ [10009] = 0.7719
 Π [10] = 0
 Π [11] = 0.256322
 Π [12] = 0.258421
 Π [13] = 0.252593
 Π [14] = 0.246754
 Π [15] = 0.241748
 Π [25] = 0.21857
 Π [50] = 0.205965
 Π [100] = 0.200737
 Π [1000] = 0.19651
 Π [10009] = 0.19611
B(10,9.6) = 0.196044

Theory Output

```

1/
*****
*****
2
3 Name:           Z#:
4 Course: Queueing Theory
5 Professor: Dr. Robert Cooper
6 Due Date:           Due Time:
7 Total Points: 100
8 Assignment:
9
10 Description:
11
12 *****
    *****/
13 #include <iostream>
14 #include <fstream>
15 #include <cmath>
16 using namespace std;
17
18 #include "erlang.h"
19 int main()
20 {
21     double P10[10010]; // final array!
22
23     double Pj[10010];
24     double Pij[10010];
25     double rho[10010];
26     double gamma[10010];
27     float norm;
28     const int SERVERS=10;
29     const double load=9.6;
30     Pj[0] = 1;
31     // for calculations assume servers = 10 so Pj[n] really is Psub(10)[n]
32     for (int n=10; n < 10010; n++)
33     {
34         for (int s = 1; s <= SERVERS; s++)
35         {
36             double aHat = (load)/(n-load);
37             Pj[s] = (n-s+1)*(aHat)*Pj[s-1]/s;
38         }
39         // normalize
40         norm = 0;
41         for (int i = 0; i<=n; i++)
42             norm += Pj[i];
43 //         for (int i= 0; i <= n; i++)
44 //             Pj[i] /= norm;
45         // do stuff here

```

HW4.cpp

```

46 /*      Pj[n] = ldchoose(n, 10)*pow(9.6/n,10)*pow((1-(9.6/n)),n-10);
47      norm = 0;
48      for (int k=0; k <= min(n, 10); k++)
49          norm += ldchoose(n, k)*pow(9.6/n,min(k, 10))*pow((1-(9.6/n)), n -
50      k); */
51      P10[n] = Pj[10] / norm ;
52  }
53
54  // for (int k = 1; k < 1001; k++) Pj[k] /= norm;
55
56  cout << "P[10] = " << P10[10] << " norm: " << norm << endl;
57  cout << "P[11] = " << P10[11] << endl;
58  cout << "P[12] = " << P10[12] << endl;
59  cout << "P[13] = " << P10[13] << endl;
60  cout << "P[14] = " << P10[14] << endl;
61  cout << "P[15] = " << P10[15] << endl;
62  cout << "P[25] = " << P10[25] << endl;
63  cout << "P[50] = " << P10[50] << endl;
64  cout << "P[100] = " << P10[100] << endl;
65  cout << "P[1000] = " << P10[1000] << endl;
66  cout << "P[10010] = " << P10[10009] << endl;
67  cout << "B(10,9.6) = " << B(10, 9.6) << endl;
68  ofstream out;
69  out.open("HW4.txt");
70  for (int i= 0; i < 1001; i++)
71      out << "P[" << i << "] = " << P10[i] << " gamma: " << gamma[i] <<
72  endl;
73
74  // utilization
75  for (int n=1; n < 10010; n++)
76  {
77      // do stuff here
78      rho[n] = 9.6*(1.0-(1.0-10.0/n)*P10[n])/10.0;
79  }
80
81
82  cout << "\u03C1[10] = " << rho[10] << endl;
83  cout << "\u03C1[11] = " << rho[11] << endl;
84  cout << "\u03C1[12] = " << rho[12] << endl;
85  cout << "\u03C1[13] = " << rho[13] << endl;
86  cout << "\u03C1[14] = " << rho[14] << endl;
87  cout << "\u03C1[15] = " << rho[15] << endl;
88  cout << "\u03C1[25] = " << rho[25] << endl;
89  cout << "\u03C1[50] = " << rho[50] << endl;
90  cout << "\u03C1[100] = " << rho[100] << endl;
91  cout << "\u03C1[1000] = " << rho[1000] << endl;

```

```

92  cout << "\u03C1[100009] = " << rho[10009] << endl;
93
94  // log the results
95
96  for (int i= 0; i < 10010; i++)
97      out << "\u03C1[" << i << "] = " << rho[i] << endl;
98
99  norm = 0;
100
101  for (int n=10; n < 10010; n++)
102  {
103      // do stuff here since I calculated rho above, use it!
104  /*
105      double aHat = (load)/(n-load);
106      Pj[n] = (n-10)*aHat*P10[n-1]/load;
107  */
108      // didnt work so go old school.
109      Pj[0] = 1;
110      // for calculations assume servers = 10 so Pj[n] really is Psub(10)
111  [n]
112      double aHat = (load)/(n-load);
113      for (int s = 1; s <= SERVERS; s++)
114      {
115          Pj[s] = ldchoose(n-1, s)*pow(aHat,s);
116      }
117      // normalize
118      Pj[n] = ldchoose(n-1, 10)*pow(aHat,10);
119      norm = 0;
120      for (int i = 1; i<=SERVERS; i++)
121          norm += Pj[i];
122  //      for (int i= 0; i <= n; i++)
123  //          Pj[i] /= norm;
124      // do stuff here
125
126      Pij[n] = Pj[n] / norm;
127  }
128
129
130  cout << "\u03A0[10] = " << Pij[10] << endl;
131  cout << "\u03A0[11] = " << Pij[11] << endl;
132  cout << "\u03A0[12] = " << Pij[12] << endl;
133  cout << "\u03A0[13] = " << Pij[13] << endl;
134  cout << "\u03A0[14] = " << Pij[14] << endl;
135  cout << "\u03A0[15] = " << Pij[15] << endl;
136  cout << "\u03A0[25] = " << Pij[25] << endl;
137  cout << "\u03A0[50] = " << Pij[50] << endl;
138  cout << "\u03A0[100] = " << Pij[100] << endl;

```

```
139 cout << "\u03A0[1000] = " << Pij[1000] << endl;
140 cout << "B(10,9.6) = " << B(10, 9.6) << endl;
141 // log the results
142
143 for (int i= 0; i < 1001; i++)
144     out << "\u03A0[" << i << "] = " << Pij[i] << endl;
145
146
147
148 }
149
150
151
152
```