

MP307 Practical 2 Discrete Population Models

Name: Dara Corr

Student ID no. : 18483836

```
In [17]: # import some standard packages
import numpy as np
import matplotlib.pyplot as plt
```

```
In [ ]:
```

Q.1

Consider the two state telephone example considered in class with transition matrix $P(i, j)$ stored in `P`

```
In [24]: P=np.array([[1/2,1/2],[2/3,1/3]])
print(P)
```

```
[[0.5      0.5      ]
 [0.66666667 0.33333333]]
```

$\text{Prob}(i \rightarrow j)$ is found by evaluating P^5 . We may multiply two matrices using the `np.matmul` Python function

```
In [25]: np.matmul(P,P)
```

```
Out[25]: array([[0.58333333, 0.41666667],
 [0.55555556, 0.44444444]])
```

We may also take the power of a matrix using the `np.linalg.matrix_power` Python function

```
In [26]: np.linalg.matrix_power(P,2)
```

```
Out[26]: array([[0.58333333, 0.41666667],
 [0.55555556, 0.44444444]])
```

```
In [6]: np.linalg.matrix_power(P,5)
```

```
Out[6]: array([[0.57137346, 0.42862654],
 [0.57150206, 0.42849794]])
```

Notice that the rows are converging which suggests that the system is ergodic. In fact the system is ergodic with equilibrium probabilities $\pi_0 = 3/7$ and $\pi_1 = 4/7$

In [7]: `print(4/7)`
`print(3/7)`

```
0.5714285714285714
0.42857142857142855
```

The equilibrium probabilities are found by looking for the **left** eigenvector of P for eigenvalue 1. We can find all eigenvectors and eigenvalues of P by use of the `np.linalg.eig` Python function as follows.

In [27]: `eigendata=np.linalg.eig(P)`
`print(eigendata)`

```
(array([ 1.          , -0.16666667]), array([[ 0.70710678, -0.6       ],
   [ 0.70710678,  0.8       ]]))
```

The output is a tuple consisting of an array and a matrix. We may unpack the tuple as follows:

In [9]: `X, V=eigendata`

In [10]: `print(X)`
`print(V)`

```
[ 1.          -0.16666667]
[[ 0.70710678 -0.6       ]
 [ 0.70710678  0.8       ]]
```

The eigenvalues of P are given in the array `X` with eigenvectors given in the matrix `V`. In particular the eigenvector for 1 is the first column vector with equal entries. We can read this off the entries of the first column as an array:

In [12]: `u=V[:,0] # Note that Python indices run as 0,1, ...`
`print(u)`

```
[0.70710678  0.70710678]
```

A constant scalar multiple of an eigenvector is also an eigenvector so in fact we have confirmed that $Pu = u$ for $u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Thus the `np.linalg.eig` Python function returns the **right** eigenvectors of P . Therefore we consider the right eigenvectors of the transpose of P , denoted by P^T , which provide us with the left eigenvectors of P .

In [13]: `PT=P.transpose()`
`print(PT)`

```
[[0.5      0.66666667]
 [0.5      0.33333333]]
```

```
In [14]: eigendata=np.linalg.eig(PT)
```

```
In [15]: X, V=eigendata
print(X)
print(V)
```

```
[ 1.           -0.16666667]
[[ 0.8          -0.70710678]
 [ 0.6          0.70710678]]
```

Thus the left eigenvector of P for eigenvalue 1 is

```
In [16]: u=V[:,0]
print(u)
```

```
[0.8 0.6]
```

We normalize this eigenvector so that the entries sum to unity. We thus obtain the equilibrium probability vector π

```
In [17]: usum=u[0]+u[1]
pi=u/usum
print(pi)
print([4/7,3/7])
```

```
[0.57142857 0.42857143]
[0.5714285714285714, 0.42857142857142855]
```

```
In [ ]:
```

Q.2

A finite queue of maximum size 3 is observed with the following transition matrix

$$\begin{bmatrix} 1/3 & 0 & 2/5 & 4/15 \\ 1/4 & 0 & 3/10 & 9/20 \\ 0 & 2/3 & 1/5 & 2/15 \\ 1/5 & 0 & 2/5 & 2/5 \end{bmatrix}$$

1. Find $\text{Prob}(i \rightarrow j \text{ in 10 steps})$.
2. Find the equilibrium probabilities $\pi_0, \pi_1, \pi_2, \pi_3$.

```
In [65]: P=np.array([
[1/3,0,2/5,4/15],
[1/4,0,3/10,9/20],
[0,2/3,1/5,2/15],
[1/5,0,2/5,2/5]
])
print(P)
```

```
[[0.33333333 0.          0.4        0.26666667]
 [0.25      0.          0.3        0.45      ]
 [0.          0.66666667 0.2        0.13333333]
 [0.2      0.          0.4        0.4       ]]
```

```
In [66]: #1):
#find Prob(i -> j in 10 steps)
print(np.linalg.matrix_power(P,10))
```

```
[[0.17004087 0.21052518 0.31578941 0.30364454]
 [0.17004046 0.21052647 0.315789     0.30364406]
 [0.17003993 0.21052791 0.31578988 0.30364227]
 [0.17004086 0.21052518 0.31578941 0.30364455]]
```

```
In [67]: #2):
PT = P.transpose()
eigendata=np.linalg.eig(PT)
X, V=eigendata

u=V[:,0]

usum=u[0]+u[1]+u[2]+u[3]
pi=u/usum
pi = pi.real
print("The Matrix of Equilibrium Probabilities are ", pi)
print("pi_0 = ",pi[0])
print("pi_1 = ",pi[1])
print("pi_2 = ",pi[2])
print("pi_3 = ",pi[3])
```

```
The Matrix of Equilibrium Probabilities are [0.17004049 0.21052632 0.3157894
7 0.30364372]
pi_0 = 0.17004048582995965
pi_1 = 0.21052631578947364
pi_2 = 0.31578947368421045
pi_3 = 0.3036437246963563
```

Q.3

Consider the random walk on 6 sites with the following transition matrix

$$\begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1/4 & 0 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/4 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix}$$

Is the system ergodic? Compare your result to that for the modified random walk with transition matrix below and explain the observed difference in behaviour.

$$\begin{bmatrix} 1/4 & 1/4 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1/4 & 0 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/4 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix}$$

```
In [26]: P=np.array([
[1/2,0,1/2,0,0,0],
[0,1/2,0,1/2,0,0],
[0,0,1/2,0,1/2,0],
[0,1/4,0,1/2,0,1/4],
[1/4,0,1/4,0,1/2,0],
[0,0,0,1/2,0,1/2],
])

PT = P.transpose()
eigendata=np.linalg.eig(PT)
X, V=eigendata

print("X =", X)
print("V =", V)

X = [ 1.0000000e+00+0.j   2.5000000e-01+0.25j   2.5000000e-01-0.25j
1.0000000e+00+0.j   -1.11022302e-16+0.j   5.0000000e-01+0.j ]
V = [[[-3.3333333e-01+0.j      -3.53553391e-01-0.35355339j
-3.53553391e-01+0.35355339j  0.0000000e+00+0.j
0.0000000e+00+0.j          0.0000000e+00+0.j
[ 0.0000000e+00+0.j      0.0000000e+00+0.j
0.0000000e+00-0.j      4.08248290e-01+0.j
4.08248290e-01+0.j      -7.07106781e-01+0.j
[-6.6666667e-01+0.j      -3.53553391e-01+0.35355339j
-3.53553391e-01-0.35355339j  0.0000000e+00+0.j
0.0000000e+00+0.j          0.0000000e+00+0.j
[ 0.0000000e+00+0.j      0.0000000e+00+0.j
0.0000000e+00-0.j      8.16496581e-01+0.j
-8.16496581e-01+0.j      -3.14018492e-16+0.j
[-6.6666667e-01+0.j      7.07106781e-01+0.j
7.07106781e-01-0.j      0.0000000e+00+0.j
0.0000000e+00+0.j          0.0000000e+00+0.j
[ 0.0000000e+00+0.j      0.0000000e+00+0.j
0.0000000e+00-0.j      4.08248290e-01+0.j
4.08248290e-01+0.j      7.07106781e-01+0.j ]]
```

