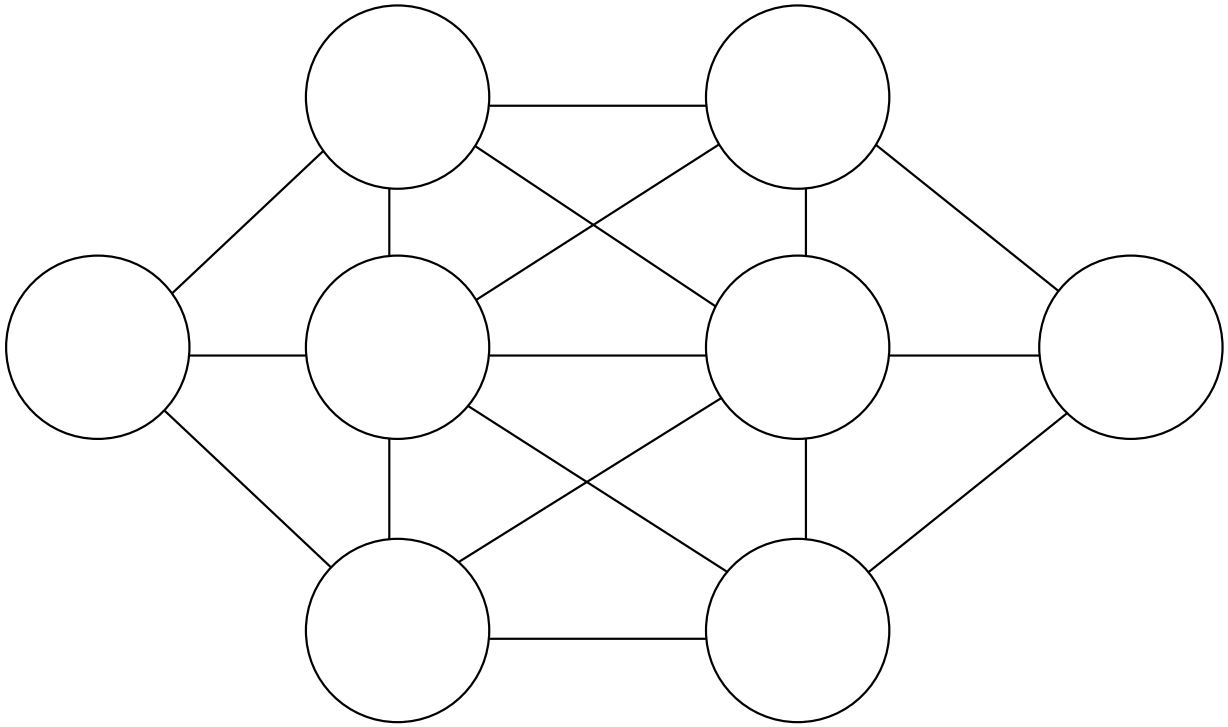
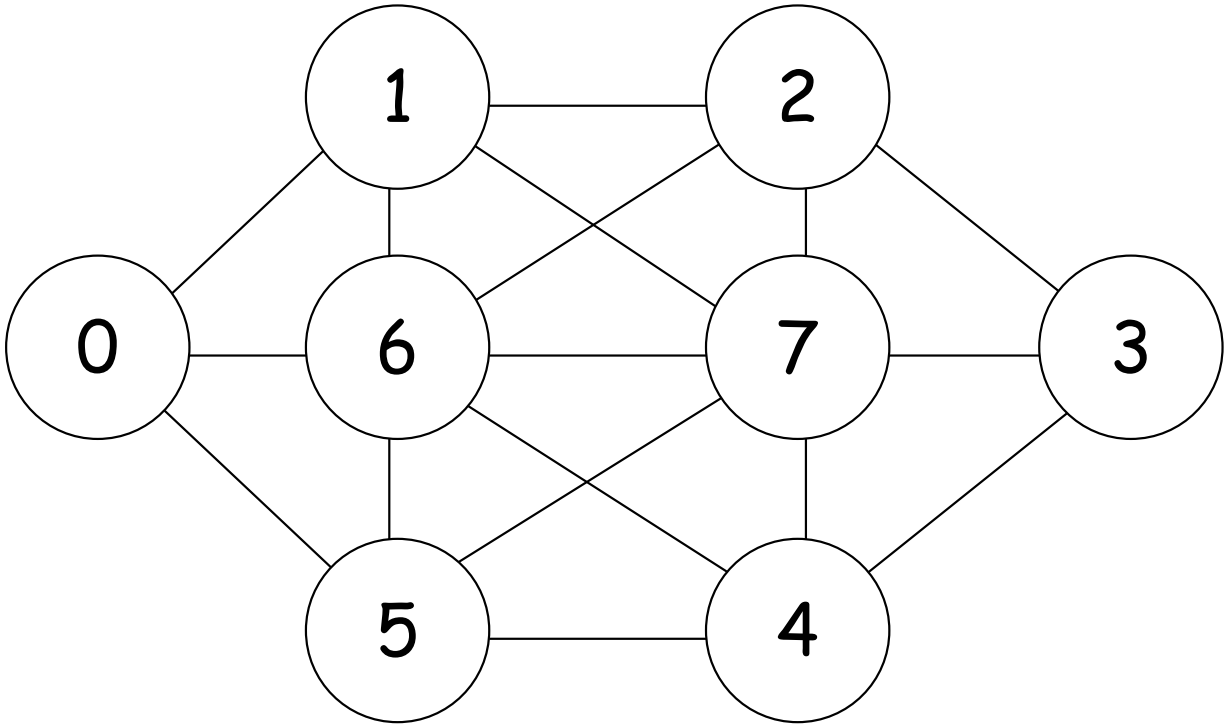
Crystal Maze

coded in minizinc

Author: Patrick Prosser (Glasgow University)

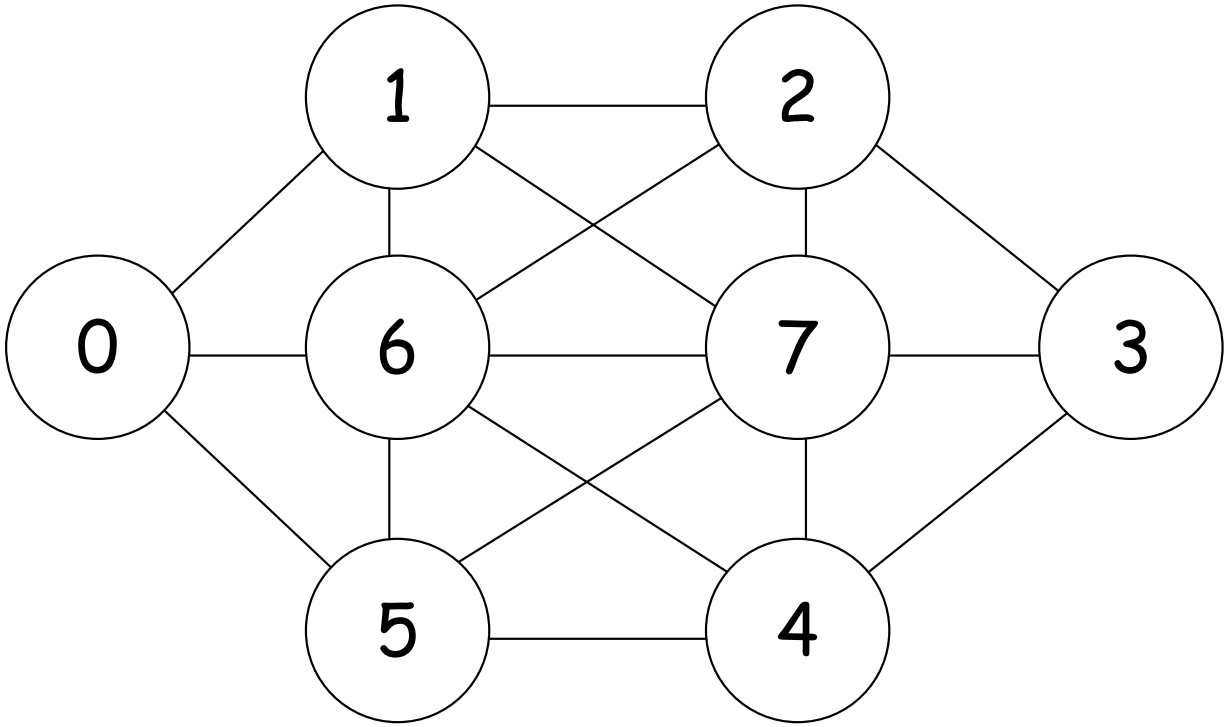


Put a different number in each circle (1 to 8) such that adjacent circles cannot take consecutive numbers

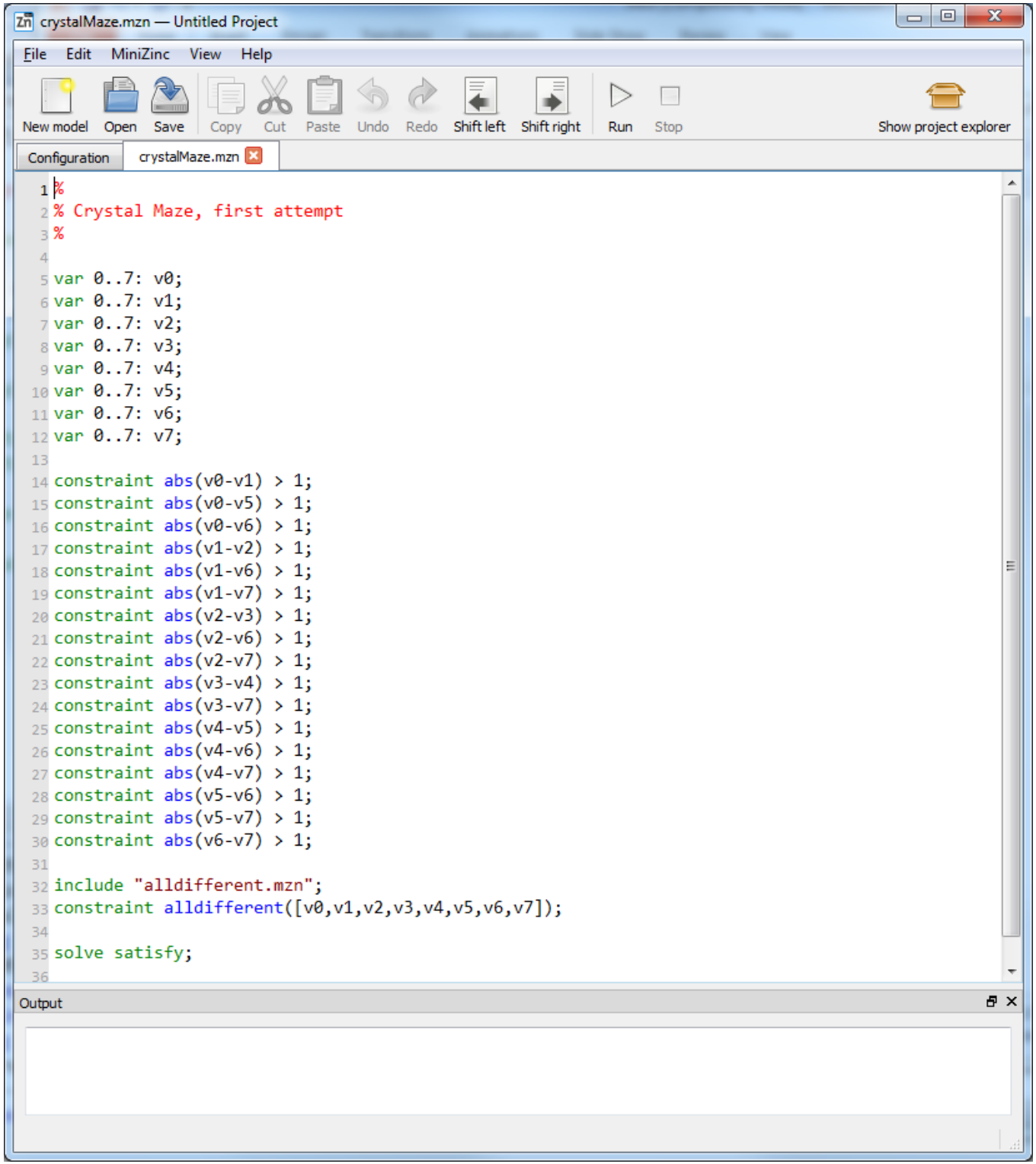


Put a different number in each circle (0 to 7) such that adjacent circles cannot take consecutive numbers

