それぞれ設問 1~4 に question01~04 が対応している．

Question01:

python question.py で実行することが出来る．

server.log を読み込み，故障状態のサーバアドレスとサーバの故障期間が出力される．連続したタイムアウトには対応していないため，タイムアウトした後に正しく ping が得られた場合のみ故障状態のアドレスとサーバの故障期間が出力される．

server.log を読み込んだ実行結果は以下の通り

```
ログファイルの行数:17
故障状態のサーバアドレス:10.20.30.1/16
サーバの故障期間 0:00:10
故障状態のサーバアドレス:10.20.30.2/16
サーバの故障期間 374 days, 23:59:00
```

question2:

python question2.py で実行することが出来る．

input 関数を使用し, ユーザーから N の入力を受け取ったのちに故障状態とサーバの故障期間を出力する．1 と入力すると question1 と同じ動作を行う．以下に入力として 1,3,5 を与えた実行結果を示す．なお，server.log は quesiton1 で用いたものと同じものである．

```
ログファイルの行数:17
何回以上連続してタイムアウトした場合にのみ故障とみなすように設定しますか?>>
1
故障状態のサーバアドレス:10.20.30.1/16
サーバの故障期間 0:00:10
故障状態のサーバアドレス:10.20.30.2/16
サーバの故障期間 374 days, 23:59:00

ログファイルの行数:17
何回以上連続してタイムアウトした場合にのみ故障とみなすように設定しますか?>>
3
サーバアドレス 10.20.30.1/16 はタイムアウトしましたが, 連続タイムアウト回数が 1 であったため，故障とみなされませんでした．
故障状態のサーバアドレス:10.20.30.2/16
サーバの故障期間 374 days, 23:59:00

ログファイルの行数:17
何回以上連続してタイムアウトした場合にのみ故障とみなすように設定しますか?>>
```

```
5
サーバアドレス 10.20.30.1/16 はタイムアウトしましたが, 連続タイムアウト回数が 1 であったため，故障とみなされませんでした.
サーバアドレス 10.20.30.2/16 はタイムアウトしましたが, 連続タイムアウト回数が 3 であったため，故障とみなされませんでした.
```

question3

python question3.py で実行することが出来る.

input 関数を使用し, ユーザーから m と t の入力を受け取ったのちにサーバの過負荷状態を出力するプログラムとなっている．以下に入力として(m,t) = (2,1)，(m,t) = (3,100)を入力した実行結果を示す．server.log は question1 で使用したものと同じである.

```
ログファイルの行数:17
直近 m 回の平均応答時間が t ミリ秒を超えた場合は、サーバが過負荷状態になっているとみなします．m と t の入力を行て
ください
m>>
2
t>>
1
key: 10.20.30.2/16, value: {1: {'time': '20211029134122', 'ping': '3'}, 2: {'time': '20201019133325', 'ping': '2'}}
key: 10.20.30.2/16, total_ping: 5,ave_ping:2.5
サーバー10.20.30.2/16 は過負荷状態であり，期間は 375 days, 0:07:57
key: 10.20.30.1/16, value: {1: {'time': '20201019133334', 'ping': '1'}, 2: {'time': '20201019133224', 'ping': '522'}}
key: 10.20.30.1/16, total_ping: 523,ave_ping:261.5
サーバー10.20.30.1/16 は過負荷状態であり，期間は 0:01:10
key: 192.168.1.2/24, value: {1: {'time': '20201019133235', 'ping': '15'}, 2: {'time': '20201019133135', 'ping': '5'}}
key: 192.168.1.2/24, total_ping: 20,ave_ping:10.0
サーバー192.168.1.2/24 は過負荷状態であり，期間は 0:01:00
key: 192.168.1.1/24, value: {1: {'time': '20201019133234', 'ping': '8'}, 2: {'time': '20201019133134', 'ping': '10'}}
key: 192.168.1.1/24, total_ping: 18,ave_ping:9.0
サーバー192.168.1.1/24 は過負荷状態であり，期間は 0:01:00
```

```
ログファイルの行数:17
m>>
3
t>>
100
key: 10.20.30.2/16, value: {1: {'time': '20211029134122', 'ping': '3'}, 2: {'time': '20201019133325', 'ping': '2'}, 3: {'time': '20201019133225', 'ping': '1'}}
key: 10.20.30.2/16, total_ping: 6,ave_ping:2.0
サーバー10.20.30.2/16 は過負荷状態ではありません.
key: 10.20.30.1/16, value: {1: {'time': '20201019133334', 'ping': '1'}, 2: {'time': '20201019133224', 'ping': '522'}, 3: {'time': '20201019133124', 'ping': '2'}}
key: 10.20.30.1/16, total_ping: 525,ave_ping:175.0
サーバー10.20.30.1/16 は過負荷状態であり，期間は 0:02:10
key: 192.168.1.2/24, value: {1: {'time': '20201019133235', 'ping': '15'}, 2: {'time': '20201019133135', 'ping': '5'}}
key: 192.168.1.2/24, total_ping: 20,ave_ping:10.0
サーバー192.168.1.2/24 は過負荷状態ではありません.
key: 192.168.1.1/24, value: {1: {'time': '20201019133234', 'ping': '8'}, 2: {'time': '20201019133134', 'ping': '10'}}
key: 192.168.1.1/24, total_ping: 18,ave_ping:9.0
サーバー192.168.1.1/24 は過負荷状態ではありません.
PS C:¥Users¥81905¥Pylesson> python -u "c:¥Users¥81905¥Pylesson¥question03.py"
ログファイルの行数:17
key: 10.20.30.2/16, value: {1: {'time': '20211029134122', 'ping': '3'}, 2: {'time': '20201019133325', 'ping': '2'}}
key: 10.20.30.2/16, total_ping: 5,ave_ping:2.5
': '2'}}
サーバー10.20.30.2/16 は過負荷状態であり，期間は 375 days, 0:07:57
key: 10.20.30.1/16, value: {1: {'time': '20201019133334', 'ping': '1'}, 2: {'time': '20201019133224', 'ping': '522'}}
': '522'}
key: 10.20.30.1/16, total_ping: 523,ave_ping:261.5
サーバー10.20.30.1/16 は過負荷状態であり，期間は 0:01:10
key: 192.168.1.2/24, value: {1: {'time': '20201019133235', 'ping': '15'}, 2: {'time': '20201019133135', 'pi }ng': '5'}}
ng': '5'}
```