This system is not ergodic since there are two different left eigenvalues of 1 here for two different left eigenvectors. P has only one eigenvalue of modulus 1 if and only if P is ergodic.

```
In [27]: P2 = np.array([
[1/4,1/4,1/2,0,0,0],
[0,1/2,0,1/2,0,0],
[0,0,1/2,0,1/2,0],
[0,1/4,0,1/2,0,1/4],
[1/4,0,1/4,0,1/2,0],
[0,0,0,1/2,0,1/2]
])

PT = P2.transpose()
eigendata=np.linalg.eig(PT)
X, V=eigendata

print("X =", X)
print("V =", V)
```

```
X = [1.0000000e+00+0.j      9.61366523e-01+0.j
5.0000000e-01+0.j      1.44316739e-01+0.2769298j
1.44316739e-01-0.2769298j 9.48221908e-19+0.j      ]
V = [[-4.74473990e-16+0.j      1.75990931e-01+0.j
3.69540676e-16+0.j      -1.78703228e-01-0.46826952j
-1.78703228e-01+0.46826952j 3.21117247e-16+0.j      ]
[ 4.08248290e-01+0.j      -2.25589053e-01+0.j
-7.07106781e-01+0.j      -1.38517226e-01+0.15608984j
-1.38517226e-01-0.15608984j 4.08248290e-01+0.j      ]
[ 7.15998345e-17+0.j      4.62082773e-01+0.j
3.17735908e-16+0.j      -4.22733046e-01+0.32913379j
-4.22733046e-01-0.32913379j -6.36391812e-17+0.j      ]
[ 8.16496581e-01+0.j      -5.92307879e-01+0.j
-1.30807038e-16+0.j      2.02872547e-01+0.09275715j
2.02872547e-01-0.09275715j -8.16496581e-01+0.j      ]
[-4.97688258e-16+0.j      5.00776227e-01+0.j
-3.86808932e-16+0.j      5.94254905e-01+0.j
5.94254905e-01-0.j      2.43113794e-16+0.j      ]
[ 4.08248290e-01+0.j      -3.20953000e-01+0.j
7.07106781e-01+0.j      -5.71739532e-02-0.10971126j
-5.71739532e-02+0.10971126j 4.08248290e-01+0.j      ]]
```

This system is Ergodic since there is only one eigenvalue of modulus 1.P has only one eigenvalue of modulus 1 if and only if P is ergodic. This condition is satisfied here.

Q.4

A queue is observed over 1000 time intervals where the size of the queue after each time step is given. Construct a simple model for this queue as a Markov chain with only nearest neighbour interactions.

1. What is the expected behaviour of the queue as time continues?
2. Is the system ergodic?

In [2]:

```

qdata = [4, 5, 6, 6, 6, 7, 6, 7, 6, 5, 4, 4, 5, 6, 7, 6, 5, 4, 3, 4, 5, 6, 5,
4, 3, 2, 1, 2, 3, 2, 1,
2, 3, 4, 3, 2, 3, 2, 1, 1, 2, 2, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1,
2, 3, 4, 5, 4, 5, 6, 7,
8, 7, 6, 7, 6, 6, 5, 4, 5, 4, 3, 2, 3, 2, 3, 2, 3, 2, 1, 1, 2, 3, 3,
4, 5, 6, 7, 6, 5, 6, 5,
6, 5, 6, 5, 4, 5, 4, 3, 4, 3, 4, 3, 2, 1, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1,
2, 3, 3, 2, 3, 4, 3, 2,
3, 2, 1, 2, 3, 2, 3, 2, 1, 0, 0, 1, 1, 2, 3, 2, 1, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 1, 2, 1, 2,
1, 2, 3, 4, 3, 4, 5, 6, 5, 6, 5, 6, 7, 7, 6, 5, 4, 3, 4, 3, 3, 4, 3,
2, 3, 2, 3, 2, 1, 2, 1,
2, 1, 0, 0, 0, 0, 1, 2, 3, 2, 2, 3, 2, 1, 0, 1, 2, 1, 0, 1, 0, 1, 0, 0,
1, 2, 1, 0, 0, 1, 0, 1,
0, 0, 1, 0, 0, 0, 0, 1, 2, 1, 1, 1, 0, 0, 1, 2, 3, 2, 3, 4,
5, 6, 5, 4, 3, 2, 1, 2,
1, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0,
0, 1, 0, 1, 0, 1, 2,
1, 0, 0, 1, 2, 3, 2, 3, 2, 3, 4, 4, 5, 4, 3, 2, 3, 2, 3, 2, 3, 2,
1, 0, 1, 2, 3, 2, 1, 2,
1, 0, 1, 0, 1, 2, 3, 3, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 3, 2, 2, 1, 0,
1, 2, 1, 2, 3, 4, 3,
2, 2, 1, 2, 3, 4, 5, 4, 4, 5, 4, 3, 4, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 1, 2,
1, 2, 2, 3, 2, 3, 4, 5, 6, 5, 6, 5, 4, 5, 6, 7, 6, 5, 6, 6, 7, 6, 7,
7, 6, 7, 6, 5, 4, 5, 5,
4, 5, 6, 6, 5, 6, 5, 6, 5, 5, 5, 4, 3, 4, 5, 6, 5, 4, 3, 3, 4, 3, 4,
3, 2, 3, 2, 3, 2, 1,
2, 1, 0, 0, 0, 0, 0, 1, 2, 3, 2, 3, 4, 5, 6, 7, 7, 8, 9, 8, 9, 8, 7,
6, 5, 5, 4, 5, 4, 3, 2,
1, 0, 1, 2, 1, 2, 3, 4, 5, 6, 7, 7, 8, 7, 8, 9, 8, 7, 6, 5, 4, 3, 4,
3, 4, 4, 5, 4, 3, 4, 3,
2, 3, 2, 3, 3, 2, 1, 1, 0, 1, 2, 1, 2, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1,
0, 1, 0, 1, 0, 1, 2,
3, 2, 1, 0, 0, 1, 2, 3, 4, 3, 2, 3, 2, 2, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2,
3, 4, 5, 6, 7, 6, 7,
6, 7, 8, 8, 9, 10, 9, 10, 9, 10, 11, 10, 9, 8, 9, 10, 11, 10, 10, 9,
10, 10, 9, 8, 7, 8, 9, 10,
9, 8, 7, 6, 7, 6, 5, 4, 4, 5, 4, 3, 2, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
0, 1, 2, 1, 2, 1, 0, 0,
0, 0, 1, 0, 0, 1, 0, 1, 2, 3, 2, 3, 2, 3, 4, 3, 2, 1, 1, 2, 1, 2, 1, 2,
2, 1, 2, 2, 1, 2, 3, 2,
2, 2, 1, 0, 1, 2, 3, 4, 5, 4, 5, 4, 3, 2, 3, 4, 5, 5, 4, 3, 4, 3, 4,
5, 4, 3, 2, 1, 0, 1, 2,
3, 4, 3, 2, 3, 4, 3, 2, 3, 2, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 3, 4, 5,
4, 5, 4, 5, 6, 5, 4, 5,
6, 5, 4, 3, 2, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 3, 4, 3, 2, 1, 1,
2, 3, 2, 3, 2, 3,
4, 3, 4, 4, 5, 6, 7, 6, 7, 7, 6, 5, 4, 3, 4, 3, 2, 2, 3, 4, 3, 4, 3, 4,
2, 1, 0, 0, 1, 2, 3, 2,
1, 2, 3, 2, 1, 0, 0, 0, 0, 1, 1, 2, 3, 2, 3, 4, 5, 6, 5, 4, 3, 2, 3, 4, 5, 6,
3, 4, 5, 4, 3, 4, 5, 4,
3, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1,
2, 3, 2, 3, 2, 3,
3, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 1, 2, 1, 0, 1, 0, 1, 2,
1, 2, 3, 2, 1, 0,
]
```

```

    0, 0, 1, 2, 2, 3, 3, 2, 3, 4, 3, 2, 1, 2, 3, 3, 2, 1, 0, 1, 2, 2, 3,
4, 5, 6, 7, 7, 6, 5, 4,
    5, 4, 5, 4, 3, 2, 3, 4, 3, 2, 1, 2, 3, 4, 3, 2, 3, 2, 1, 0, 1, 2, 1,
0, 0, 0, 0, 0, 1, 0,
    0, 1, 2, 2, 3, 2, 1, 0, 1, 2, 3, 4, 4, 3, 4, 3, 2, 1, 0, 0, 0, 1,
0, 1, 1, 0, 1, 2, 2, 1,
    0, 0, 1, 0, 1, 2, 2, 3, 3, 4, 3, 2, 3, 2, 1, 0, 0, 1, 2, 1, 2, 3, 2,
1, 2, 3, 2, 3, 2, 1, 0,
    1, 2, 1, 2, 3, 2, 1, 1, 0]