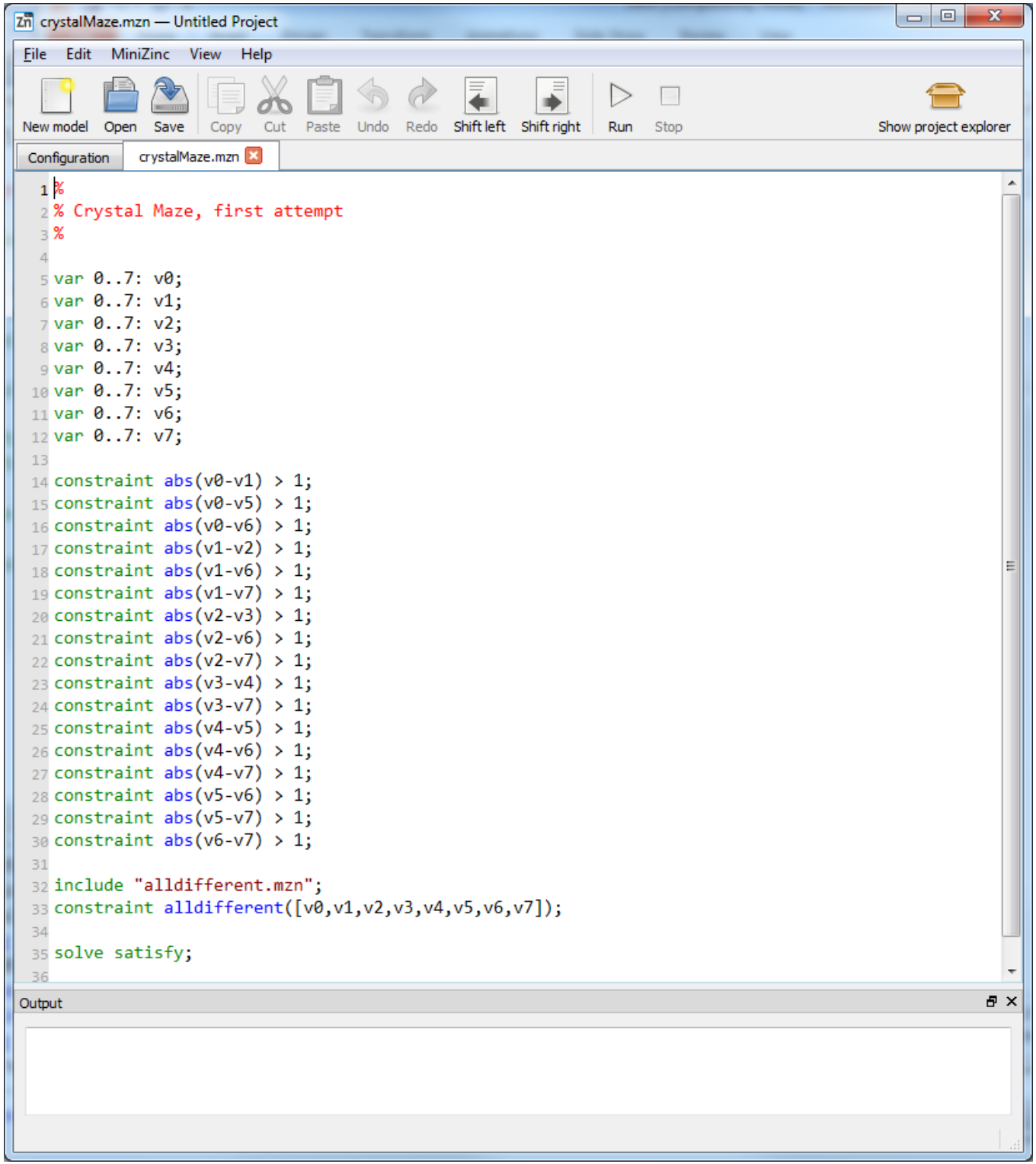
The numbers are the identification of a circle



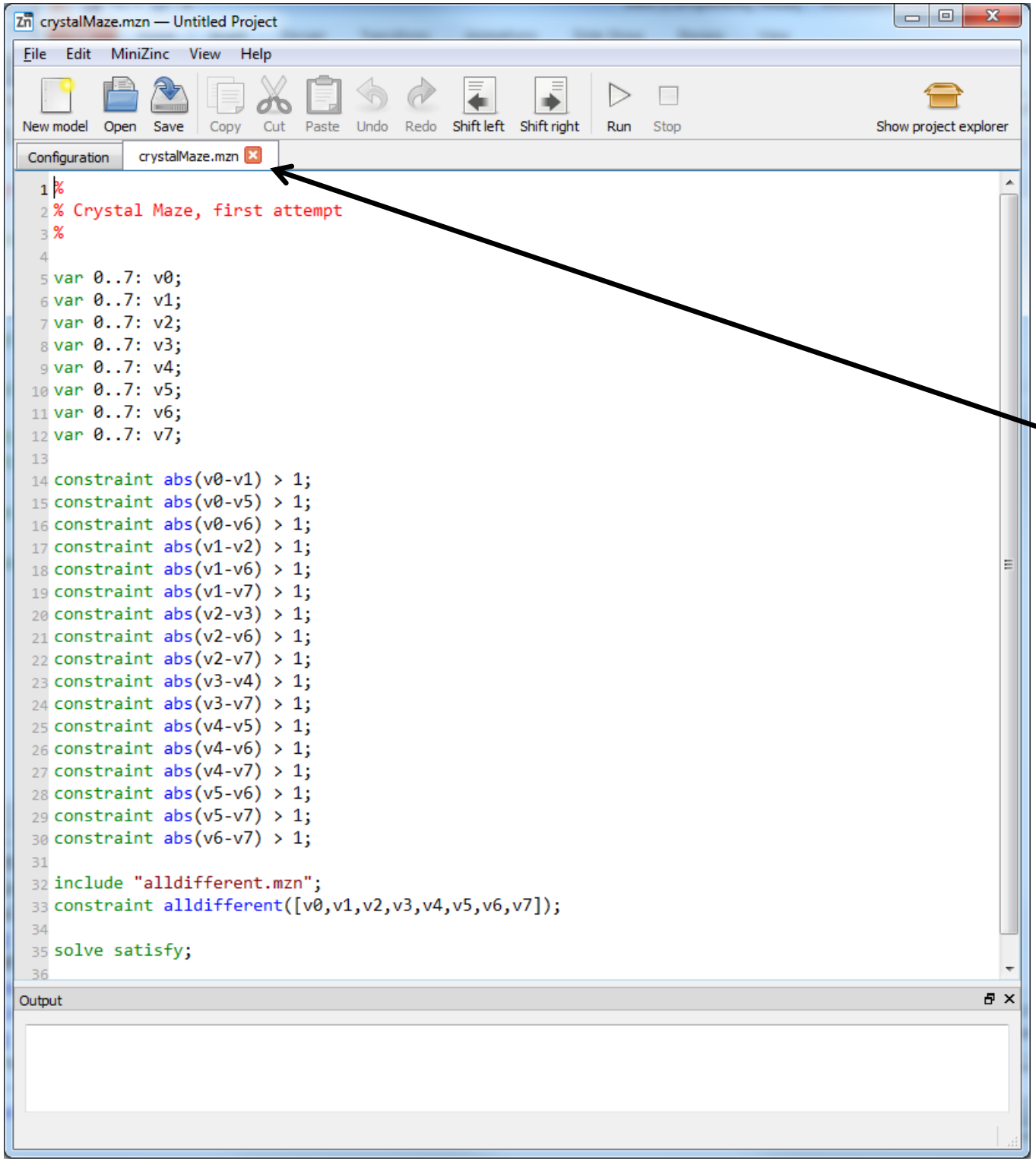
Put a different number in each circle (0 to 7) such that adjacent circles cannot take consecutive numbers


crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
1 %
2 % Crystal Maze, first attempt
3 %
4
5 var 0..7: v0;
6 var 0..7: v1;
7 var 0..7: v2;
8 var 0..7: v3;
9 var 0..7: v4;
10 var 0..7: v5;
11 var 0..7: v6;
12 var 0..7: v7;
13
14 constraint abs(v0-v1) > 1;
15 constraint abs(v0-v5) > 1;
16 constraint abs(v0-v6) > 1;
17 constraint abs(v1-v2) > 1;
18 constraint abs(v1-v6) > 1;
19 constraint abs(v1-v7) > 1;
20 constraint abs(v2-v3) > 1;
21 constraint abs(v2-v6) > 1;
22 constraint abs(v2-v7) > 1;
23 constraint abs(v3-v4) > 1;
24 constraint abs(v3-v7) > 1;
25 constraint abs(v4-v5) > 1;
26 constraint abs(v4-v6) > 1;
27 constraint abs(v4-v7) > 1;
28 constraint abs(v5-v6) > 1;
29 constraint abs(v5-v7) > 1;
30 constraint abs(v6-v7) > 1;
31
32 include "alldifferent.mzn";
33 constraint alldifferent([v0,v1,v2,v3,v4,v5,v6,v7]);
34
35 solve satisfy;
36
Output

