DATABASE

Rooms	Room id	Room capacity	Room equipment
1	R01	12	Projector, smart board, and access for handicapped
2	R02	10	Projector
3	R03	5	Projector and access for handicapped
4	R04	8	Smart board and access for handicapped
5	R05	10	Smart board

Courses	Course id	Course capacity	Course instructor	Course needs
1	Course101	13	Instructor01	Access for handicapped
2	Course102	25	Instructor02	Smart board and projector
3	Course103	16	Instructor03	Projector
4	Course104	9	Instructor04	Projector and access for handicapped
5	Course105	15	Instructor05	Smart board
6	Course106	18	Instructor06	Projector, smart board, and access for handicapped

Occupied	course	room	hours
1	Course101	R03	9
2	Course101	R03	10
3	Course102	R01	8
4	Course104	R03	13
5	Course102	R01	10
6	Course103	R02	11
7	Course102	R01	9
8	Course104	R03	12
9	Course106	R01	10
10	Course105	R04	12
11	Course103	R02	8
12	Course106	R01	15
13	Course104	R03	9
14	Course103	R02	14

Students	Student id	Taken courses	Handicapped?
1	19010401	101	No
2	19010402	103, 106	Yes
3	19010403	104, 106	No
4	19010404	102	No
5	19010405	101	Yes
6	19010406	103	Yes
7	19010407	105,104	No

PART 1- TESTS

Add student:

```
1 ?- student(19010410,A,B).
false.

2 ?- add_student(19010410,[course101,course103],handcp).
true.

3 ?- student(19010410,A,B).
A = [course101, course103],
B = handcp.

4 ?-
```

Add room:

```
7 ?- room(room10,A,B).
false.

8 ?- add_room(room10,55,[projector,smart]).
true.

9 ?- room(room10,A,B).
A = 55,
B = [projector, smart].

10 ?-
```

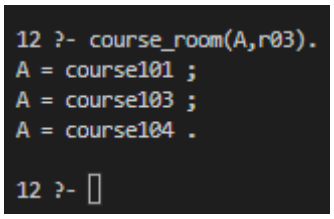
Add course:

```
4 ?- course(course10,A,B,C).
false.

5 ?- add_course(course10,11,new_ins,[projector,smart]).
true.

6 ?- course(course10,A,B,C).
A = 11,
B = new_ins,
C = [projector, smart].

7 ?-
```

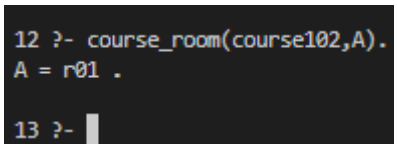
Check conflicts:

```
10 ?- check_conflict(course101,course102)
false.

11 ?- check_conflict(course101,course104)
true .

12 ?-
```

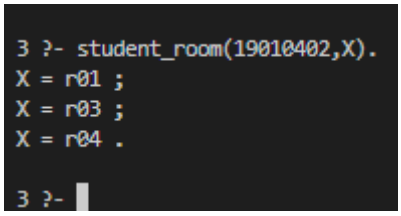
Check courses for given room:



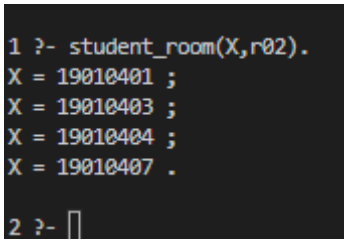
Check rooms for given course:



Check available rooms for given student:



Check students can use given room:



PART 2- TESTS

Find connected cities to given city and distance:

```
1 ?- connection(istanbul,A,B).
A = izmir,
B = 2 ;
A = rize,
B = 4 ;
A = ankara,
B = 1 ;
A = istanbul,
B = 4 ;
A = ankara,
B = 8 ;
A = antalya,
B = 4 ;
A = istanbul,
B = 9 ;
A = izmir,
B = 14 ;
A = rize,
B = 13 ;
A = van,
B = 12 ;
A = diyarbakir,
B = 16 ;
A = istanbul,
B = 17 ;
A = ankara,
B = 18 ;
A = gaziantep,
B = 15 ;
A = ankara,
B = 16 ;
A = van,
B = 18 ;
A = ankara,
B = 24 ;
A = antalya,
B = 20 ;
A = erzincan,
B = 23 ;
A = diyarbakir,
B = 24 ;
A = izmir,
B = 22 ;
A = canakkale,
B = 29 ;
A = antalya,
B = 26 ;
A = erzincan,
B = 35 ;
A = erzincan,
B = 7 ;
A = diyarbakir,
B = 8 ;
A = izmir,
B = 6 ;
A = canakkale,
B = 13
```

Find possible distances between given cities:

```
2 ?- connection(gaziantep,izmir,A).
A = 13 ;
A = 10 ;
A = 18 ;
A = 21 ;
false.

3 ?-
```

Find cities which have given distance between:

```
3 ?- connection(A,B,26).
A = canakkale,
B = rize ;
A = canakkale,
B = rize ;
A = canakkale,
B = istanbul ;
A = canakkale,
B = antalya ;
A = canakkale,
B = van ;
A = canakkale,
B = antalya ;
A = canakkale,
B = istanbul ;
A = canakkale,
B = istanbul ;
A = erzincan,
B = izmir ;
A = erzincan,
B = van ;
A = izmir,
B = erzincan ;
A = istanbul,
B = antalya ;
A = istanbul,
B = ankara ;
A = istanbul,
B = van ;
A = istanbul,
B = canakkale ;
A = van,
B = canakkale ;
A = rize,
B = izmir ;
A = rize,
B = izmir ;
A = rize,
B = canakkale ;
A = rize,
B = canakkale ;
A = gaziantep,
B = antalya ;
false.
```