```

In [28]:

```

N=len(qdata) #N counts no of observations
nup = 0 #nup counts number of steps
ndown = 0 #ndown counts no of down steps
n0 = 0 #n0 counts no of times queue is 0

```

```

for i in range(1,N):
    if qdata[i] > qdata[i-1]:
        nup = nup+1
    elif qdata[i] < qdata[i-1]:
        ndown = ndown + 1

#1st if statements complete
if qdata[i-1] == 0:
    n0 = n0+1

#2nd if statement complete
print(N, "observations", nup, "step up,", ndown, "step down", n0, "cannot step
down")
p1 = nup/(N-1)
p2 = ndown/(N-1-n0)
print("prob up = ", p1,", prob down =", p2 )

traffic_density = p1/p2
print("traffic density = ", traffic_density)

```

```

1000 observations 417 step up, 421 step down 170 cannot step down
prob up =  0.4174174174174 , prob down = 0.5078407720144753
traffic density =  0.8219454609003302

```

In [4]:

N

Out[4]:

1000

N is the number of events recorded

We can easily compute the average with the following do loop

In [11]:

```

ans=0
for i in range(1,N):
    ans=ans+qdata[i-1]
#do Loop complete
average=ans/N

```

In [12]: average

Out[12]: 2.718

Apply the simplest model of an infinite nearest neighbour queue with p_1 = prob of jumping up and p_2 = prob of jumping down. These can be estimated by counting the number of jumps up vs the number of jumps down. Consider the following do loop with if statements.

```
In [13]: nup=0 # nup counts no of up steps
ndown=0 # ndown counts no of down steps

for i in range(1,N):
    if qdata[i]>qdata[i-1]:
        nup=nup+1
    elif qdata[i]<qdata[i-1]:
        ndown=ndown+1
# if statements complete
print(nup,"up steps", ndown, "down steps")
```

417 up steps 421 down steps

```
In [14]: p1=nup/N
p2=ndown/N
print(p1,p2)
```

0.417 0.421

Do you consider this to be a good estimate? What is wrong?

No, I do not consider this to be a good estimate since it does not account for when the queue is 0. This gives an inaccurate value of p_2 , since there is a probability of 0 of an empty queue decreasing so the number of zeros must be subtracted from N when finding p_2 .

1) As time continues we would expect the size of the queue to tend towards zero. we expect the queue to die down as the traffic density of 0.82 is less than 1, so there is a greater probability of customers getting serviced than there is a probability of arrivals.

2) The system is Ergodic because it is modelled as a markov chain with only nearest neighbour interactions. Since p_1 and p_2 are neither 0 nor 1, this system is Ergodic. its equilibrium probabilities can be found using the formula:

$$\pi_k = \rho_k(1 - \rho)$$

where ρ is the traffic density

Q.5

A queue is observed over 10000 one-second time intervals with data as given below in the array `qdata`. Construct a Poisson nearest neighbour model with a single arrival and servicing pattern and hence answer the following questions:

1. What is the average time taken for 1 customer to arrive? \item What is the average number of customer serviceings per second?
2. What is your estimate for the equilibrium probability $P(n \geq 4)$, where n is the queue size in this model?
3. Suppose that two equivalent servers are introduced.
4. What would the equilibrium probability $P(n \geq 4)$ then be?

3, 2, 3, 2, 1, 0, 0, 0, 1, 0, 1, 2, 2, 2, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 0, 0, 1, 2, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0,
1, 1, 2, 1, 0, 1, 1, 2, 3, 3, 2, 2, 3, 4, 5, 5, 4, 4, 5, 5, 6, 5, 5, 6, 5, 5,
4, 5, 4, 5, 6, 5, 5, 4, 5, 5, 6, 7, 7, 8, 9, 9, 8, 7, 6, 5, 4, 5, 4, 5, 4,
5, 5, 4, 3, 4, 5, 4, 4, 3, 3, 2, 1, 1, 2, 2, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0,
0, 1, 1, 2, 2, 2, 3, 2, 2, 3, 3, 2, 1, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
1, 2, 3, 4, 5, 4, 3, 3, 4, 3, 2, 2, 1, 0, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 2,
2, 3, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 4, 3, 4, 4, 4, 4, 3,
3, 4, 5, 4, 5, 5, 4, 5, 6, 5, 5, 6, 5, 6, 7, 6, 5, 5, 5, 4, 5, 4, 5, 4, 5, 4,
3, 3, 3, 3, 4, 5, 4, 3, 4, 3, 2, 2, 1, 1, 2, 3, 2, 3, 3, 4, 4, 4, 5, 4, 5,
6, 6, 7, 8, 7, 7, 8, 9, 9, 8, 9, 8, 9, 8, 7, 7, 6, 6, 7, 8, 9, 9, 10, 9, 8,
7, 7, 6, 5, 5, 4, 3, 3, 4, 5, 6, 6, 7, 7, 8, 9, 9, 10, 10, 10, 11, 12, 13, 14,
15, 14, 15, 14, 13, 13, 13, 14, 15, 15, 14, 13, 13, 13, 12, 12, 12, 13, 12, 12
, 13, 12, 11, 12, 11, 10, 10, 9, 8, 8, 9, 9, 10, 11, 11, 10, 10, 9, 9, 9, 9, 1
0, 10, 9, 10, 11, 10, 11, 11, 11, 10, 10, 11, 12, 11, 12, 13, 12, 11, 12, 11,
10, 10, 9, 9, 10, 11, 10, 11, 10, 10, 9, 9, 9, 8, 9, 8, 9, 9, 10, 10, 9, 8, 7,
8, 7, 6, 5, 6, 5, 4, 5, 5, 6, 5, 4, 5, 5, 4, 3, 4, 3, 2, 1, 2, 3, 3, 3, 2, 1,
1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 2, 3, 2, 3, 4, 3, 2, 2, 2, 2, 1, 2,
1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 2, 2, 2, 3, 3, 4, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
1, 2, 3, 3, 3, 4, 4, 3, 3, 2, 3, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2,
1, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0,
0, 0, 1, 0, 0, 0, 1, 2, 3, 3, 3, 3, 2, 3, 4, 3, 2, 2, 3, 3, 2, 2, 2, 2, 1,
2, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1, 2, 3, 2, 2, 1, 2, 1, 0, 0, 0,
1, 2, 1, 0, 0, 1, 2, 2, 2, 1, 0, 0, 0, 0, 1, 2, 3, 2, 3, 2, 3, 4, 3, 4, 4,
4, 4, 5, 5, 4, 5, 4, 5, 6, 7, 8, 8, 7, 6, 7, 7, 8, 8, 7, 6, 7, 7, 6, 6, 7,
7, 8, 9, 8, 7, 6, 7, 7, 8, 8, 7, 8, 9, 10, 9, 10, 11, 12, 11, 11, 12, 11, 10,
10, 10, 10, 11, 10, 11, 12, 12, 12, 12, 12, 11, 12, 11, 11, 10, 9, 9, 8, 8
, 7, 6, 7, 7, 6, 7, 6, 5, 6, 7, 7, 7, 8, 8, 9, 8, 8, 7, 6, 5, 4, 5, 6, 7, 7
, 6, 7, 8, 7, 8, 8, 8, 7, 8, 7, 6, 6, 5, 5, 6, 6, 5, 6, 7, 6, 7, 6, 6, 5, 5, 6
, 5, 6, 6, 7, 7, 6, 7, 7, 8, 9, 9, 8, 7, 8, 7, 6, 7, 8, 9, 10, 11, 10, 9, 9, 9
, 8, 9, 8, 7, 6, 6, 7, 6, 5, 4, 3, 2, 3, 4, 3, 2, 2, 2, 3, 4, 3, 3, 4, 3, 2, 3
, 4, 4, 4, 3, 2, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1
, 2, 3, 2, 3, 3, 2, 3, 3, 2, 2, 3, 2, 3, 2, 1, 2, 3, 2, 2, 1, 2, 2, 3, 2, 2, 1
, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0
, 1, 1, 0, 0, 0, 0, 1, 1, 2, 2, 2, 3, 2, 2, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1
, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1
, 2, 3, 3, 2, 2, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1
, 1, 0, 1, 1, 2, 1, 1, 1, 2, 3, 2, 3, 3, 2, 1, 0, 0, 0, 0, 0, 1, 2, 3, 2, 2, 3
, 2, 1, 0, 1, 1, 2, 3, 2, 2, 3, 3, 3, 2, 1, 0, 0, 0, 0, 1, 2, 3, 2, 2, 3, 2, 1, 2, 1
, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0
, 1, 2, 2, 2, 2, 3, 2, 1, 1, 1, 2, 3, 4, 4, 4, 4, 4, 5, 6, 5, 5, 4, 4, 4, 3, 2, 1
, 1, 2, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 1, 2
, 2, 1, 0, 0, 1, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 2, 2, 1
, 0, 0, 0, 1, 0, 1, 2, 3, 4, 4, 4, 5, 4, 4, 5, 4, 5, 6, 5, 5, 6, 7, 7, 7, 6
, 7, 6, 6, 5, 4, 3, 3, 4, 3, 3, 3, 4, 3, 3, 2, 3, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1
, 2, 1, 1, 2, 2, 1, 2, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 2, 2, 3
, 3, 4, 5, 4, 3, 3, 3, 2, 1, 1, 2, 2, 2, 3, 2, 2, 1, 2, 3, 4, 4, 4, 3, 2, 3
, 2, 3, 4, 4, 3, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2
, 2, 1, 1, 1, 1, 0, 0, 1, 2, 3, 4, 5, 4, 5, 5, 4, 4, 5, 4, 3, 2, 1, 2, 1, 2, 3, 2
, 3, 4, 4, 5, 4, 5, 6, 6, 7, 8, 8, 9, 9, 10, 11, 12, 11, 10, 11, 10, 9, 8, 7,
6, 5, 6, 6, 7, 6, 6, 5, 4, 3, 2, 2, 1, 0, 0, 0, 0, 1, 0, 1, 2, 3, 4, 5, 5, 5, 4,
3, 2, 2, 3, 3, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 3, 2, 2, 2, 3, 3, 3, 3,
4, 3, 2, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,

```

1, 1, 1, 0, 1, 2, 3, 3, 4, 5, 6, 5, 5, 4, 5, 4, 5, 4, 5, 4, 3, 3, 3, 2, 1,
1, 2, 2, 3, 4, 5, 6, 7, 7, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 14, 14, 15
, 16, 16, 15, 15, 16, 17, 18, 17, 16, 15, 15, 14, 15, 14, 13, 14, 15, 14, 13,
13, 12, 12, 13, 12, 13, 13, 13, 12, 13, 14, 13, 13, 13, 13, 13, 12, 11, 11
, 10, 9, 9, 10, 9, 9, 10, 9, 9, 10, 10, 11, 10, 9, 9, 10, 10, 10, 10, 10, 1
0, 9, 8, 8, 7, 7, 8, 7, 8, 7, 7, 8, 7, 7, 6, 5, 4, 3, 4, 5, 5, 4, 3,
2, 1, 1, 1, 2, 3, 2, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1,
2, 2, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 2, 1, 0, 0,
0, 1, 2, 2, 2, 2, 3, 2, 1, 0, 0, 0, 0, 1, 2, 1, 2, 2, 3, 4, 5, 6, 5, 6, 5,
4, 5, 6, 5, 5, 4, 5, 4, 4, 4, 3, 4, 3, 2, 1, 0, 0, 1, 2, 3, 4, 4, 4, 5,
5, 5, 5, 4, 5, 4, 5, 6, 7, 6, 5, 5, 6, 5, 5, 6, 7, 7, 8, 9, 9, 10, 10
, 9, 9, 8, 7, 8, 9, 10, 9, 9, 9, 10, 10, 10, 9, 8, 8, 9, 8, 7, 8, 9, 9, 10,
10, 11, 12, 11, 10, 11, 10, 9, 9, 10, 10, 10, 10, 10, 10, 10, 9, 9, 8, 9, 9, 1
0, 11, 12, 12, 12, 12, 12, 13, 12, 12, 11, 10, 9, 10, 10, 10, 10, 10, 9, 10, 9, 9, 9,
9, 8, 7, 6, 7, 7, 6, 6, 7, 8, 9, 8, 8, 9, 9, 10, 9, 10, 9, 8, 8, 8, 9, 10, 11,
10, 11, 12, 13, 14, 14, 14, 13, 13, 12, 12, 11, 12, 13, 14, 13, 13, 12, 13, 13
, 12, 12, 12, 13, 13, 13, 12, 13, 12, 13, 14, 15, 15, 15, 15, 16, 16, 16, 16,
16, 16, 15, 15, 16, 15, 15, 15, 15, 14, 14, 15, 15, 16, 16, 16, 16, 17, 18, 19, 19,
18, 19, 18, 19, 20, 19, 18, 19, 20, 20, 19, 19, 19, 20, 20, 19, 18, 18, 19, 18, 19, 18
, 17, 16, 15, 15, 14, 14, 14, 14, 14, 13, 12, 12, 13, 13, 14, 15, 15, 15, 15, 15,
16, 17, 18, 18, 17, 18, 17, 17, 17, 16, 15, 14, 14, 15, 14, 13, 14, 14, 15, 16
, 17, 16, 17, 18, 19, 18, 19, 19, 20, 19, 18, 19, 18, 17, 17, 18, 19, 20,
19, 18, 19, 20, 19, 20, 20, 20, 19, 18, 19, 19, 19, 18, 18, 18, 19, 18, 19, 18
, 17, 16, 15, 14, 15, 16, 15, 14, 14, 14, 13, 12, 11, 10, 11, 10, 9, 10, 9, 9,
8, 8, 7, 7, 6, 5, 6, 7, 7, 8, 7, 8, 9, 10, 11, 10, 11, 12, 11, 10, 10, 9
, 8, 7, 8, 8, 7, 7, 8, 8, 7, 8, 7, 7, 6, 5, 4, 3, 4, 3, 2, 3, 2, 3, 2, 1, 0, 0
, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 1, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 2, 3, 4, 3, 2, 2, 2, 2, 3, 4, 5
, 5, 6, 5, 6, 5, 5, 6, 6, 5, 4, 4, 4, 3, 4, 3, 2, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0
, 3, 3, 2, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 3, 4, 3, 2, 1, 2, 2, 1, 2, 2, 3, 4, 3, 4, 3, 2, 1, 1, 0, 1, 0, 1, 2, 3, 4, 4, 3, 4
, 5, 6, 6, 6, 5, 5, 6, 5, 4, 3, 2, 2, 2, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 2, 1, 1, 2, 3, 3, 2, 3, 2, 3, 3, 4, 3, 4, 3, 3
, 2, 2, 1, 0, 1, 2, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 2, 1, 1, 0, 1, 0, 1, 0
, 1, 0, 1, 2, 1, 1, 1, 2, 1, 0, 0, 0, 0, 1, 0, 1, 2, 3, 4, 3, 4, 5, 6, 5, 4, 3, 2, 2
, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 1, 1, 2, 1, 0, 0, 0, 0
, 4, 3, 4, 4, 3, 4, 3, 4, 3, 3, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0
, 1, 0, 1, 1, 1, 1, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 2, 1, 2, 2
, 2, 3, 2, 3, 2, 1, 2, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 1, 2, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 2, 3, 2, 2, 2, 1, 2, 2, 3, 2
, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 6, 6, 7, 6, 5, 5, 5, 5, 6, 5, 6, 5, 6, 5, 6, 5, 6
, 4, 4, 3, 2, 3, 2, 1, 0, 0, 1, 2, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 3, 4, 4, 3, 2, 3, 3, 4, 4, 3, 2, 3, 3, 2, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0
, 1, 1, 2, 2, 1, 2, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 3, 2, 3, 4, 5, 5, 6, 6, 5, 5, 6, 5, 6, 5, 6, 5, 6
, 5, 5, 4, 3, 4, 3, 2, 2, 3, 4, 3, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 2, 1, 2, 3, 2, 2, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 1, 1, 1, 0, 0, 0, 1, 2, 2, 1, 1, 1, 0, 1, 0, 1, 2, 2, 3, 3, 2, 3, 4, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3
, 4, 3, 4, 5, 5, 4, 4, 3, 3, 2, 2, 3, 4, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3
, 2, 2, 3, 4, 5, 5, 6, 7, 7, 6, 6, 5, 4, 3, 4, 4, 4, 4, 5, 4, 3, 2, 1, 2, 3, 4, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3
, 1, 0, 0, 0, 1, 2, 1, 1, 2, 3, 3, 2, 2, 3, 2, 1, 2, 3, 4, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3
, 2, 1, 2, 1, 2, 3, 4, 5, 5, 6, 6, 6, 6, 7, 8, 8, 8, 9, 8, 9, 8, 9, 8, 7, 8, 7, 8, 7, 7, 6, 6, 6, 6, 6, 6, 6, 6
, 6, 7, 6, 6, 5, 4, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 5, 6, 6, 6, 6, 6, 5, 5, 5, 5, 4, 3, 3
, 3, 2, 3, 2, 1, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

