In [1]:

```
import numpy as np
from statistics import mode
```

In [6]:

```
c = 3
p = 12
t = list(range(0,11))
optimal_hour_list = []
for i in t:
    profit_list = []
    for j in range(100000):
        D = np.random.poisson(float(np.log(1+t[i])))
        profit = D*p - c*t[i]
        profit_list.append(profit)
    expected_value = np.mean(profit_list)
    optimal_hour_list.append(expected_value)
    print('working ' + str(i) + ' hours per day gives expected net income = ' + str(expected_value))

print('The optimal working hour that maximize expected net income is ' + str(optimal_hour_list.index(max(optimal_hour_list))))
```

```
working 0 hours per day gives expected net income = 0.0
working 1 hours per day gives expected net income = 5.28948
working 2 hours per day gives expected net income = 7.21548
working 3 hours per day gives expected net income = 7.58832
working 4 hours per day gives expected net income = 7.32012
working 5 hours per day gives expected net income = 6.52416
working 6 hours per day gives expected net income = 5.39772
working 7 hours per day gives expected net income = 3.89916
working 8 hours per day gives expected net income = 2.25108
working 9 hours per day gives expected net income = 0.66216
working 10 hours per day gives expected net income = -1.22124
The optimal working hour that maximize expected net income is 3
```

In [ ]: