German University in Cairo
Faculty of Media Engineering and Technology
Dr. Nada Sharaf

Concepts of Programming Languages, Spring 2019
The Helsinki Puzzle
Test Cases: Part I

grid_build(N,M)

The unbound variables are random so you should **not** expect these exact numbers.

```
?- grid_build(3,M).
M = [[_3322,_3328,_3334],[_3346,_3352,_3358],[_3370,_3376,_3382]].

?- grid_build(4,M).
M = [[_3160, _3166, _3172, _3178],[_3190, _3196, _3202, _3208],
[_3220, _3226, _3232, _3238], [_3250, _3256, _3262, _3268]].
```

grid_gen(N,M)

If the M is variable you are expected to generate a NxN matrix with the specified conditions for example the following query generates different matrices and the answer is only one of them.

```
?- grid_gen(3,M).
 M = [[2, 2, 1], [1, 2, 2], [2, 1, 2]].

?- grid_gen(4,[[1, 1, 2, 3], [1, 1, 3, 2], [3, 2, 4, 4], [2, 3, 4, 4]]).
 true ;
 false.

?- grid_gen(4,[[1, 1, 2, 3], [1, 1, 3, 2], [3, 2, 4, 4], [2, 3, 4, 5]]).
 false.

?- grid_gen(5,[[2, 3, 1, 3, 2], [2, 2, 3, 1, 3],[3, 2, 2, 3, 1],
[1, 3, 2, 2, 3], [3, 1, 3, 2, 2]]).
 true ;
 false.
```

German University in Cairo
Faculty of Media Engineering and Technology
Dr. Nada Sharaf

num_gen(F,L,R)

```
?- num_gen(1,3,R).
 R = [1,2,3].

?- num_gen(2,5,R).
 R = [2,3,4,5].
```

check_num_grid(N,M)

```
?-  check_num_grid([[3,1,3],[3,3,1],[1,3,3]]).
 false.

?-  check_num_grid([[3,1,3],[3,2,1],[1,3,3]]).
 true;
 false.

?-  check_num_grid([[3,1,3],[3,2,1],[1,3,4]]).
 false.

?-  check_num_grid([[1, 1, 2, 3], [1, 1, 3, 2], [3, 2, 4, 4], [2, 3, 4, 4]]).
 true;
 false.

?-  check_num_grid([[1, 1, 5, 3], [1, 1, 3, 2], [3, 5, 4, 4], [5, 3, 4, 4]]).
 false.
```

acceptable_distribution(M)

```
?-  acceptable_distribution([[1,2,3,4],[2,1,2,2],[3,3,1,3],[3,4,2,1]]).
 true;
 false.

?-  acceptable_distribution([[3,1,3],[3,3,1],[1,3,3]]).
 true;
 false.

?-  acceptable_distribution([[1, 2, 1], [2, 2, 1], [1, 2, 2]]).
 false.
```

German University in Cairo
Faculty of Media Engineering and Technology
Dr. Nada Sharaf

row_col_match(M)

```
?- row_col_match([[3,1,3],[3,3,1],[1,3,3]]).
 true;
 false.

   ?- row_col_match([[1,2,3,4],[2,1,2,2],[3,3,1,3],[3,4,2,1]]).
 false.

?- row_col_match([[1, 2, 1], [2, 2, 1], [1, 2, 2]]).
 false.
```

acceptable_permutation(L,R)

```
?- acceptable_permutation([1,2,3],R).
 R = [2,3,1] ;
 R = [3,1,2] ;
false

?- acceptable_permutation([1,2,3,4],R).
 R = [2,1,4,3] ;
 R = [2,3,4,1] ;
 R = [2,4,1,3] ;
 R = [3,1,4,2] ;
 R = [3,4,1,2] ;
 R = [3,4,2,1] ;
 R = [4,1,2,3] ;
 R = [4,3,1,2] ;
 R = [4,3,2,1] ;
 false.
```

trans(M,R)

```
?- trans([[1,2],[3,4]],R).
 R = [[1, 3], [2, 4]] ;
 false.
```

```
?- trans([[2,5,8],[3,6,9],[1,9,7]],R).
 R = [[2, 3, 1], [5, 6, 9], [8, 9, 7]] ;
 false.
```