```

```

, 0, 1, 0, 1, 1, 0, 0, 0, 1, 2, 2, 2, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1
, 0, 1, 1, 1, 2, 1, 1, 2, 3, 2, 3, 3, 4, 4, 5, 4, 3, 2, 1, 2, 1, 2, 2, 1, 0
, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2, 1, 0, 0, 1, 1, 0, 0
, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 2
, 1, 1, 2, 2, 3, 2, 2, 3, 4, 4, 4, 3, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 1, 1, 0, 1, 0, 1, 2, 1, 1, 0, 0, 0, 0, 1, 0, 1, 2, 2, 1, 1, 0, 1, 0, 0
, 1, 1, 0, 1, 1, 2, 3, 4, 5, 6, 6, 5, 4, 4, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 1, 0
, 1, 0, 0, 0, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 3, 2, 3, 2, 3, 4, 4, 5, 5, 4, 5, 6
, 5, 5, 5, 6, 5, 5, 6, 6, 7, 8, 8, 9, 8, 9, 9, 8, 7, 7, 6, 6, 5, 4, 4, 3, 3
, 2, 2, 1, 1, 0, 0, 1, 0, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 3, 4, 3, 3, 2, 2, 3, 4
, 4, 3, 2, 3, 4, 4, 3, 4, 5, 5, 5, 5, 4, 3, 3, 4, 3, 3, 4, 5, 4, 4, 3, 4, 3, 4
, 3, 4, 5, 5, 4, 5, 6, 7, 6, 5, 4, 5, 6, 5, 4, 4, 4, 4, 5, 4, 3, 4, 3, 2, 1, 0
, 0, 0, 1, 1, 2, 1, 2, 2, 2, 3, 3, 4, 5, 5, 6, 5, 4, 3, 2, 3, 3, 4, 3, 3, 2
, 2, 1, 2, 2, 1, 2, 3, 3, 3, 4, 5, 5, 4, 5, 5, 4, 3, 3, 3, 2, 2, 1, 2, 1, 2, 3
, 4, 4, 4, 5, 4, 4, 4, 3, 3, 4, 4, 4, 3, 4, 4, 5, 6, 5, 5, 6, 7, 7, 7, 6, 6, 5
, 4, 4, 3, 3, 2, 3, 2, 1, 1, 2, 3, 2, 1, 1, 1, 1, 2, 2, 2, 1, 0, 0, 0, 1, 0, 0
, 1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 4, 3, 3, 4, 3, 3, 4, 5, 5, 6, 6, 7, 6, 6, 7
, 7, 6, 5, 6, 7, 8, 8, 9, 9, 10, 9, 8, 7, 7, 7, 6, 5, 4, 4, 5, 4, 4, 3, 3
2, 1, 2, 3, 2, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1
1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 0, 0, 1, 2, 1, 2, 2, 2, 2, 2, 3, 2, 3
4, 3, 4, 5, 4, 4, 5, 6, 6, 7, 7, 6, 7, 6, 6, 5, 5, 6, 6, 7, 6, 5, 5, 4, 5, 5
4, 3, 2, 2, 2, 2, 3, 3, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2
2, 2, 3, 4, 4, 5, 4, 4, 5, 5, 4, 4, 5, 4, 4, 3, 2, 1, 0, 0, 0, 1, 0, 0, 1, 1, 2
2, 2, 3, 2, 3, 3, 2, 1, 2, 3, 3, 3, 2, 2, 2, 2, 3, 4, 3, 3, 4, 5, 6, 7, 6, 6, 6
7, 6, 5, 4, 3, 4, 3, 2, 3, 3, 3, 2, 3, 2, 2, 2, 1, 0, 1, 1, 2, 2, 1, 2, 3, 2
2, 1, 0, 0, 1, 2, 3, 3, 2, 2, 2, 3, 4, 5, 5, 4, 5, 4, 3, 4, 5, 4, 3, 3, 4, 3, 4
3, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 2, 3
3, 3, 2, 2, 3, 4, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0
1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 4, 4, 5, 4, 3, 2, 3, 3
4, 3, 2, 1, 1, 1, 2, 3, 4, 5, 5, 6, 5, 5, 4, 5, 4, 5, 5, 4, 4, 4, 4, 5, 5, 4, 4
3, 3, 4, 5, 5, 4, 4, 4, 3, 2, 3, 3, 3, 2, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 2, 3
2, 1, 0, 1, 1, 2, 3, 4, 3, 4, 4, 3, 2, 1, 0, 0, 0, 1, 0, 1, 0, 1, 2, 1, 2, 2, 3, 2
1, 1, 2, 1, 2, 3, 4, 4, 5, 6, 5, 5, 4, 3, 2, 1, 2, 3, 2, 2, 2, 2, 1, 0, 0, 0, 0
0, 1, 0, 0, 0, 1, 0, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1
2, 1, 1, 2, 3, 2, 2, 1, 1, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 1, 2, 1, 2, 1, 0, 0, 1, 1, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1, 1
2, 3, 4, 3, 3, 2, 3, 2, 3, 3, 2, 1, 0, 0, 0, 0, 0, 0, 1, 2, 1, 2, 3, 2, 3, 3, 4, 4
5, 6, 7, 7, 7, 7, 7, 8, 7, 8, 7, 7, 6, 7, 6, 5, 6, 5, 5, 4, 4, 4, 4, 3, 2, 2, 3, 3
3, 2, 3, 2, 2, 3, 4, 4, 4, 5, 6, 7, 7, 6, 6, 5, 4, 5, 4, 3, 4, 4, 5, 6, 6, 6, 6
7, 7, 7, 6, 5, 6, 6, 7, 7, 8, 7, 8, 8, 9, 10, 10, 10, 9, 9, 9, 8, 7, 7, 6, 7, 7
6, 5, 6, 7, 7, 7, 8, 9, 10, 10, 11, 11, 12, 11, 11, 12, 11, 11, 10, 11, 12, 11
, 10, 10, 11, 10, 10, 9, 8, 8, 7, 8, 9, 10, 9, 9, 8, 9, 8, 9, 9, 10, 10, 11, 12, 11
0, 10, 11, 10, 10, 9, 8, 8, 9, 10, 11, 11, 12, 12, 13, 14, 14, 14, 13, 12, 11, 12
13, 14, 13, 12, 11, 10, 11, 10, 10, 11, 11, 11, 10, 11, 10, 9, 9, 8, 9, 9
10, 11, 12, 13, 12, 11, 12, 13, 13, 12, 12, 12, 11, 12, 11, 10, 9, 9
9, 9, 9, 10, 9, 9, 10, 11, 11, 12, 11, 10, 11, 11, 10, 9, 8, 7, 7
8, 8, 7, 7, 6, 5, 6, 7, 8, 8, 8, 9, 8, 7, 6, 7, 6, 5, 5, 6, 7, 8, 7
6, 5, 5, 6, 7, 6, 7, 7, 8, 7, 6, 5, 4, 3, 2, 1, 2, 3, 4, 3, 2, 3, 4, 3, 2, 2
2, 1, 1, 2, 3, 4, 4, 5, 4, 4, 3, 2, 2, 3, 2, 1, 0, 1, 1, 2, 1, 0, 0, 0, 0, 0
1, 2, 1, 0, 1, 2, 1, 0, 1, 1, 2, 2, 3, 2, 3, 2, 1, 2, 3, 4, 4, 5, 6, 7, 7, 6
6, 7, 6, 7, 8, 8, 8, 9, 8, 8, 9, 9, 10, 9, 9, 8, 9, 9, 9, 10, 9, 9, 10, 9, 10, 9
10, 9, 8, 8, 9, 9, 10, 9, 10, 9, 8, 9, 9, 8, 9, 9, 9, 8, 7, 6, 5, 4, 4, 3
4, 4, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 2
0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 2
3, 4, 4, 5, 4, 4, 4, 3, 4, 5, 5, 4, 4, 3, 2, 3, 2, 3, 2, 2, 1, 0, 0, 0, 1, 0, 0, 0, 1
0, 0, 1, 0, 1, 0, 1, 2, 3, 3, 2, 2, 3, 3, 2, 1, 0, 0, 0, 1, 2, 1, 0, 1, 0, 0, 0, 0

