Pythonシステム上で本シミュレーションを実施しているターミナル画面からコピーしたテキスト。

```
Python 2.7.5 (default, Oct  2 2013, 22:34:09)
[GCC 4.8.1] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from LLib import *
>>> import Wcenter
>>> Mayor=Cities["〇〇県△△市"].roles["市長"]
>>> Manager=Wcenter.Center.roles["指定管理者"]
>>> User=Wcenter.Center.roles["利用者"]
>>> X=corporation("(株)鈴木販売")
>>> apply(X,Mayor,Manager,None,[])
True
>>> assign(Mayor,X,Manager)
True
>>> if X in Manager.member : print "OK"
... else: print "NG"
...
OK
>>> Y=person("山田太郎")
>>> Y.age=35
>>> apply(Y,Manager,User,None,[])
True
>>> permit(Manager,Y,User)
True
>>> if Y in User.member : print "OK"
... else: print "NG"
...
OK
```

Pythonシステムが立ち上がったことを示すメッセージ。

※以下、「>>>」の行の部分が入力しているコマンド部分。
シミュレータの基本システムをロード。
本件「△△市総合福祉センター条例」のスクリプトをロード。
変数Mayorに当該市長の機関エージェントを代入。
※「機関エージェント」とは、その期間を担う個人ではなくては、機関そのものをシミュレータ上で実体化させたもの。
変数Managerに当該管理者の機関エージェントを代入。
変数Userに当該施設利用者(という役割)自体を代入。
※これも機関と同じで、制度上の登場者(役割)であり、個別の個人ではなく、概念的なもので、その役割を担う個人は、通常は複数存在し、どんどん入れ替わる。
指定管理者への希望者のシステム上の実体を作成し、Xに代入。
その希望者が市長に申請。第4,5引数は本来は申請者が添付する書類や情報を記載するが、本デモでは省略。
申請操作は有効。
市長がその希望者を管理者として割り当てた。
その割り当て操作自体も有効。
現在、その管理者として決定しているメンバーにたしかに本希望者が含まれているか確認する条件文。
本希望者が管理者となっていることが確認された。
施設利用希望者のシステム上の実体を作成し、Xに代入。
例えば、年齢を35歳、のように設定できる。
この利用希望者が指定管理者に利用申請。
申請自体は有効。
指定管理者がその希望者に利用許可を実施。
その許可は有効。
現在、その利用許可取得者として決定しているメンバーにたしかにその希望者が含まれているか確認する条件文。
本利用希望者が許可されていることが確認された。

```
>>> len(Y.right)
1
>>> Right=Y.right[0]
>>> print Right.name
利用権
>>> Z=person("鈴木花子")
>>> Right.transfer(Y,Z)
False
>>> len(Y.right)
1
>>> len(Z.right)
0
>>> Right.execute(Y)
True
>>> len(Y.right)
0
>>>
```

その利用者が実際に持っている権利の数を表示。
1個だけである。
その1つだけの権利を変数Rightに代入。
その権利の名称を表示させる。
その名称は確かに「利用権」となっている。

第三者を登場させるために、システム上の実体を作成し、変数Zに代入。
変すRightに代入された権利を第三者Zに譲渡しようとする。
失敗に終わる。
※なぜなら、条例で禁じられている操作だからである。

当該利用者の権利の数を表示。
まだ当該権利が1つ入ったままである。

第三者Zは何の権利数を表示させる。
権利譲渡がなかったので、0のままである。

当該権利を行使する。(実際に利用することに相当)
有効に行使された。

この段階で当該利用者の権利の数を表示。
もう利用してしまったので、権利は残っていない。