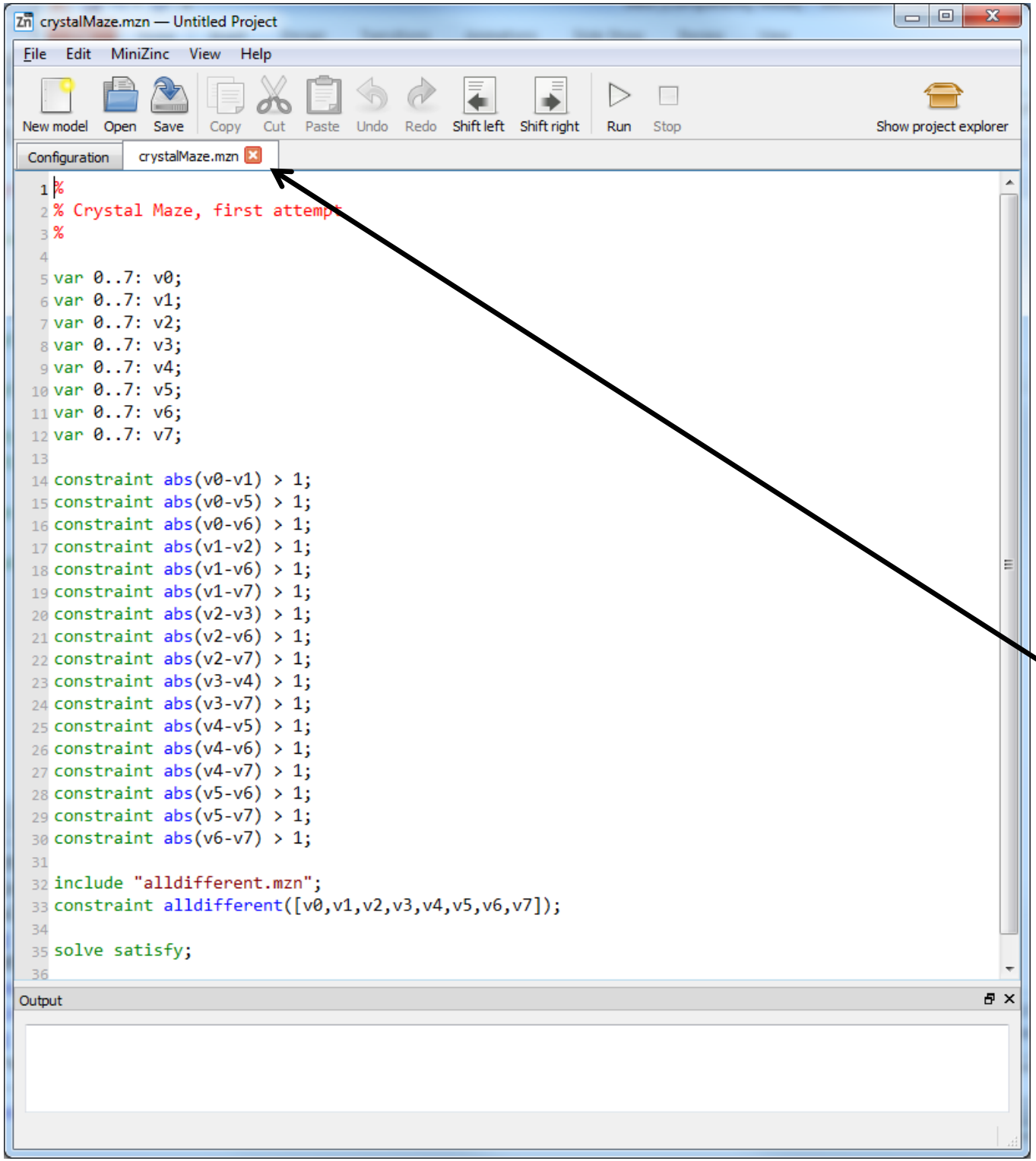
1st stab …



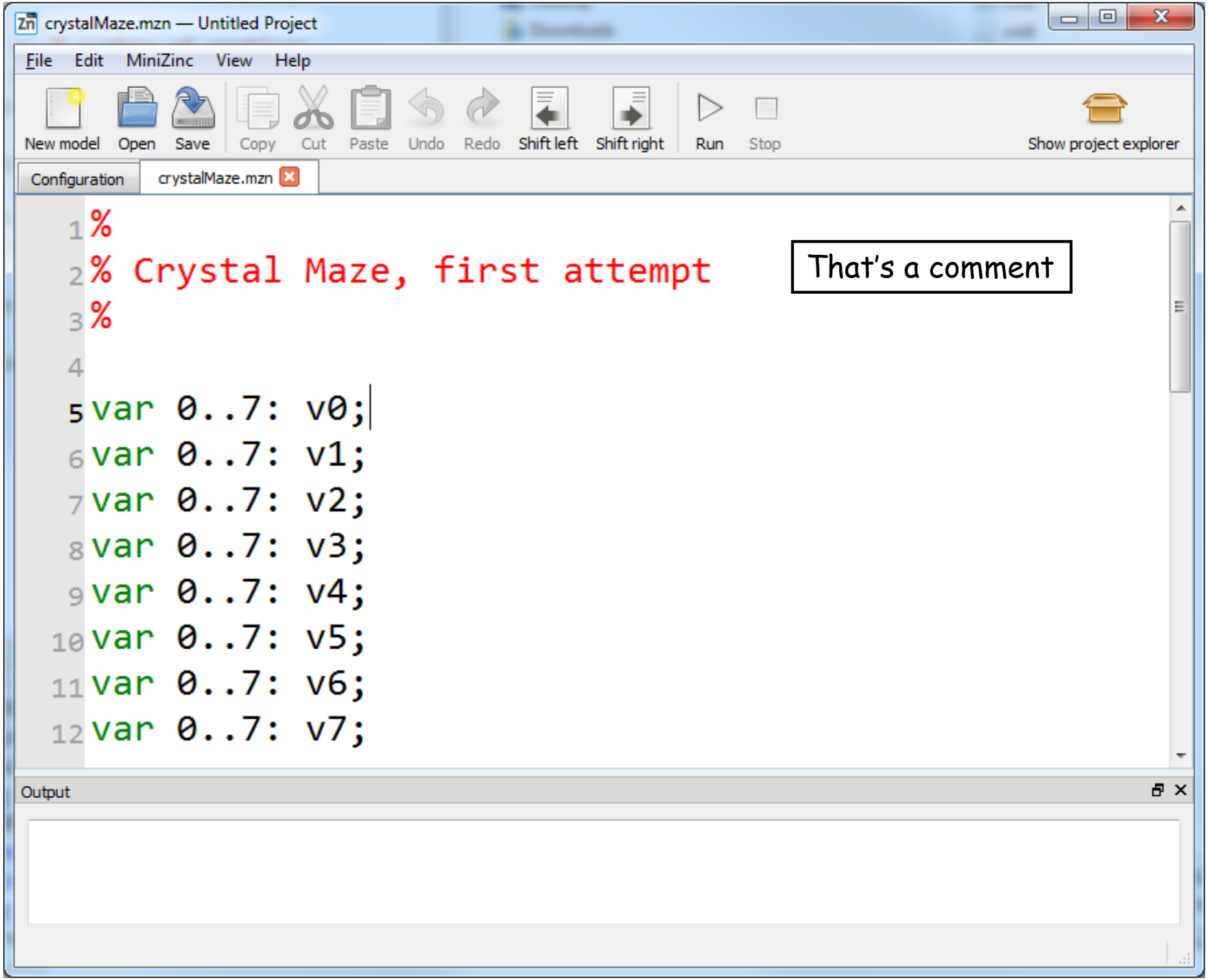
We are in the IDE

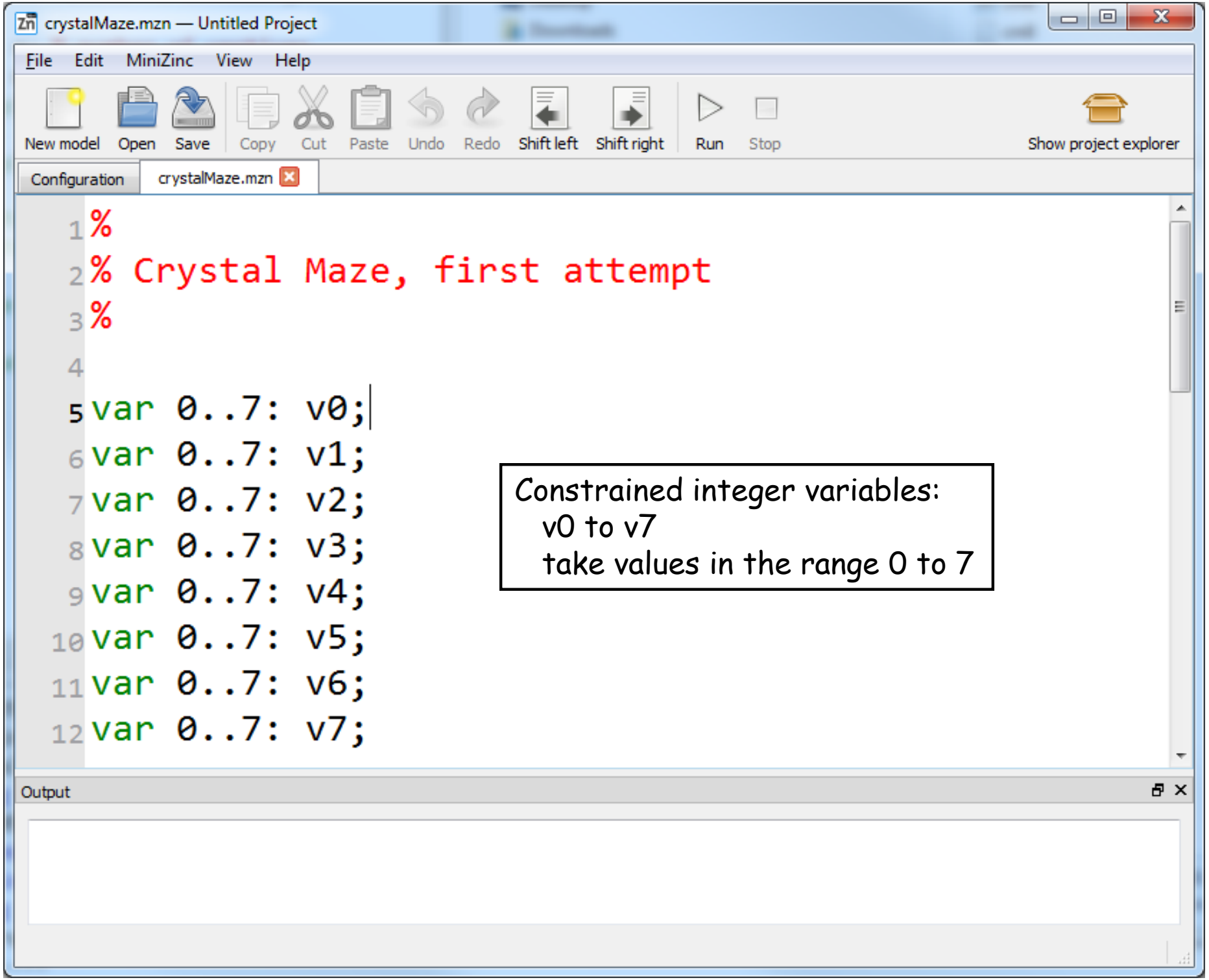


crystalMaze.mzn

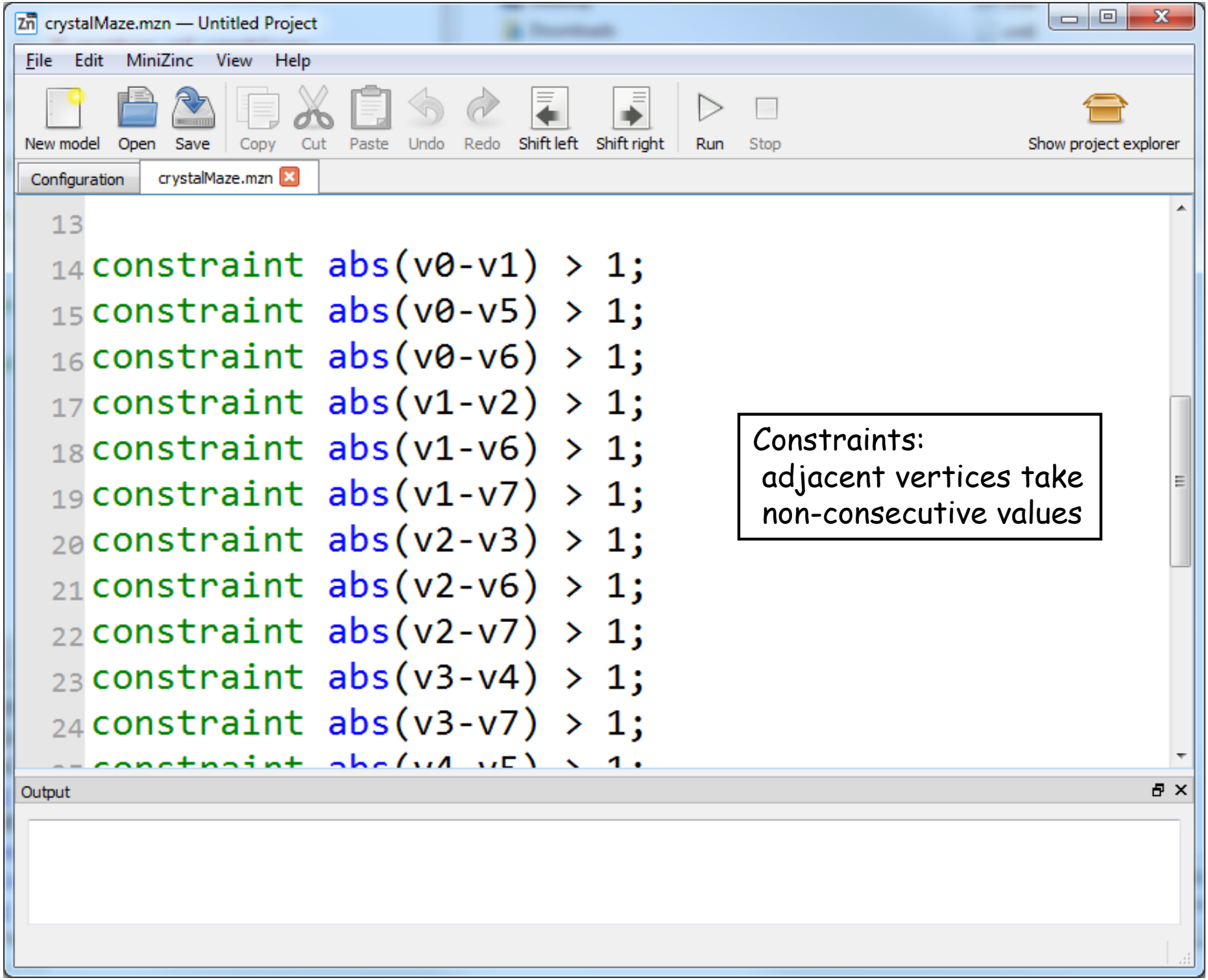


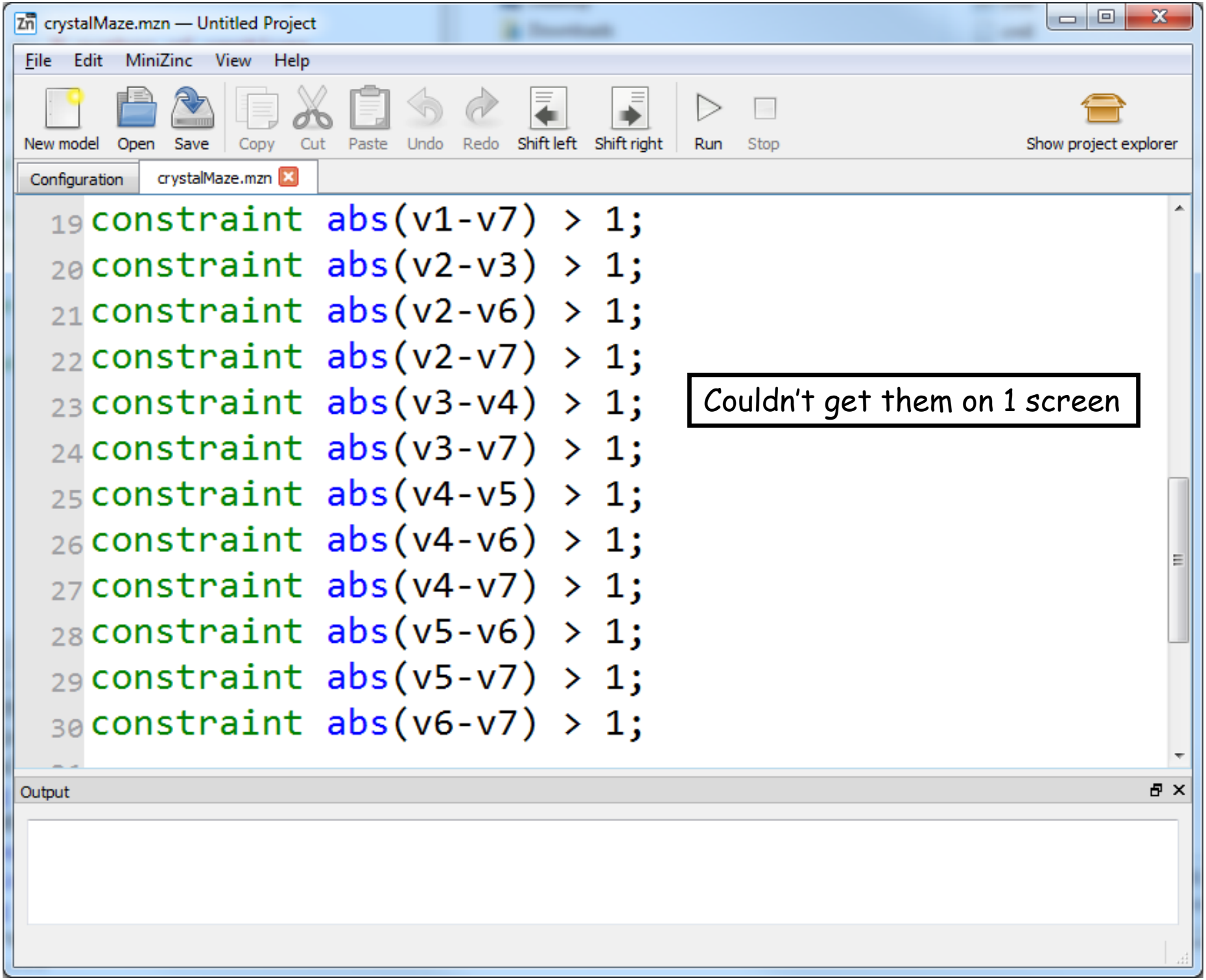
We call this "a model"