```

0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 2, 2, 1, 2, 3, 3, 3,
2, 3, 2, 3, 3, 4, 3, 3, 3, 3, 4, 4, 5, 4, 4, 3, 3, 3, 3, 2, 3, 3, 3, 4, 3, 3,
4, 5, 4, 5, 6, 5, 6, 6, 6, 6, 7, 6, 7, 6, 5, 5, 5, 4, 4, 5, 5, 4, 3, 3, 4, 4, 4, 3, 3, 4,
4, 5, 4, 3, 3, 2, 1, 1, 1, 2, 3, 4, 5, 4, 4, 4, 3, 3, 4, 4, 4, 3, 3, 4, 3, 3, 4, 3, 4,
5, 4, 4, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
2, 1, 0, 1, 2, 2, 2, 2, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 1, 2, 3, 4, 3, 2, 3, 4, 3, 4, 3, 3, 4, 5, 4, 5, 4, 5, 4, 3, 3, 3,
2, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 3,
4, 5, 4, 4, 3, 4, 4, 3, 4, 5, 6, 6, 6, 5, 5, 4, 3, 3, 3, 4, 4, 4, 3, 2, 1, 0, 1, 0, 1,
2, 1, 0, 1, 2, 1, 1, 0, 0, 0, 1, 0, 0, 1, 2, 1, 2, 1, 1, 0, 0, 1, 2, 1, 1, 0, 0, 0, 0, 0,
0, 1, 0, 0, 1, 1, 2, 3, 3, 3, 2, 1, 2, 1, 2, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1, 0, 0, 1, 2, 2, 3, 4, 4, 3, 3, 3, 4, 4, 5, 6, 5, 4, 3, 3, 3,
3, 3, 2, 2, 1, 1, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 2, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 1, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 1,
0, 1, 2, 3, 4, 5, 6, 7, 6, 7, 8, 7, 6, 7, 8, 9, 8, 9, 10, 9, 8, 7, 6, 7, 7, 6,
7, 6, 5, 4, 3, 3, 2, 3, 4, 3, 3, 3, 3, 4, 3, 4, 5, 4, 5, 4, 3, 2, 3, 3, 2, 2,
3, 4, 4, 3, 2, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 2, 2, 3, 4, 5, 4, 3,
2, 2, 3, 3, 4, 3, 4, 4, 3, 3, 4, 3, 3, 2, 1, 0, 1, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0,
0, 0, 1, 2, 3, 3, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1, 0, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1,
2, 2, 3, 4, 3, 2, 3, 3, 3, 3, 4, 5, 4, 3, 2, 1, 1, 0, 1, 0, 1, 1, 2, 2, 3, 3, 2, 3,
3, 2, 3, 2, 1, 2, 1, 1, 1, 0, 0, 0, 0, 1, 2, 3, 3, 2, 3, 2, 2, 2, 1, 1, 2, 1, 0, 0, 0,
0, 0, 0, 1, 1, 2, 3, 3, 3, 4, 3, 3, 2, 2, 3, 2, 3, 4, 3, 2, 3, 2, 1, 1, 2, 1, 1, 2,
1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 2, 1, 1, 1, 2, 3, 4, 5, 4, 4, 3, 3,
3, 4, 4, 3, 4, 4, 3, 2, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 2, 2, 3, 4, 5, 4, 3,
4, 3, 2, 1, 2, 3, 2, 1, 2, 2, 1, 1, 1, 0, 1, 2, 3, 4, 5, 6, 7, 6, 5, 4,
5, 4, 4, 3, 2, 3, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 2, 2, 1, 0, 0,
0, 1, 1, 0, 0, 1, 2, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 1, 1, 1, 2, 3, 3, 3, 2,
2, 2, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 2, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 1, 2, 2, 2, 1, 2, 3, 3, 2, 3, 2, 3, 2, 1, 0, 1, 0, 1, 2, 3, 2, 2, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
1, 2, 3, 3, 4, 3, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 3, 2, 3, 2, 3, 3, 4, 5, 6,
6, 5, 6, 7, 8, 8, 7, 6, 5, 4, 3, 3, 2, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 0, 1,
0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 2, 1, 1, 0, 0, 0, 1, 1, 2, 1, 2, 1, 0, 0, 1, 2, 1, 0,
0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 2, 1, 1, 0, 1, 2, 1, 0, 0, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 0, 1, 2, 1, 0, 0, 0, 0,
1, 0, 0, 1, 2, 1, 1, 0, 0, 0, 1, 1, 2, 3, 4, 5, 5, 4, 4, 4, 3, 3, 2,
2, 1, 1, 0, 0, 1, 0, 1, 1, 2,
3, 2, 3, 3, 2, 1, 2, 1, 0, 1, 0, 0, 0, 0, 1, 1, 2, 1, 2, 3, 3, 2, 2, 3, 2, 2,
1, 2, 3, 4, 5, 5, 6, 5, 4, 3, 3, 4, 3, 4, 4, 4, 4, 5, 6, 5, 5, 5, 4, 4, 3,
4, 3, 2, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1,
0, 1, 0, 0, 1, 2, 2, 1, 2, 3, 4, 3, 3, 4, 5, 4, 3, 2, 2, 3, 4, 3, 4, 3, 2, 2,
3, 2, 2, 3, 2, 3, 4, 5, 4, 5, 6, 5, 4, 3, 2, 1, 1, 2, 2, 3, 3, 3, 2, 2, 1, 0,
1, 0, 0, 0, 1, 1, 2, 2, 1, 0, 1, 2, 1, 0, 0, 1, 2, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0,
1, 1, 2, 3, 3, 4, 3, 3, 3, 3, 3, 4, 3, 4, 5, 5, 4, 4, 3, 3, 4, 3, 2, 2, 2,
3, 2, 1, 2, 2, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
2, 2, 3, 2, 1, 2, 2, 2, 3, 2, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 3, 2, 1,
1, 1, 2, 2, 1, 2, 1, 0, 0, 0, 1, 1, 2, 3, 4, 3, 2, 2, 1, 1, 2, 2, 3, 2, 2, 1,
1, 0,
0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1,
1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 2, 1, 1, 0, 0, 0, 1, 2, 1, 2, 1, 0, 0, 0, 0, 1,

```

4, 3, 4, 5, 4, 5, 4, 3, 4, 3, 2, 1, 2, 3, 4, 5, 4, 4, 4, 4, 4, 4, 4, 3, 4, 5, 4, 5,
4, 4, 4, 5, 5, 4, 5, 6, 7, 8, 7, 6, 7, 6, 6, 5, 4, 5, 4, 4, 5, 4, 4, 4, 4, 4, 4, 4,
4, 5, 5, 4, 4, 5, 6, 5, 4, 3, 4, 4, 3, 4, 3, 4, 5, 5, 4, 5, 6, 7, 8, 8, 9, 8,
9, 9, 10, 9, 9, 10, 10, 10, 9, 9, 10, 11, 10, 9, 8, 7, 6, 5, 5, 4, 4, 3, 2, 2,