distinct_rows(M)

```
?- distinct_rows([[2,5,8],[3,6,9],[1,9,7]]).
 true ;
 false.

?- distinct_rows([[2,5,8],[3,6,9],[2,5,8]]).
 false.

?- distinct_rows([[1,3,1,3],[2,1,2,1],[2,3,2,1],[1,2,1,2]]).
 true ;
 false.

?- distinct_rows([[1,2,2,1],[3,1,3,2],[1,2,2,1],[3,1,1,2]]).
 false.
```

distinct_columns(M)

```
?- distinct_columns([[2,5,8],[3,6,9],[1,9,7]]).
 true ;
 false.

?- distinct_columns([[2,5,2],[3,6,3],[2,6,2]]).
 false.

?- distinct_columns([[1,3,1,3],[2,1,2,1],[2,3,2,1],[1,2,1,2]]).
 false.

?- distinct_columns([[1,2,2,1],[3,1,3,2],[1,2,2,1],[3,1,1,2]]).
 true ;
 false.
```

German University in Cairo
Faculty of Media Engineering and Technology
Dr. Nada Sharaf

helsinki(N,M)

Depending on your approach, the order of the resulting grids and the arrangements of values inside of them might be completely different.

```
?- helsinki(3,M).
 M = [[2, 1, 2], [2, 2, 1], [1, 2, 2]] ;
 M = [[2, 2, 1], [1, 2, 2], [2, 1, 2]] ;
   ⋮
   ⋮
 false.

?- helsinki(4,M).
 M = [[1, 1, 2, 3], [1, 1, 3, 2], [3, 2, 4, 4], [2, 3, 4, 4]] ;
 M = [[1, 2, 2, 1], [1, 1, 2, 2], [2, 1, 1, 2], [2, 2, 1, 1]] ;
 M = [[1, 2, 1, 2], [1, 1, 2, 2], [2, 2, 1, 1], [2, 1, 2, 1]] ;
 M = [[1, 1, 2, 2], [2, 1, 2, 1], [1, 2, 1, 2], [2, 2, 1, 1]] ;
 M = [[1, 2, 1, 3], [3, 4, 2, 4], [1, 3, 1, 2], [2, 4, 3, 4]] ;
 M = [[1, 2, 2, 1], [2, 1, 1, 2], [1, 2, 1, 2], [2, 1, 2, 1]] ;
 M = [[1, 1, 2, 2], [2, 1, 1, 2], [2, 2, 1, 1], [1, 2, 2, 1]] ;
 M = [[1, 2, 1, 2], [2, 1, 2, 1], [2, 1, 1, 2], [1, 2, 2, 1]] ;
 M = [[1, 2, 3, 1], [3, 4, 4, 2], [2, 4, 4, 3], [1, 3, 2, 1]] ;
   ⋮
   ⋮
 false.
```

The default length of the output might not be able to display all the values in the grid, at which point you could just click "w" on the keyboard.

```
?- helsinki(5,M).
 M = [[2, 3, 1, 3, 2], [2, 2, 3, 1, 3], [3, 2, 2, 3, 1],
 [1, 3, 2, 2, 3], [3, 1, 3, 2|...]]
```