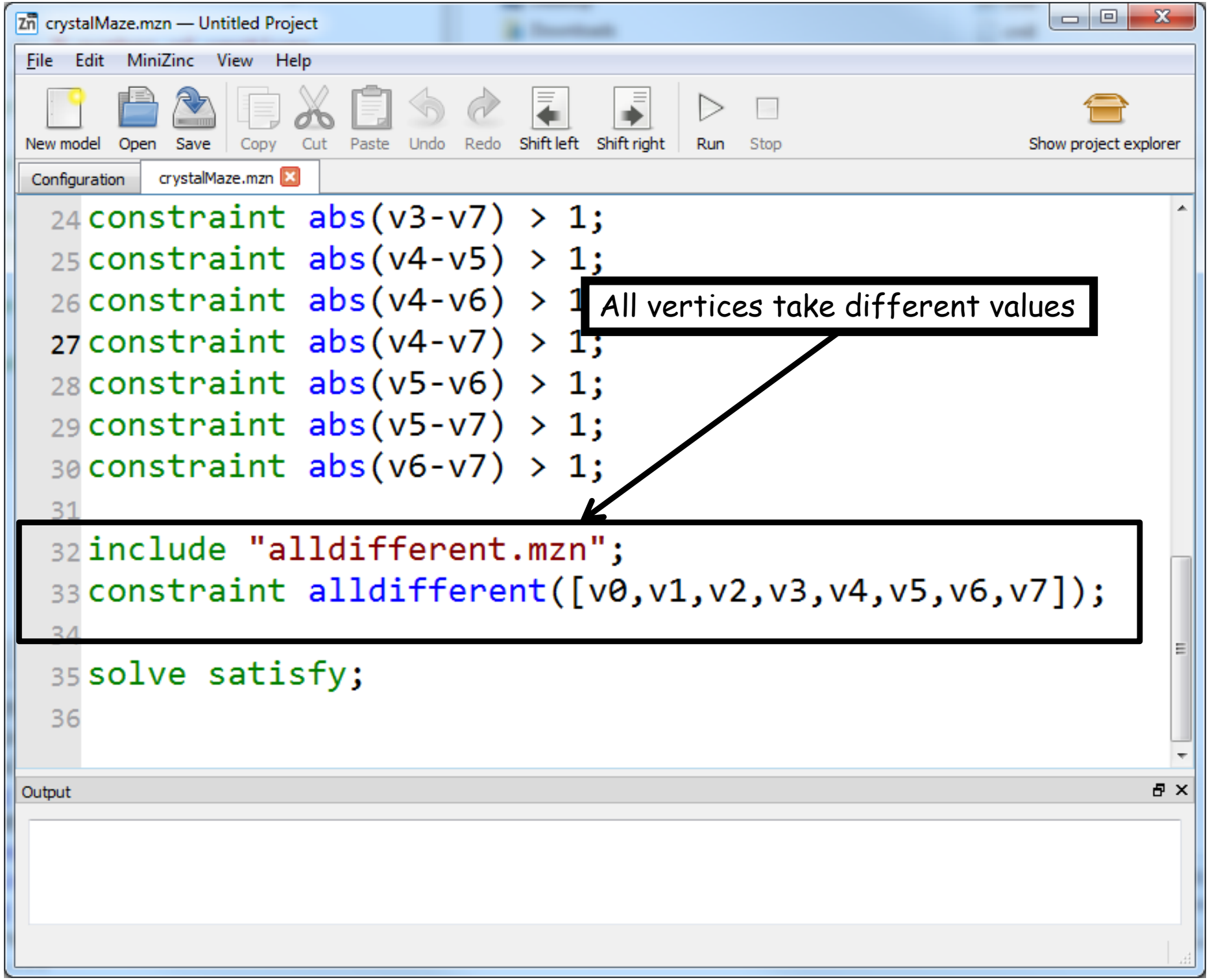

crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
1 %
2 % Crystal Maze, first attempt
3 %
4
5 var 0..7: v0;
6 var 0..7: v1;
7 var 0..7: v2;
8 var 0..7: v3;
9 var 0..7: v4;
10 var 0..7: v5;
11 var 0..7: v6;
12 var 0..7: v7;
That's a comment
Output