1, 2, 1, 2, 3, 4, 3, 4, 3, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 0, 0, 0, 1, 0, 0,
1, 2, 2, 1, 1, 2, 3, 2, 3, 3, 3, 4, 3, 4, 5, 5, 5, 4, 3, 2, 3, 2, 3, 2, 3, 2,
3, 4, 5, 5, 5, 4, 5, 6, 5, 5, 4, 3, 3, 3, 2, 1, 2, 1, 1, 0, 1, 1, 1, 0, 1,
2, 1, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 3, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 2, 1, 2, 3, 2,
3, 4, 3, 4, 3, 5, 5, 6, 6, 5, 6, 6, 5, 6, 5, 5, 4, 5, 6, 7, 8, 7, 7, 8,
7, 6, 6, 5, 6, 7, 8, 8, 7, 6, 5, 4, 5, 5, 4, 4, 4, 3, 3, 2, 1, 2, 3, 3, 2, 2,
1, 1, 2, 2, 1, 1, 2, 1, 2, 3, 4, 4, 3, 2, 3, 3, 2, 3, 2, 3, 2, 2, 1, 2, 1,
1, 0, 1, 2, 2, 1, 1, 2, 1, 2, 3, 2, 1, 2, 2, 2, 3, 4, 3, 2, 1, 2, 1, 0, 0, 0,
0, 0, 0, 1, 2, 3, 3, 2, 3, 3, 4, 5, 5, 5, 6, 5, 4, 3, 2, 2, 1, 1, 2, 3, 2,
1, 0, 0, 0, 1, 1, 2, 2, 3, 4, 5, 6, 7, 8, 7, 8, 7, 7, 8, 7, 8, 9, 9, 8, 8,
8, 8, 8, 8, 7, 7, 7, 6, 6, 7, 8, 9, 10, 9, 8, 9, 8, 7, 6, 7, 8, 7, 6, 6, 7, 8,
9, 10, 10, 9, 10, 10, 9, 9, 8, 7, 7, 6, 7, 6, 5, 6, 5, 4, 5, 5, 6,
, 7, 8, 8, 7, 7, 8, 8, 7, 6, 6, 5, 5, 4, 3, 3, 4, 5, 5, 4, 4, 3, 2, 3, 4, 3, 4
, 5, 4, 5, 5, 6, 5, 6, 5, 4, 4, 5, 4, 3, 2, 2, 2, 3, 3, 2, 2, 3, 3, 2, 1, 1
, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 2, 3, 2, 3, 3, 4, 4, 3, 4, 3, 4, 5, 5
, 6, 5, 5, 6, 7, 8, 7, 7, 6, 7, 8, 7, 6, 6, 7, 6, 7, 7, 7, 8, 9, 8, 9, 8, 7
, 8, 7, 6, 6, 7, 6, 6, 5, 6, 7, 7, 6, 7, 7, 8, 9, 9, 9, 8, 8, 7, 8, 9, 10,
11, 12, 12, 11, 11, 11, 12, 11, 11, 11, 11, 12, 13, 13, 14, 13, 14, 15, 14, 13
, 12, 13, 12, 12, 13, 13, 14, 13, 13, 12, 13, 13, 14, 14, 14, 15, 16, 15, 16,
17, 17, 17, 18, 18, 19, 18, 17, 17, 18, 17, 16, 16, 17, 16, 16, 17, 16, 15, 15
, 16, 15, 16, 15, 14, 13, 14, 13, 14, 15, 14, 15, 14, 15, 14, 15, 15, 16, 15, 16, 17,
16, 16, 16, 17, 17, 16, 17, 18, 17, 17, 16, 16, 17, 16, 15, 15, 14, 13, 13
, 12, 11, 11, 12, 11, 11, 12, 12, 11, 11, 11, 12, 13, 14, 13, 13, 14, 13,
14, 13, 14, 15, 16, 16, 16, 17, 16, 15, 14, 15, 14, 14, 13, 14, 14, 14, 13
, 14, 14, 14, 15, 16, 16, 15, 16, 17, 18, 17, 18, 17, 18, 18, 18, 19, 19, 20, 19,
20, 19, 19, 20, 19, 18, 19, 20, 19, 18, 17, 16, 16, 15, 16, 17, 16, 15, 16, 15
, 16, 15, 14, 15, 14, 13, 14, 15, 14, 15, 14, 15, 14, 15, 14, 15, 16, 15, 16, 17,
15, 15, 14, 14, 13, 12, 13, 13, 12, 11, 11, 11, 10, 11, 11, 11, 11, 11, 11, 12
, 13, 14, 15, 14, 13, 12, 12, 11, 10, 10, 9, 10, 11, 10, 10, 11, 10, 11, 10, 11, 1
0, 9, 8, 8, 8, 7, 7, 6, 6, 5, 5, 4, 3, 2, 2, 3, 2, 3, 4, 3, 2, 1, 2, 2, 1,
0, 0, 0, 1, 2, 2, 3, 2, 3, 3, 4, 5, 5, 5, 4, 3, 2, 1, 2, 1, 1, 0, 0, 1, 0, 0, 0,
0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 2, 3, 4, 3, 3, 3, 4, 5, 5, 4, 3,
2, 2, 3, 2, 1, 1, 2, 1, 0, 0, 1, 2, 2, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 1, 2, 2, 1, 1, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 3, 4, 3, 3, 3, 4, 5, 5, 4, 3,
3, 3, 4, 5, 5, 5, 6, 7, 6, 6, 5, 6, 6, 7, 8, 9, 9, 10, 9, 8, 7, 7, 8, 9, 10,
9, 10, 11, 11, 10, 11, 10, 11, 12, 13, 14, 13, 14, 13, 12, 11, 10, 11, 10, 11, 10,
9, 8, 9, 8, 7, 8, 7, 6, 7, 8, 7, 6, 6, 5, 4, 3, 2, 3, 4, 3, 2, 3, 4, 5, 4, 3, 2,
4, 3, 3, 2, 3, 2, 2, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
3, 2, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2, 1, 2, 3, 2, 2, 1, 0, 0, 0, 0, 1, 0,
1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
1, 2, 2, 1, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
5, 6, 5, 5, 6, 6, 6, 7, 6, 6, 5, 7, 8, 7, 8, 7, 8, 9, 10, 9, 8, 7, 7, 8, 7, 8, 9,
9, 8, 8, 7, 8, 9, 9, 8, 8, 7, 6, 5, 5, 5, 6, 7, 6, 7, 6, 7, 8, 8, 9, 9, 10, 9, 8, 7,
8, 9, 9, 10, 10, 11, 10, 9, 8, 8, 8, 8, 9, 8, 8, 7, 6, 5, 5, 5, 6, 7, 6, 7, 6, 7, 8,
8, 9, 8, 7, 7, 7, 6, 6, 5, 4, 3, 2, 3, 4, 3, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 4, 3, 2,
6, 5, 4, 4, 4, 3, 4, 3, 2, 1, 2, 1, 1, 2, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,
3, 3, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2, 1, 2, 3, 2, 2, 1, 0, 0, 0, 0, 1, 0,
1, 2, 2, 2, 3, 2, 1, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
5, 6, 5, 6, 5, 5, 6, 7, 8, 7, 8, 9, 8, 7, 6, 6, 7, 8, 7, 8, 6, 6, 5, 4, 3, 2, 3, 4, 3, 2,
4, 3, 2, 2, 1, 1, 2, 1, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
1, 2, 2, 2, 1, 1, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 2, 2, 3, 2, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
0, 1, 1, 2, 3, 4, 3, 3, 2, 1, 2, 3, 2, 2, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
5, 5, 5, 4, 3, 2, 1, 0, 0, 0, 0, 0, 1, 2, 1, 2, 3, 1, 2, 3, 2, 3, 1, 2, 3, 2, 3, 1, 2, 3, 2, 3, 1, 2, 3

```

```
2, 3, 2, 1, 0, 1, 1, 1, 0, 1, 1, 2, 3, 3, 3, 4, 3, 2, 2, 2, 1, 0, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1, 2, 2, 1, 1, 0, 0, 0, 0, 1, 2, 2,
2, 1, 0, 0, 0, 0, 1, 2, 1, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 1, 2, 1, 2, 3, 4,
4, 3, 2, 3, 4, 5, 4, 3, 4, 5, 6, 7, 7, 8, 7, 8, 9, 8, 9, 8, 9, 9, 9, 10, 9, 9,
9, 10, 11, 11, 10, 9, 9, 8, 7, 8, 8, 7, 8, 7, 6, 7, 6, 7, 7, 7, 8, 8, 7, 7, 8, 9,
10, 9, 10, 9, 9, 8, 7, 7, 6, 5, 5, 4, 3, 3, 4, 5, 4, 3, 3, 4, 4, 5, 5, 5, 4, 5
, 6, 6, 5, 6, 6, 7, 8, 8, 9, 8, 7, 8, 7, 8, 8, 9, 10, 9, 8, 8, 7, 6, 5, 6, 6,
7, 8, 7, 7, 6, 5, 4, 3, 2, 1, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 1,
0, 0, 0, 0, 1, 1, 1, 2, 3, 3, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 1, 0, 1, 0,
2, 2, 2, 3, 2, 2, 3, 2, 1, 2, 3, 3, 4, 4, 4, 4, 3, 2, 3, 2, 2, 2, 3, 3,
3, 3, 2, 3, 2, 2, 1, 1, 0, 0, 0, 0, 0, 1, 2, 1, 2, 3, 2, 2, 1, 2, 1, 2, 1, 1, 0,
0, 1, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 1, 2, 1, 1, 1, 2, 3, 4, 4, 5, 5, 6,
5, 4, 4, 4, 3, 3, 3, 2, 1, 1, 0, 0, 0, 1, 0, 1, 2, 3, 2, 3, 4, 5, 6, 5, 5, 4,
3, 4, 5, 5, 4, 5, 6, 6, 5, 4, 4, 3, 3, 3, 4, 3, 3, 3, 3, 4, 3, 3, 4,
3, 2, 2, 3, 3, 4, 3, 2, 2, 2, 3, 3, 2, 1, 2, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1, 0,
0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 2, 3, 4, 3, 3, 3, 2, 2, 1, 1, 2,
2, 1, 0, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 2, 1, 0,
1, 0, 1, 0, 0, 0, 0, 1, 1, 2, 2, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 3, 4, 3, 2,
1, 1, 0, 1, 2, 3, 2, 1, 0, 0, 0, 1, 2, 2, 3, 3, 2, 1, 0, 1, 2, 2, 3, 3, 2,
2, 2, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 2, 2, 2, 3, 3, 3, 3, 4, 3, 3, 3, 4, 5, 4, 5, 4, 3, 2, 2, 1,
0, 0, 1, 2, 1, 0, 0, 1, 2, 1, 2, 3, 2, 3, 3, 2, 1, 2, 3, 3, 4, 3, 2, 3,
1, 1, 1, 1, 2, 3, 2, 2, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 2, 2, 1, 2, 2,
1, 0, 0, 0, 1, 2, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
3, 2, 2, 1, 1, 2, 2, 2, 3, 4, 5, 4, 4, 5, 5, 6, 5, 4, 4, 3, 3, 3, 4, 3, 3, 4,
5, 6, 7, 7, 7, 7, 7, 6, 7, 8, 7, 8, 7, 8, 9, 9, 9, 8, 8, 8, 8, 7, 7, 8, 7, 8, 8,
8, 7, 8, 7, 6, 5, 4, 4, 5, 4, 5, 6, 5, 4, 4, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 4,
4, 5, 4, 4, 3, 4, 3, 4, 3, 3, 4, 3, 3, 4, 3, 3, 4, 3, 3, 4, 3, 3, 4, 3, 3, 4,
3, 2, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
3, 4, 5, 6, 5, 6, 6, 6, 7, 6, 6, 6, 7, 6, 5, 5, 6, 5, 6, 5, 6, 5, 4, 3, 2, 3,
5, 4, 4, 3, 4, 3, 2, 3, 4, 3, 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
6, 5, 4, 5, 6, 7, 6, 5, 4, 3, 3, 3, 4, 3, 4, 4, 4, 4, 4, 4, 3, 2, 1, 0, 0, 0,
4, 3, 3, 4, 5, 6, 7, 8, 8, 9, 9, 8, 9, 8, 8, 7, 6, 5, 5, 5, 5, 5, 4, 4, 3, 2,
1, 2, 1, 0, 1, 1, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 1, 0, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 5, 6, 5, 6, 7, 8, 8, 9, 9, 8, 9, 8, 9, 8, 7, 6, 5, 5, 5, 5, 4, 4, 3, 2,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
In [66]: N=len(qdata)

nup = 0 #nup counts number of steps
ndown = 0 #ndown counts no of down steps
n0 = 0 #n0 counts no of times queue is 0

for i in range(1,N):
    if qdata[i] > qdata[i-1]:
        nup = nup+1
    elif qdata[i] < qdata[i-1]:
        ndown = ndown + 1

#1st if statements complete
if qdata[i-1] == 0:
    n0 = n0+1

#2nd if statement complete
print(N, "observations", nup, "step up,", ndown, "step down", n0, "cannot step down")
p1 = nup/(N-1)
p2 = ndown/(N-1-n0)
print("prob up = ", p1, ", prob down =", p2 )

traffic_density = p1/p2
print("traffic density = ", traffic_density)

#note in poisson model, p1 = alpha * delta_t
#take delta t = 1 second
#alpha = p1 sec^{-1}

alpha = p1
beta = p2
```

10001 observations 3068 step up, 3067 step down 2178 cannot step down
 prob up = 0.3068 , prob down = 0.39209920736384557
 traffic density = 0.7824550374959244

```
In [67]: print("1")
print("average time taken for customer to arrive = {0:0.3f} seconds".format(1/
alpha)) #1/alpha is the average time taken for 1 customer to arrive
print("average number of customers served per second = {0:0.3f} customers".for-
mat(1*beta))
```

1)
 average time taken for customer to arrive = 3.259 seconds
 average number of customers served per second = 0.392 customers

```
In [68]: print("2")
n = 4
pi_4_1server = traffic_density**n #prob(k>=n) = 1 - prob(k<=n) = rho**n
print("My estimate of the equilibrium probability P(n>=4) for 1 server = {0:.3f}".format(pi_4_1server))
```

2)
My estimate of the equilibrium probability P(n>=4) for 1 server = 0.375

3) If two equivalent servers are introduced then β triples

```
In [69]: print("4")

#Beta doubled when 2 servers introduced
p2 = 3*beta
traffic_density2 = p1/p2
pi_4_3server = traffic_density2**n

#P(n>=4) becomes

print("My estimate of the equilibrium probability P(n>=4) for 3 servers = {0:.0.5f}".format(pi_4_3server))
```

4)
My estimate of the equilibrium probability P(n>=4) for 3 servers = 0.00463

In []: