Stable marriage using backtracking

- You have n men and n woman, and their preference rankings of each other, and you need to match them up so that the total matching is “stable.”

- You are given two N * N arrays, mp (men’s preference) which gives the men’s ranking of the women, and wp (women’s preference) which gives the women’s ranking of the men. So mp[i][j] gives man i's ranking of woman j and likewise for the women’s ranking of the men in wp.
**Stable Marriage**

<table>
<thead>
<tr>
<th>Men’s preference</th>
<th>W0</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>M1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>M2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Women’s preference</th>
<th>M0</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>W1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>W2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Man 1 assigns Woman 2 the rank of 1(second highest) and prefers Woman 0 the best.**
Instability

For example: Man 1 marries Woman 1;
   Man 2 marries Woman 2;
There is an instability if:
   - Man 1 and Woman 2 like each other more than their partner;
   - Man 2 and Woman 1 like each other more than their partner.

But if Man 1 likes Woman 2 but Woman 2 hates Man 1, it is not considered as instability.
Preference rankings:

```java
static int mp[3][3] = {{0,2,1}, // Man#0's preferences
                      {0,2,1}, // Man#1's preferences
                      {1,2,0}}; // Man#2's preferences

static int wp[3][3] = {{2,1,0}, // Woman#0's preferences
                      {0,1,2}, // Woman#1's preferences
                      {2,0,1}}; // Woman#2's preferences
```

- `mp[i][j]` tells how much Man#i likes Woman#j.
- `wp[i][j]` tells how much Woman#i likes Man#j.
- `mp[i][q[i]]` tells how much Man#i likes his own wife.
- `mp[i][q[c]]` tells how much Man#i likes Man#c’s wife.
- `wp[q[i]][i]` tells how much Man#i’s wife likes Man#i.
- `wp[q[i]][c]` tells how much Man#i’s wife likes Man#c.

If `mp[i][q[c]] > mp[i][q[i]]`, it means that Man#i likes his own wife more than Man#c’s wife.
Stable Marriage Using Backtracking

- The main body should have similar code to what we did in 1D Eight Queens.

- The major part we need to modify is ok function.
Print function & Sample Output:

```cpp
void print(int q[]) {
    static int solution = 0;
    cout << "Solution #" << ++solution << ":
Man\tWoman\n";
    for (int i = 0; i < 3; ++i)
        cout << i << "\t" << q[i] << "\n";
    cout << "\n";
}

Solution #1:
Man   Woman
0     1
1     0
2     2

Solution #2:
Man   Woman
0     1
1     2
2     0
```
bool ok(int q[], int col){
    int mp[3][3] = {{0,2,1},{0,2,1},{1,2,0}};
    int wp[3][3] = {{2,1,0},{0,1,2},{2,0,1}};
    int nm = col; // new man
    int nw = q[col]; // new woman

    for(int i = 0; i < 3; i++){
        // if the new woman has already been assigned to some man then return false

        // check the new pair (new man, new woman) against each existing pair consisting of
        // (current man, current woman) to see if the new pair would make the match unstable. So
        // a. if the current man prefers the new woman to his partner and
        // b. if the new woman prefers the current man to her partner
        //    i. this is unstable, so return false
        // c. if the new man prefers the current woman to his partner and
        // d. if the current woman prefers the new man to her partner
        //    i. This is unstable, so return false
    }
    return true;
}