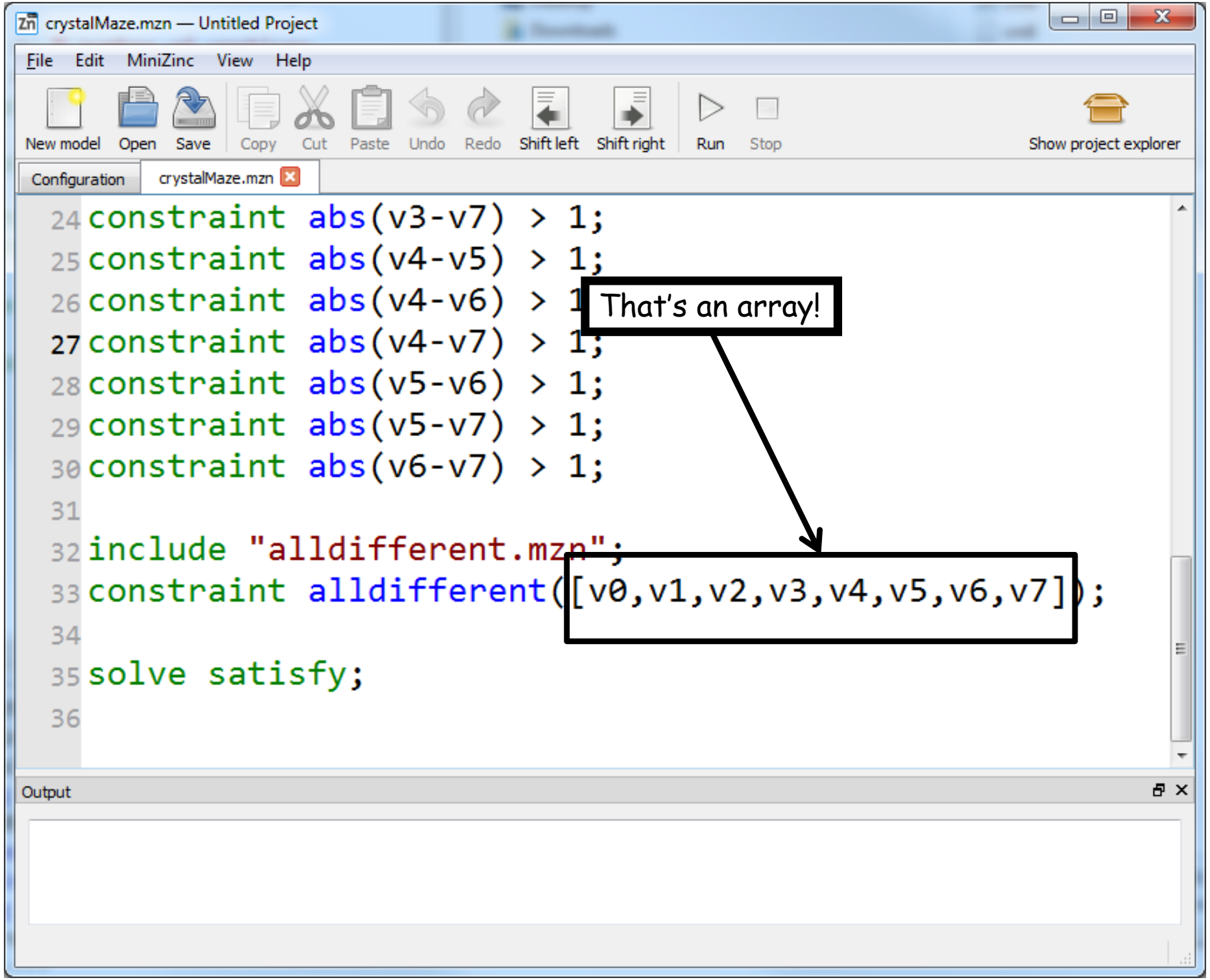


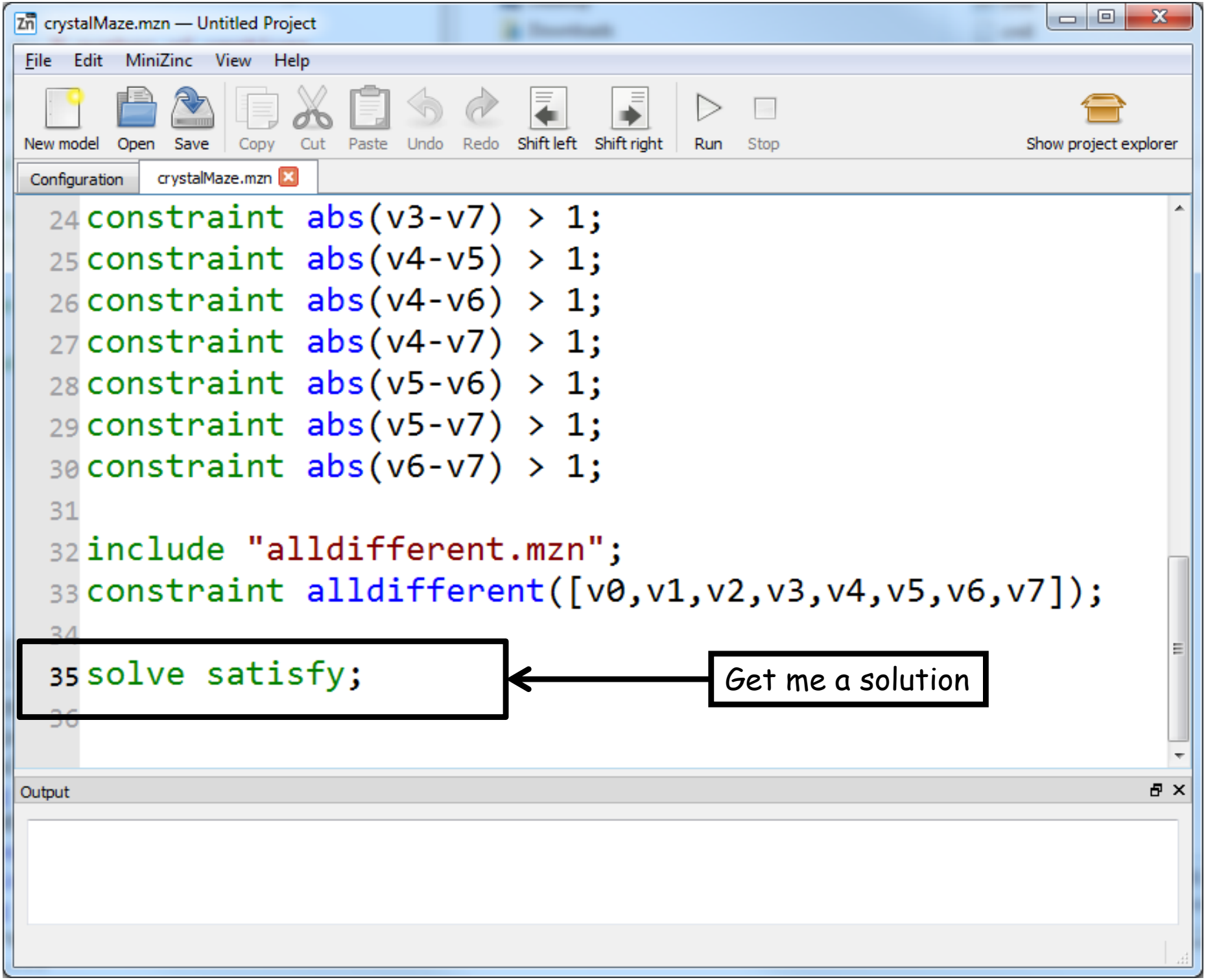
Constrained integer variables:
v0 to v7
take values in the range 0 to 7

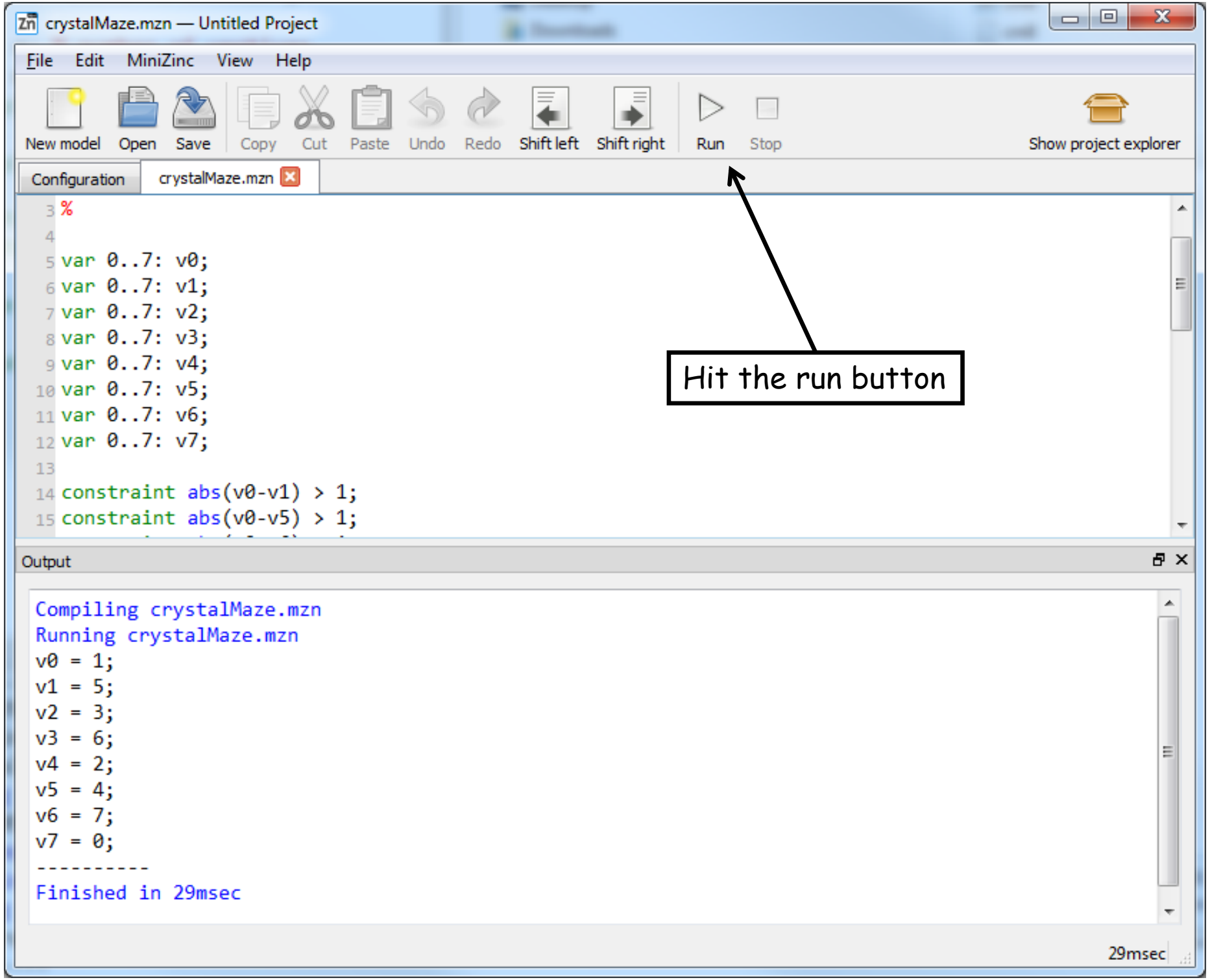

crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
13
14 constraint abs(v0-v1) > 1;
15 constraint abs(v0-v5) > 1;
16 constraint abs(v0-v6) > 1;
17 constraint abs(v1-v2) > 1;
18 constraint abs(v1-v6) > 1;
19 constraint abs(v1-v7) > 1;
20 constraint abs(v2-v3) > 1;
21 constraint abs(v2-v6) > 1;
22 constraint abs(v2-v7) > 1;
23 constraint abs(v3-v4) > 1;
24 constraint abs(v3-v7) > 1;
Output
Constraints:
adjacent vertices take
non-consecutive values


crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
19 constraint abs(v1-v7) > 1;
20 constraint abs(v2-v3) > 1;
21 constraint abs(v2-v6) > 1;
22 constraint abs(v2-v7) > 1;
23 constraint abs(v3-v4) > 1;
24 constraint abs(v3-v7) > 1;
25 constraint abs(v4-v5) > 1;
26 constraint abs(v4-v6) > 1;
27 constraint abs(v4-v7) > 1;
28 constraint abs(v5-v6) > 1;
29 constraint abs(v5-v7) > 1;
30 constraint abs(v6-v7) > 1;
Couldn't get them on 1 screen
Output

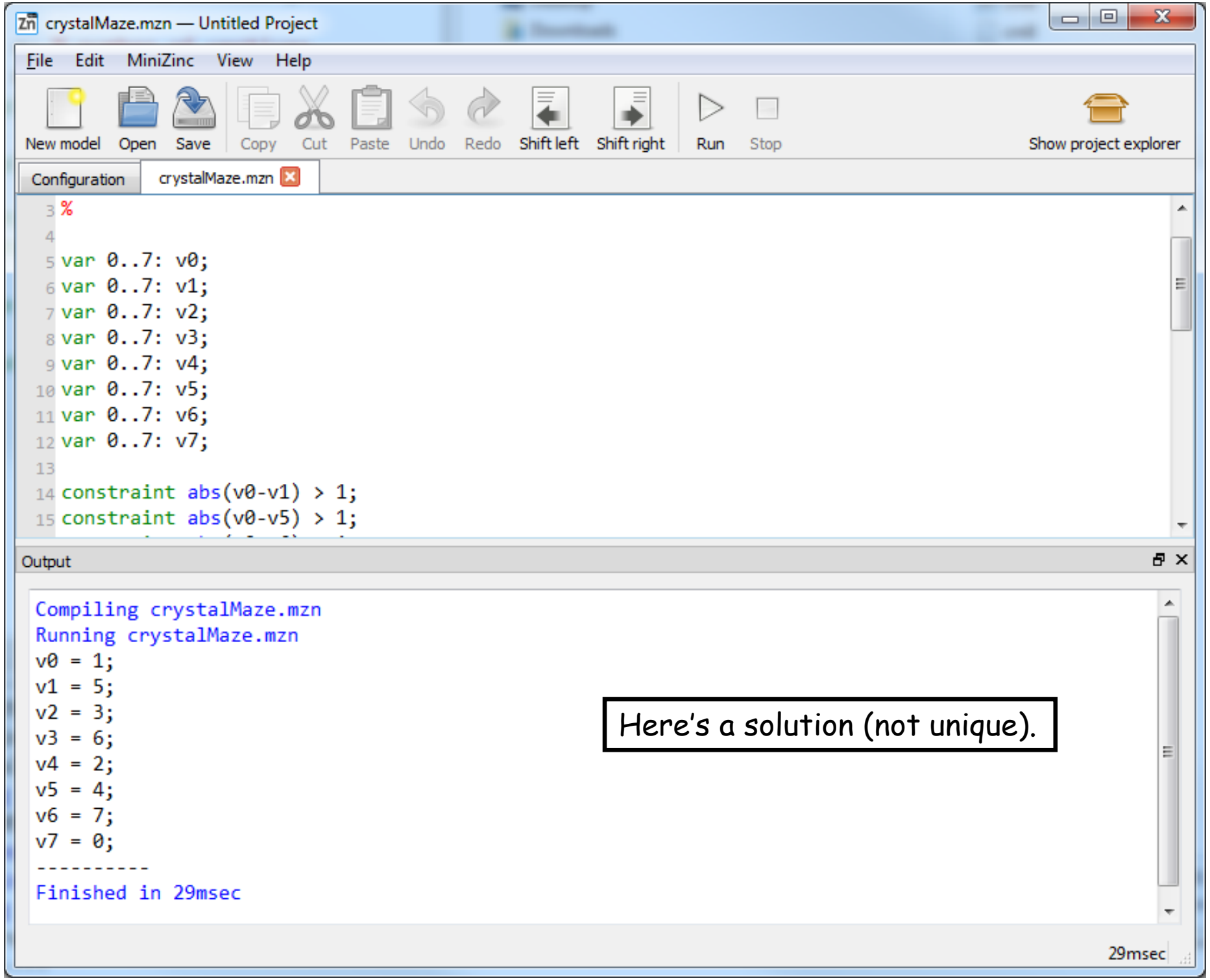

crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
24 constraint abs(v3-v7) > 1;
25 constraint abs(v4-v5) > 1;
26 constraint abs(v4-v6) > 1;
27 constraint abs(v4-v7) > 1;
28 constraint abs(v5-v6) > 1;
29 constraint abs(v5-v7) > 1;
30 constraint abs(v6-v7) > 1;
31
32 include "alldifferent.mzn";
33 constraint alldifferent([v0,v1,v2,v3,v4,v5,v6,v7]);
34
35 solve satisfy;
36
All vertices take different values
Output


crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze.mzn
24 constraint abs(v3-v7) > 1;
25 constraint abs(v4-v5) > 1;
26 constraint abs(v4-v6) > 1;
27 constraint abs(v4-v7) > 1;
28 constraint abs(v5-v6) > 1;
29 constraint abs(v5-v7) > 1;
30 constraint abs(v6-v7) > 1;
31
32 include "alldifferent.mzn";
33 constraint alldifferent([v0,v1,v2,v3,v4,v5,v6,v7]);
34
35 solve satisfy;
36
That's an array!
Output


crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze.mzn
24 constraint abs(v3-v7) > 1;
25 constraint abs(v4-v5) > 1;
26 constraint abs(v4-v6) > 1;
27 constraint abs(v4-v7) > 1;
28 constraint abs(v5-v6) > 1;
29 constraint abs(v5-v7) > 1;
30 constraint abs(v6-v7) > 1;
31
32 include "alldifferent.mzn";
33 constraint alldifferent([v0,v1,v2,v3,v4,v5,v6,v7]);
34
35 solve satisfy;
Get me a solution
Output


crystalMaze.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze.mzn
3 %
4
5 var 0..7: v0;
6 var 0..7: v1;
7 var 0..7: v2;
8 var 0..7: v3;
9 var 0..7: v4;
10 var 0..7: v5;
11 var 0..7: v6;
12 var 0..7: v7;
13
14 constraint abs(v0-v1) > 1;
15 constraint abs(v0-v5) > 1;
Hit the run button
Output
Compiling crystalMaze.mzn
Running crystalMaze.mzn
v0 = 1;
v1 = 5;
v2 = 3;
v3 = 6;
v4 = 2;
v5 = 4;
v6 = 7;
v7 = 0;

Finished in 29msec
29msec



Here's a solution (not unique).

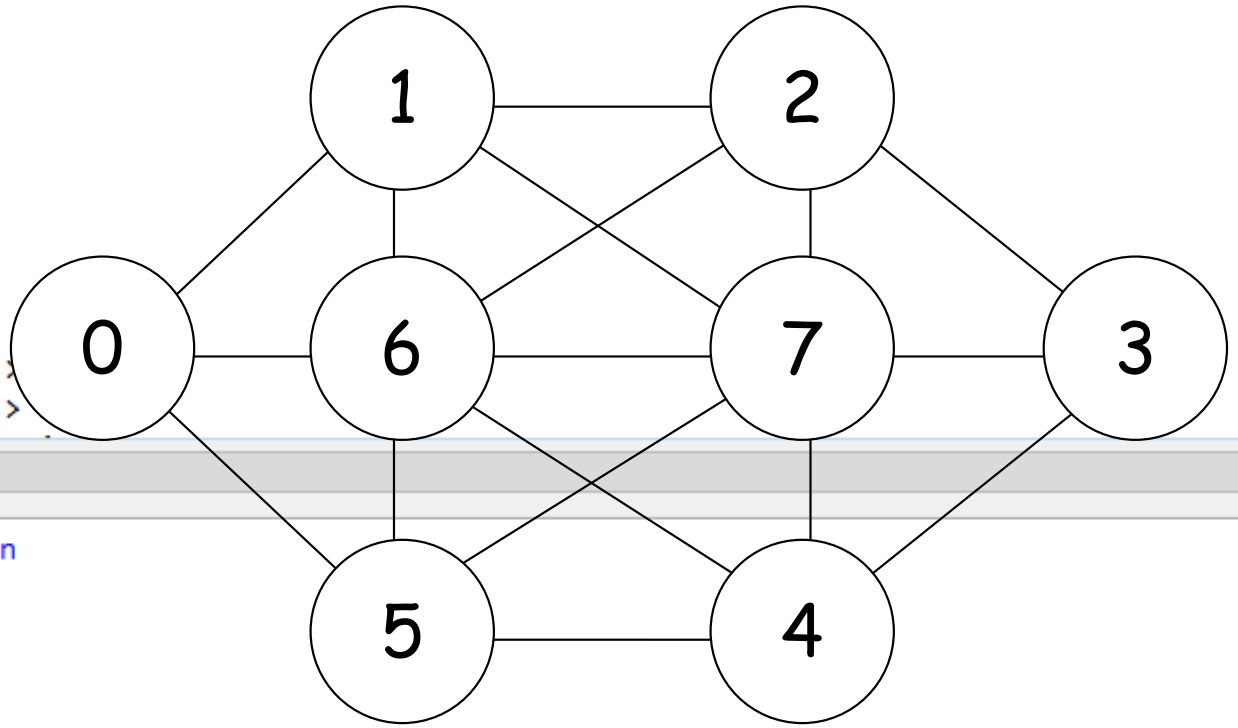
crystalMaze.mzn — Untitled Project

File Edit MiniZinc View Help

New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer

Configuration crystalMaze.mzn

```
3 %
4
5 var 0..7: v0;
6 var 0..7: v1;
7 var 0..7: v2;
8 var 0..7: v3;
9 var 0..7: v4;
10 var 0..7: v5;
11 var 0..7: v6;
12 var 0..7: v7;
13
14 constraint abs(v0-v1) >
15 constraint abs(v0-v5) >
```



Output

```
Compiling crystalMaze.mzn
Running crystalMaze.mzn
v0 = 1;
v1 = 5;
v2 = 3;
v3 = 6;
v4 = 2;
v5 = 4;
v6 = 7;
v7 = 0;
----------
Finished in 29msec
```

29msec

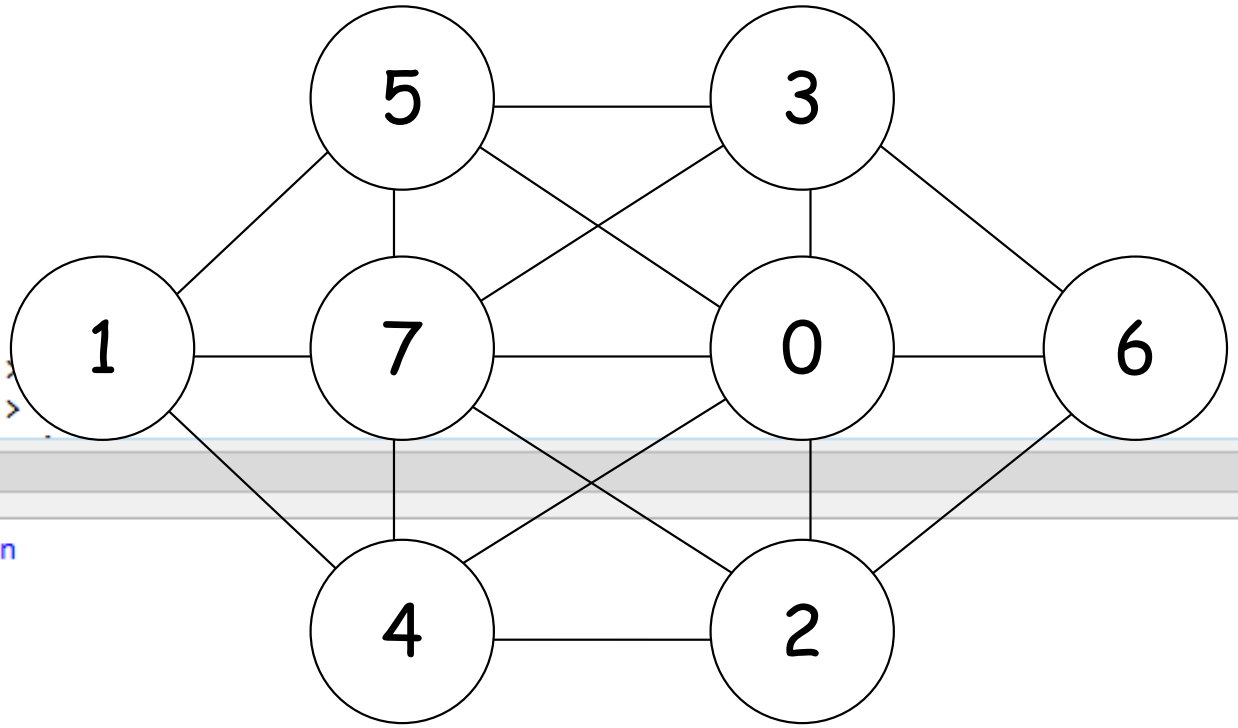
crystalMaze.mzn — Untitled Project

File Edit MiniZinc View Help

New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer

Configuration crystalMaze.mzn

```
3 %
4
5 var 0..7: v0;
6 var 0..7: v1;
7 var 0..7: v2;
8 var 0..7: v3;
9 var 0..7: v4;
10 var 0..7: v5;
11 var 0..7: v6;
12 var 0..7: v7;
13
14 constraint abs(v0-v1) >
15 constraint abs(v0-v5) >
```

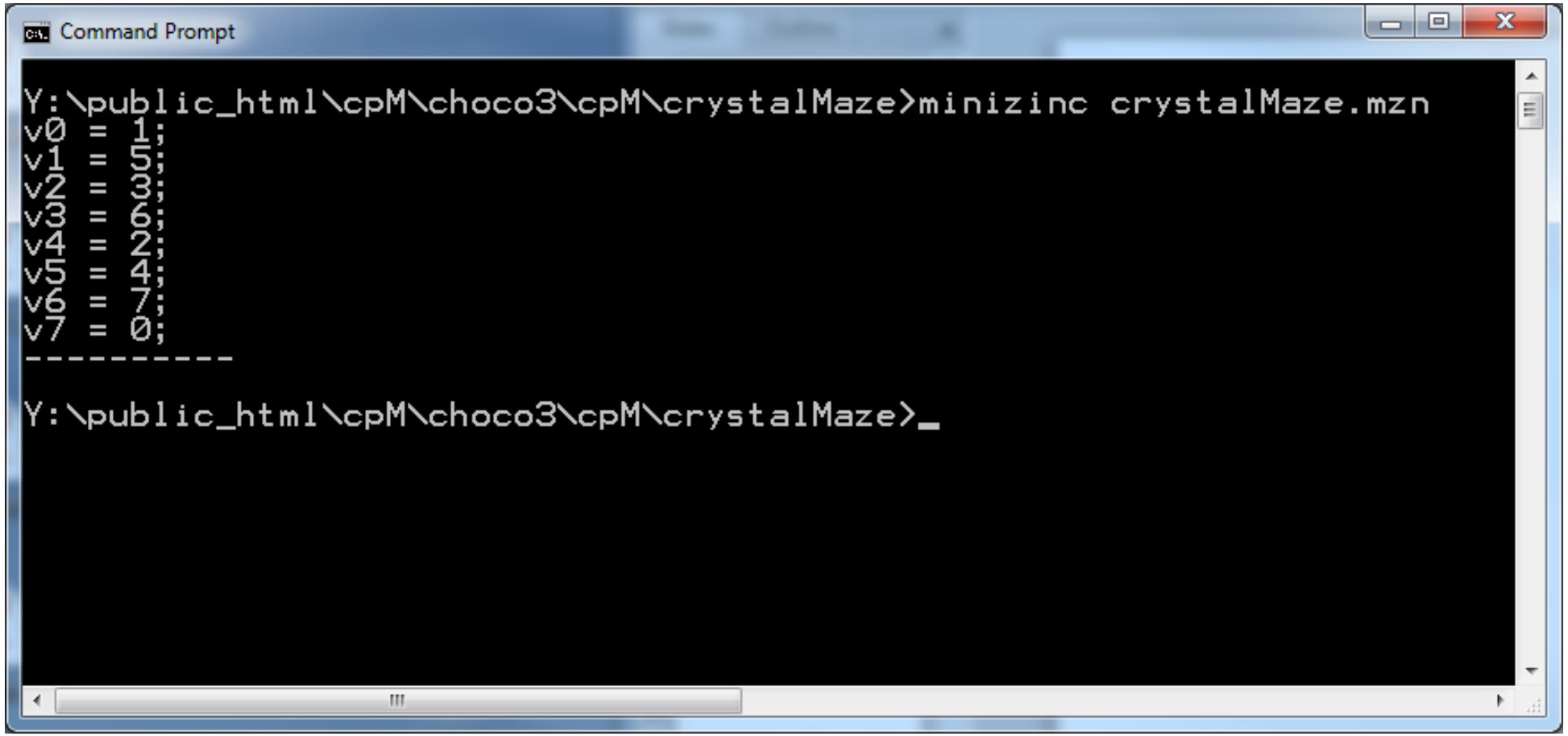


Output

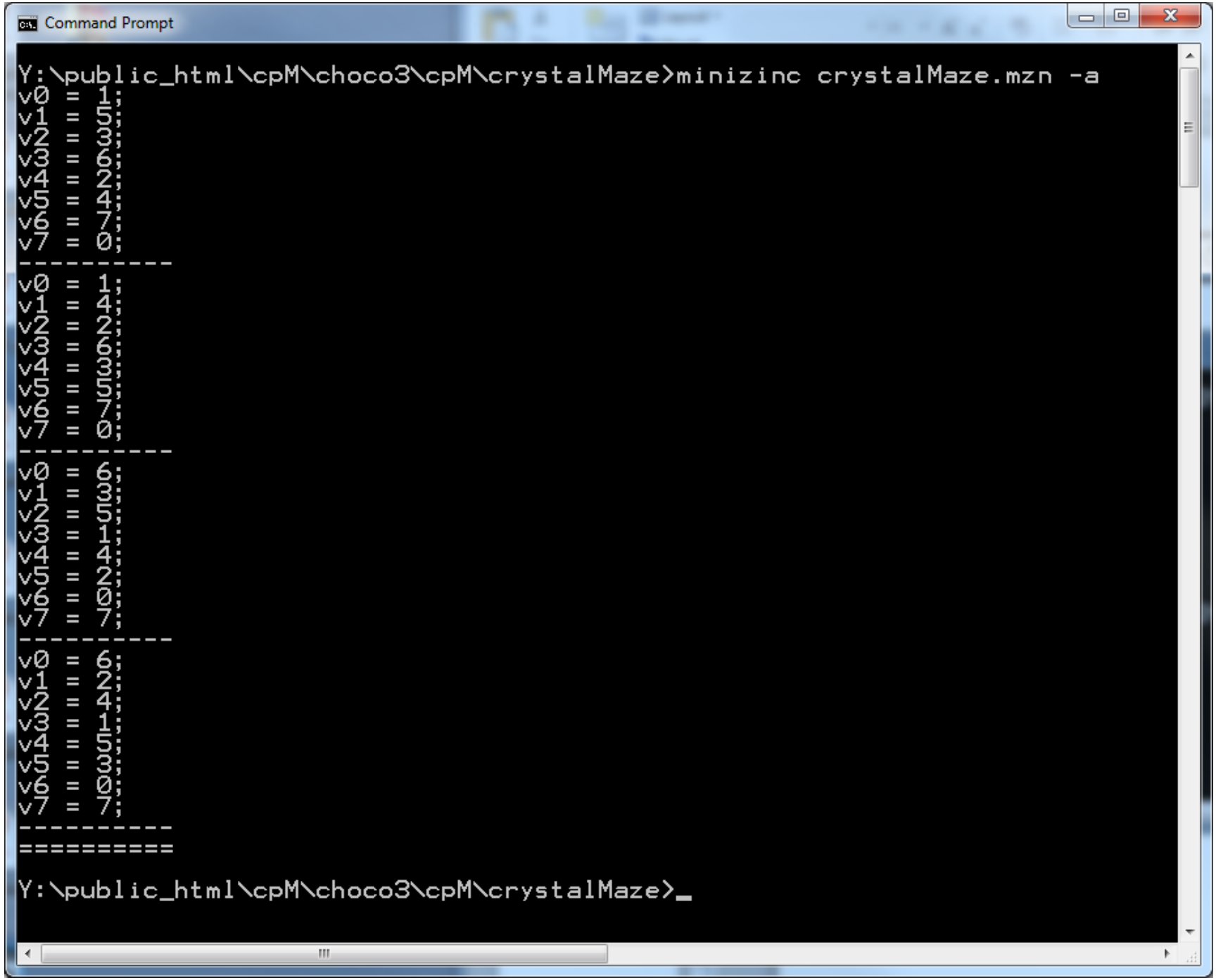
```
Compiling crystalMaze.mzn
Running crystalMaze.mzn
v0 = 1;
v1 = 5;
v2 = 3;
v3 = 6;
v4 = 2;
v5 = 4;
v6 = 7;
v7 = 0;
----------
Finished in 29msec
```

