

% Input Facts

% Define legacy configurations for people, rooms, cabinets, and things

legacyConfig(person(1)).

legacyConfig(person(2)).

% Rooms and cabinets from legacy configurations

legacyConfig(room(3)).

legacyConfig(room(4)).

legacyConfig(roomTOcabinet(3,5)).

legacyConfig(roomTOcabinet(3,6)).

legacyConfig(roomTOcabinet(3,7)).

legacyConfig(roomTOcabinet(4,8)).

legacyConfig(roomTOcabinet(4,9)).

legacyConfig(roomTOcabinet(4,10)).

legacyConfig(cabinet(5)).

legacyConfig(cabinet(6)).

legacyConfig(cabinet(7)).

legacyConfig(cabinet(8)).

legacyConfig(cabinet(9)).

legacyConfig(cabinet(10)).

legacyConfig(cabinetTOthing(5,11..15)).

legacyConfig(cabinetTOthing(6,16..19)).

legacyConfig(cabinetTOthing(7,20..22)).

legacyConfig(cabinetTOthing(8,23..27)).

legacyConfig(cabinetTOthing(9,28..31)).

legacyConfig(cabinetTOthing(10,32..34)).

legacyConfig(thing(11..34)).

thingShort(11;12;13;16;17;20;23;24;25;28;29;32).

```
thingLong(14;15;18;19;21;22;26;27;30;31;33;34).
```

```
legacyConfig(personTOthing(1,11..22)).
```

```
legacyConfig(personTOthing(2,23..34)).
```

```
% Explicitly define cabinets and rooms
```

```
cabinet(C) :- legacyConfig(cabinet(C)).
```

```
room(R) :- legacyConfig(room(R)).
```

```
% Define new domains for rooms and cabinets
```

```
roomDomainNew(1000..1020).
```

```
cabinetDomainNew(500..510).
```

```
% Define cost values as facts
```

```
reuseRoomCost(0).
```

```
reuseCabinetAsHighCost(3).
```

```
reuseCabinetAsSmallCost(0).
```

```
cabinetHighCost(100).
```

```
cabinetSmallCost(1).
```

```
roomCost(5).
```

```
removeCabinetCost(2).
```

```
removeRoomCost(2).
```

```
% Capacity Constraints
```

```
% Ensure cabinets hold at most 5 things
```

```
:- cabinet(C), #count { T : cabinetTOthing(C, T) } > 5.
```

```
% Ensure rooms hold at most 4 cabinets
```

```
:- room(R), #count { C : roomTOcabinet(R, C) } > 4.
```

```
% Compatibility Constraints
```

% Ensure only owned items are stored in a cabinet within the same room

$\text{:- room(R), roomTOcabinet(R, C), cabinetTOthing(C, T),}$   
 $\text{legacyConfig(person(P)), not legacyConfig(personTOthing(P, T)).}$

% Size Constraints

% Long items must be stored in high cabinets

$\text{:- thingLong(T), cabinetTOthing(C, T), not cabinetHigh(C).}$

% Assignments

% Ensure every thing is stored in exactly one cabinet

$1 \{ \text{cabinetTOthing(C, T) : cabinet(C) } \} 1 \text{ :- thing(T).}$

% Ensure every cabinet is assigned to exactly one room

$1 \{ \text{roomTOcabinet(R, C) : room(R) } \} 1 \text{ :- cabinet(C).}$

% Cabinet Types

% Define high or small cabinets

$\text{cabinetHigh(C) :- cabinet(C), not cabinetSmall(C).}$

$\text{cabinetSmall(C) :- cabinet(C), not cabinetHigh(C).}$

% Optimization

#minimize {

0 : room(R);

3 : cabinetHigh(C);

1 : cabinetSmall(C);

100 : not legacyConfig(cabinet(C)), cabinetHigh(C);

1 : not legacyConfig(cabinet(C)), cabinetSmall(C);

5 : not legacyConfig(room(R)), room(R);

2 : legacyConfig(cabinet(C)), not cabinet(C);

2 : legacyConfig(room(R)), not room(R)

}.

% Outputs

#show room/1.

#show cabinet/1.

#show cabinetTOthing/2.

#show roomTOcabinet/2.

#show thing/1.