key: 192.168.1.2/24, total_ping: 20,ave_ping:10.0
サーバー 192.168.1.2/24 は過負荷状態であり，期間は 0:01:00 }
key: 192.168.1.1/24, value: {1: {'time': '20201019133234', 'ping': '8'}, 2: {'time': '20201019133134', 'ping': '10'}}
g': '10'}
key: 192.168.1.1/24, total_ping: 18,ave_ping:9.0
}
サーバー192.168.1.1/24 は過負荷状態であり，期間は 0:01:00

question04:
最終的に完成には至らなかったため，説明を補足しておく．
まず，サブネットマスクとネットワークプレフィックス長についてであるが，私のプログラムでは makeipgroup()内の ipgroup において，区別される．
ipgroup は
ipgroup = {
　　ネットワークプレフィックス長:{
　　　　サブネットマスク:{
　　　　　　サーバーアドレス:{
　　　　　　　　様々な情報（ping がハイフンの時の time 等）
　　　　　　}
　　　　}
　　}
}
というような構成の連想配列になっている．ネットワークプレフィックス長が異なるとそもそもサブネットマスクも異なるため，ネットワークプレフィックス長が共通しており，かつサブネットマスクも同じ場合に同じサブネットマスクであるという条件式にすることを考えた．具体的な数値を与えると
16:{'10.20': {'10.20.30.2/16': {'errorstarttime': '20201019133125', 'counttimeout': 1, 'errorendtime': '20201020133325'}, '10.20.30.1/16': {'errorstarttime': '20201019133324', 'counttimeout': 2, 'errorendtime': '20201020133224'}}}

というような風になっている．ここで，10.20 内のすべての要素について errorstarttime が存在すること，errorendtime が存在すること，counttimeout が N 以上であることを満たすことが出来ればそのサブネットの故障とみなせると考えた．また，その際に故障期間を求め

る必要があるが，これは 10.20 内の各要素の errorstarttime のうちもっとも小さいものを errorendtime の最も大きいものから引くことで求められる．
しかし，この部分でうまく実装することが出来なかった．

今回のプログラミング試験では応用性に力を入れ，連想配列を用いることで応用性を持たせつつ直感的にわかりやすいプログラムになるように心がけた．だが，for 文を繰り返すことによる計算量の増加に対しての問題意識が低くなってしまった．