Just click "w" on your keyboard

```
 M = [[2, 3, 1, 3, 2], [2, 2, 3, 1, 3], [3, 2, 2, 3, 1],
 [1, 3, 2, 2, 3], [3, 1, 3, 2, 2]] ;
 M = [[2, 3, 1, 2, 3], [2, 2, 3, 3, 1], [3, 2, 2, 1, 3],
 [3, 1, 3, 2, 2], [1, 3, 2, 3, 2]] ;
 M = [[2, 3, 2, 1, 3], [2, 2, 3, 3, 1], [3, 1, 2, 3, 2],
 [3, 2, 1, 2, 3], [1, 3, 3, 2, 2]] ;
```

```
M = [[2, 3, 3, 1, 2], [2, 2, 1, 3, 3], [1, 3, 2, 2, 3],
[3, 2, 3, 2, 1], [3, 1, 2, 3, 2]] ;
M = [[2, 3, 2, 3, 1], [2, 2, 3, 1, 3], [3, 1, 2, 2, 3],
[1, 3, 3, 2, 2], [3, 2, 1, 3, 2]] ;
M = [[2, 3, 3, 2, 1], [2, 2, 1, 3, 3], [1, 3, 2, 3, 2],
[3, 1, 2, 2, 3], [3, 2, 3, 1, 2]] ;
M = [[2, 2, 3, 1, 3], [3, 2, 1, 3, 2], [2, 3, 2, 3, 1],
[3, 1, 2, 2, 3], [1, 3, 3, 2, 2]] ;
M = [[2, 2, 3, 3, 1], [3, 2, 1, 2, 3], [2, 3, 2, 1, 3],
[1, 3, 3, 2, 2], [3, 1, 2, 3, 2]] ;
M = [[2, 1, 3, 3, 2], [3, 2, 2, 3, 1], [2, 3, 2, 1, 3],
[1, 2, 3, 2, 3], [3, 3, 1, 2, 2]] ;
M = [[2, 3, 3, 2, 1], [1, 2, 3, 3, 2], [2, 1, 2, 3, 3],
[3, 2, 1, 2, 3], [3, 3, 2, 1, 2]] ;
M = [[2, 1, 3, 2, 3], [3, 2, 2, 1, 3], [2, 3, 2, 3, 1],
[3, 3, 1, 2, 2], [1, 2, 3, 3, 2]] ;
M = [[2, 3, 3, 1, 2], [1, 2, 3, 2, 3], [2, 1, 2, 3, 3],
[3, 3, 2, 2, 1], [3, 2, 1, 3, 2]] ;
M = [[2, 2, 3, 3, 1], [3, 2, 2, 1, 3], [1, 3, 2, 3, 2],
[2, 3, 1, 2, 3], [3, 1, 3, 2, 2]] ;
M = [[2, 2, 1, 3, 3], [3, 2, 3, 1, 2], [3, 1, 2, 2, 3],
[2, 3, 3, 2, 1], [1, 3, 2, 3, 2]] ;
M = [[2, 3, 2, 3, 1], [1, 2, 3, 3, 2], [3, 2, 2, 1, 3],
[2, 1, 3, 2, 3], [3, 3, 1, 2, 2]] ;
M = [[2, 1, 3, 3, 2], [3, 2, 3, 2, 1], [1, 2, 2, 3, 3],
[2, 3, 1, 2, 3], [3, 3, 2, 1, 2]] ;
M = [[2, 1, 2, 3, 3], [3, 2, 1, 2, 3], [3, 3, 2, 1, 2],
[2, 3, 3, 2, 1], [1, 2, 3, 3, 2]] ;
M = [[2, 3, 1, 3, 2], [1, 2, 2, 3, 3], [3, 3, 2, 2, 1],
[2, 1, 3, 2, 3], [3, 2, 3, 1, 2]] ;
M = [[2, 2, 3, 1, 3], [3, 2, 2, 3, 1], [1, 3, 2, 2, 3],
[3, 1, 3, 2, 2], [2, 3, 1, 3, 2]] ;
M = [[2, 2, 1, 3, 3], [3, 2, 3, 2, 1], [3, 1, 2, 3, 2],
[1, 3, 2, 2, 3], [2, 3, 3, 1, 2]] ;
M = [[2, 3, 2, 1, 3], [1, 2, 3, 2, 3], [3, 2, 2, 3, 1],
[3, 3, 1, 2, 2], [2, 1, 3, 3, 2]] ;
M = [[2, 1, 3, 2, 3], [3, 2, 3, 1, 2], [1, 2, 2, 3, 3],
[3, 3, 2, 2, 1], [2, 3, 1, 3, 2]] ;
M = [[2, 1, 2, 3, 3], [3, 2, 1, 3, 2], [3, 3, 2, 2, 1],
[1, 2, 3, 2, 3], [2, 3, 3, 1, 2]] ;
```

```
M = [[2, 3, 1, 2, 3], [1, 2, 2, 3, 3], [3, 3, 2, 1, 2],
[3, 2, 3, 2, 1], [2, 1, 3, 3, 2]] ;
  ⋮
  ⋮
false.
```

If it takes 2-3 minutes to compute results for N = 4 then you could simply pass the matrix itself as an argument to the helsinki predicate.

```
?- helsinki(5,[[2, 3, 1, 3, 2], [2, 2, 3, 1, 3],
[3, 2, 2, 3, 1], [1, 3, 2, 2, 3], [3, 1, 3, 2, 2]]).
true ;
false.
```