29msec

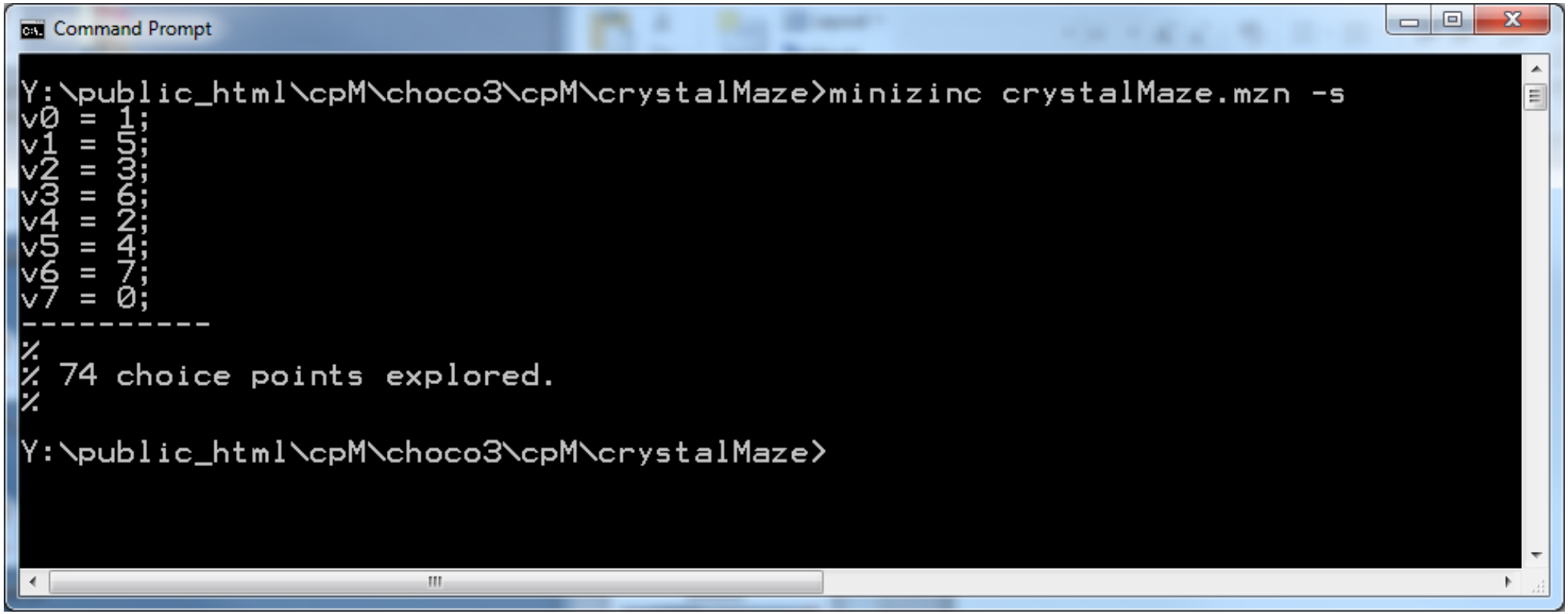
Can use command line



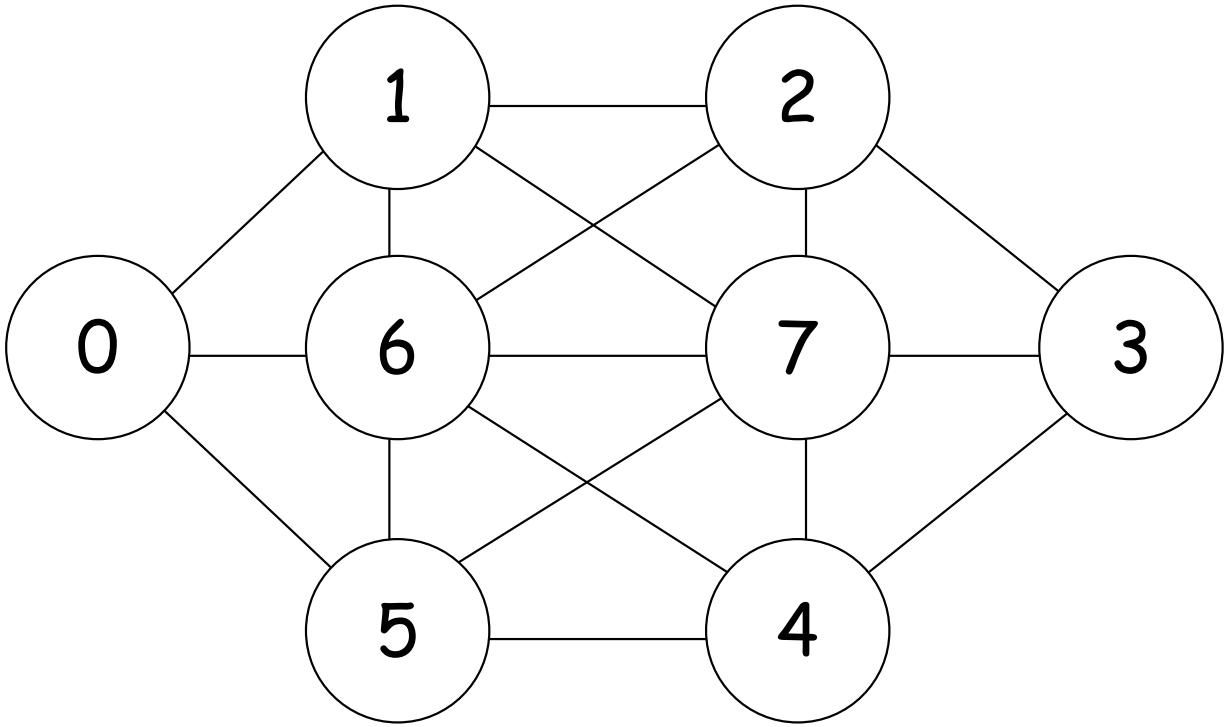
Can get all solutions



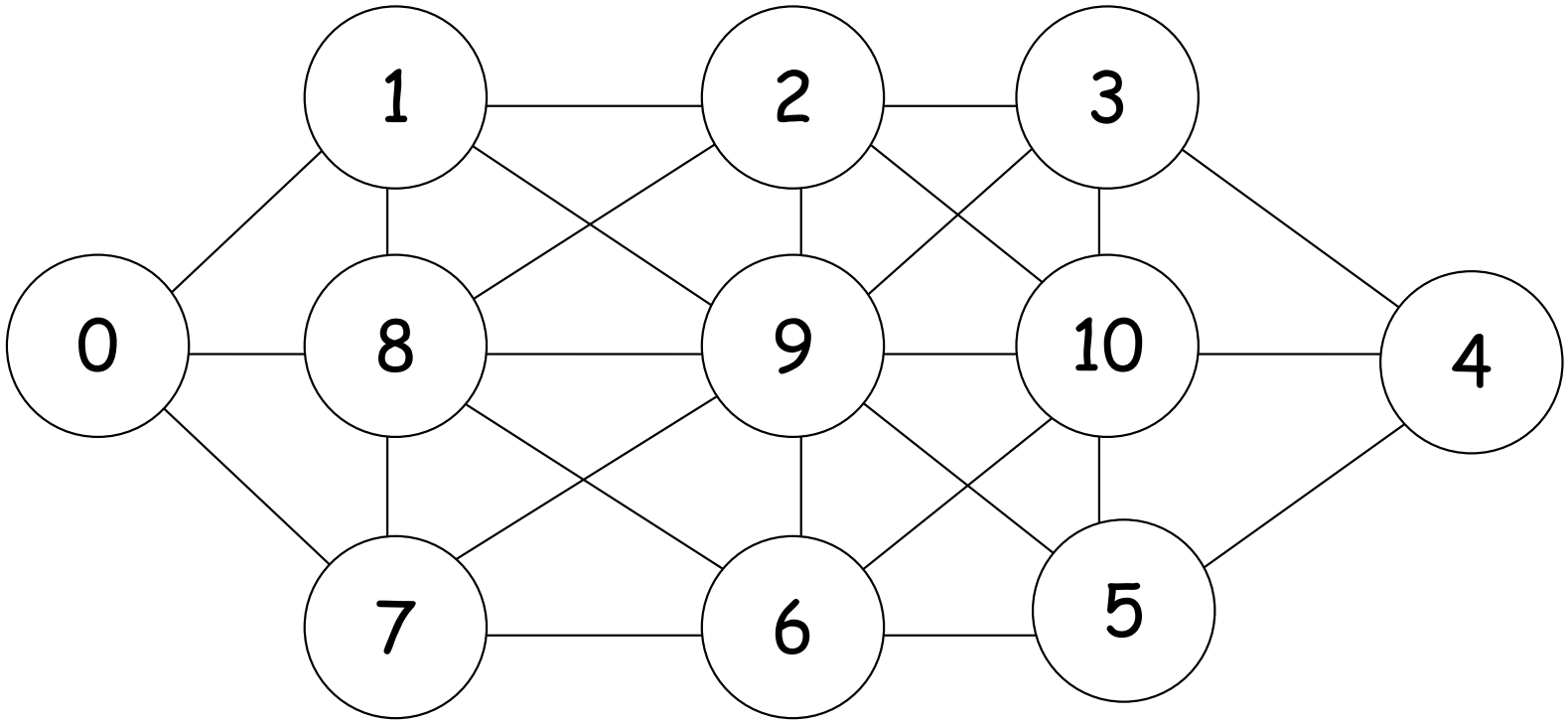
Can get statistics

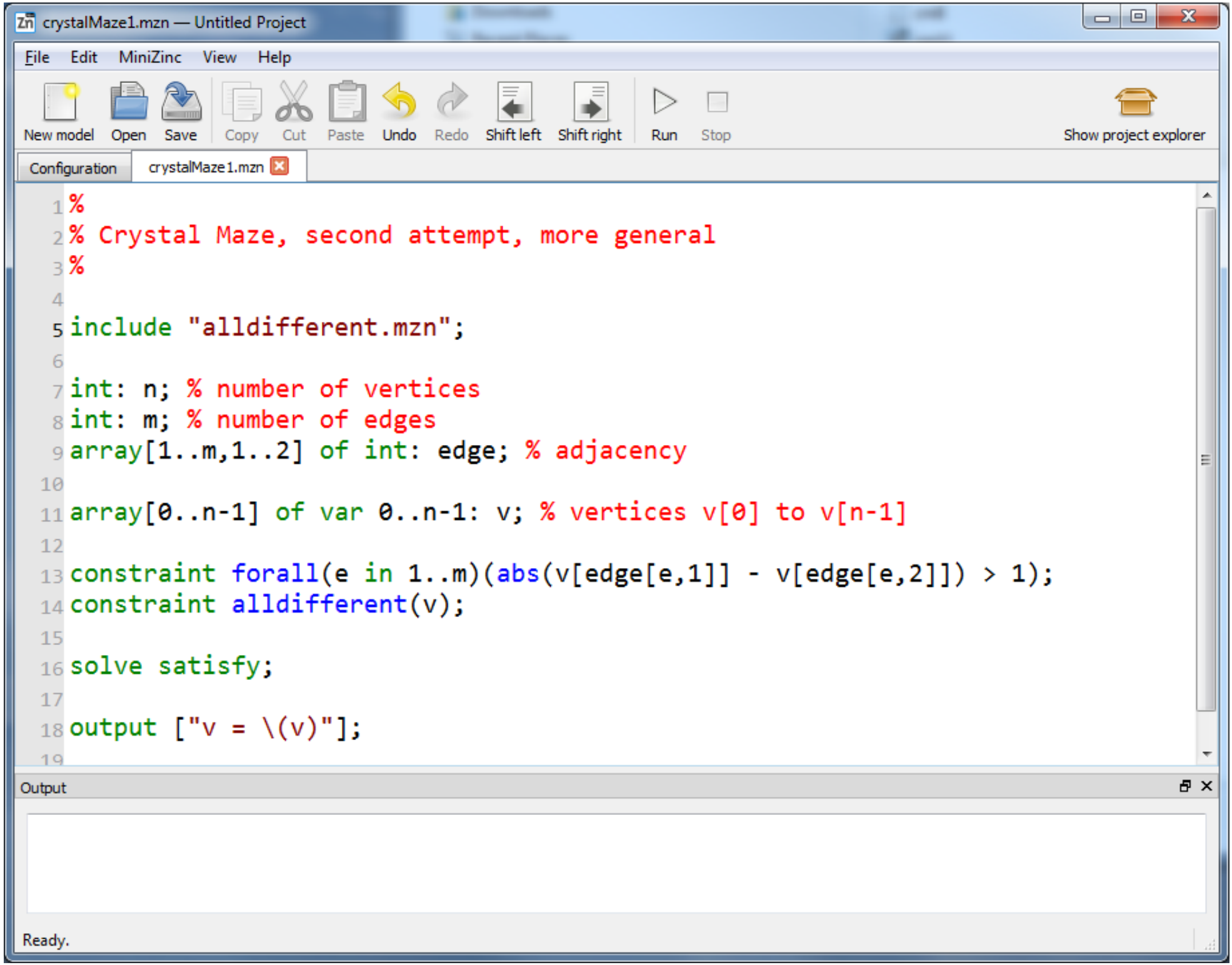


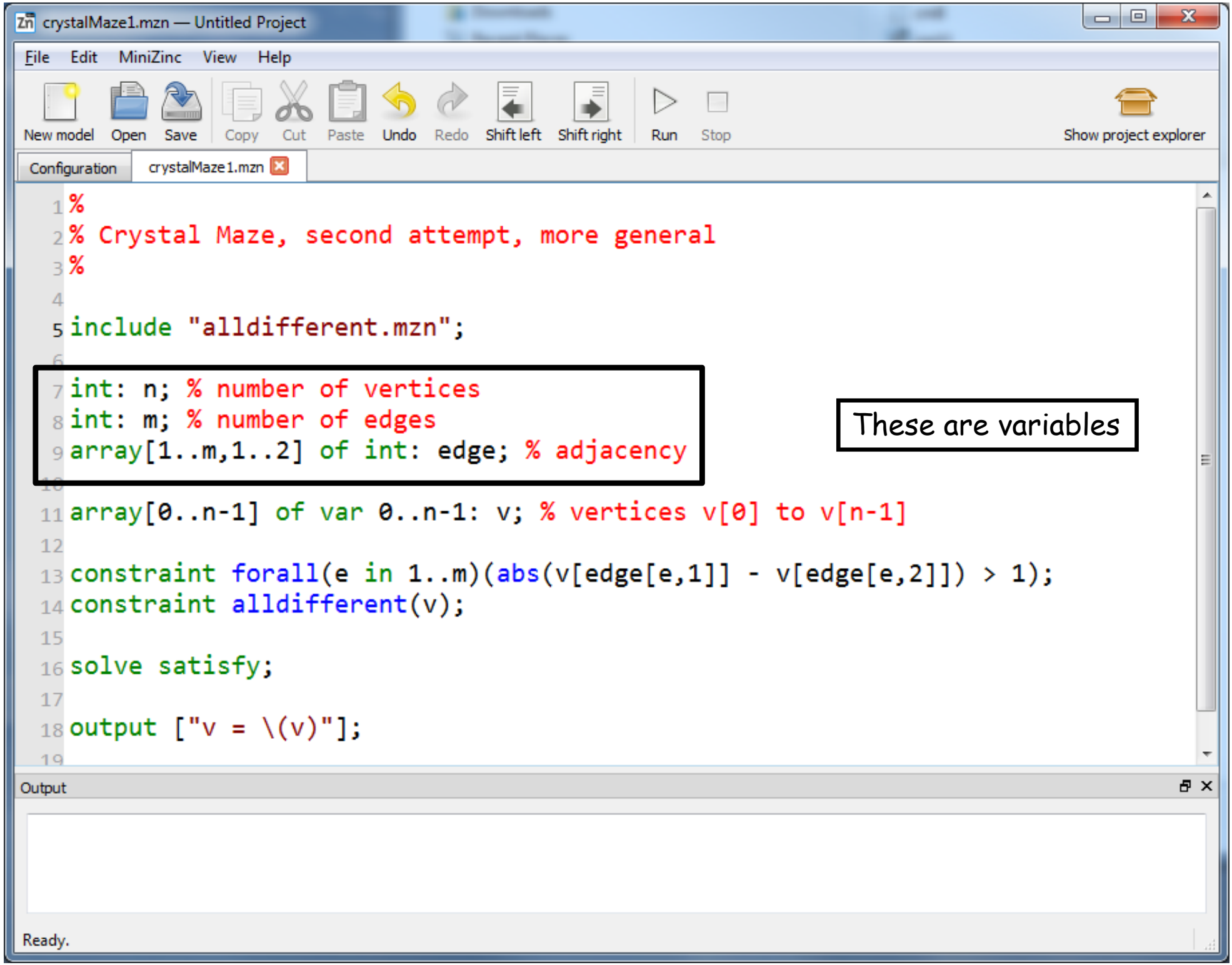
Let's make a more general model

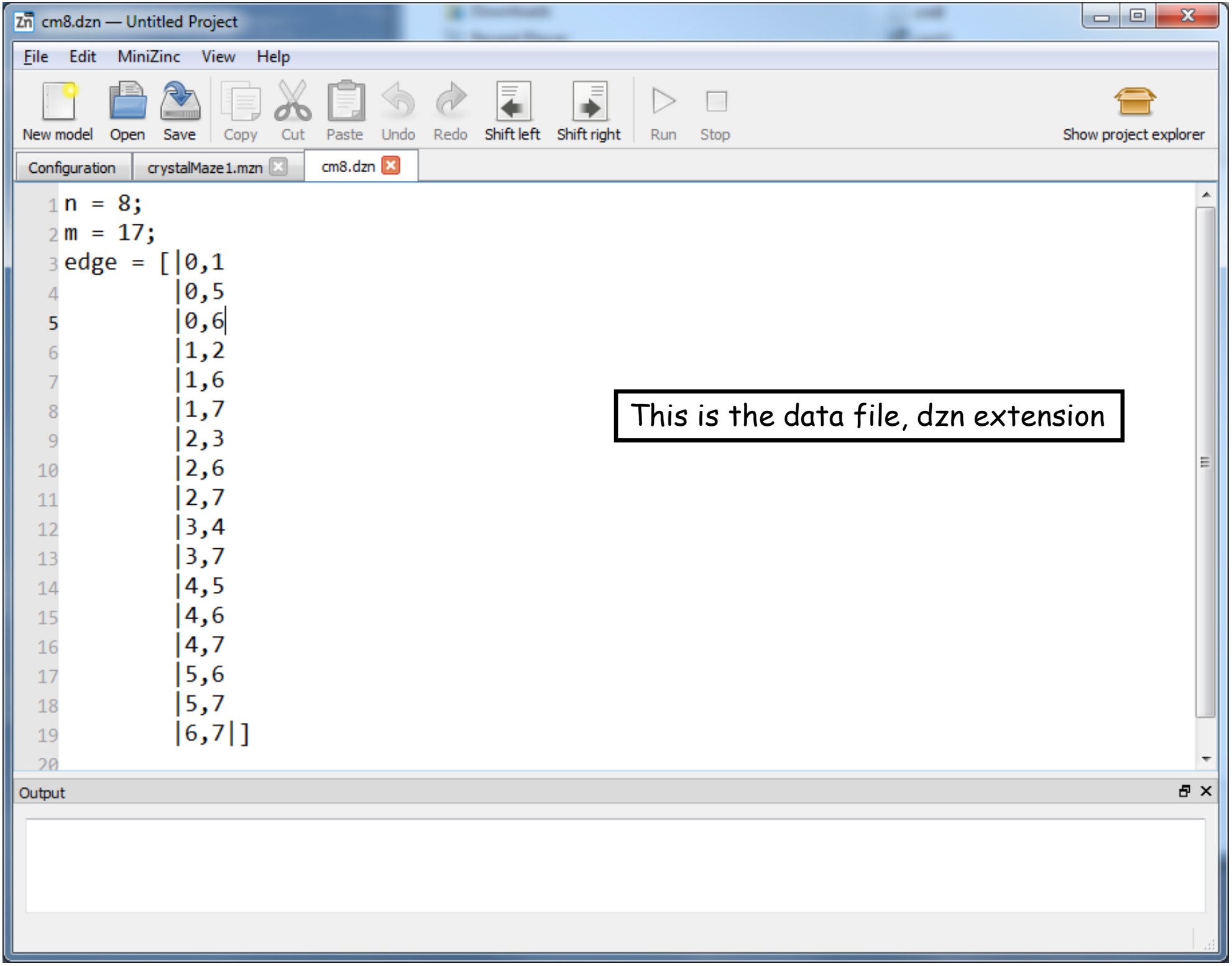


One program to solve both




crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Output
Ready.


crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze1.mzn
1 %
2 % Crystal Maze, second attempt, more general
3 %
4
5 include "alldifferent.mzn";
6
7 int: n; % number of vertices
8 int: m; % number of edges
9 array[1..m,1..2] of int: edge; % adjacency
10
11 array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
12
13 constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
14 constraint alldifferent(v);
15
16 solve satisfy;
17
18 output ["v = \(v)"];
19
These are variables
Output
Ready.


cm8.dzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze1.mzn cm8.dzn
n = 8;
m = 17;
edge = [|0,1
|0,5
|0,6
|1,2
|1,6
|1,7
|2,3
|2,6
|2,7
|3,4
|3,7
|4,5
|4,6
|4,7
|5,6
|5,7
|6,7|]
This is the data file, dzn extension
Output

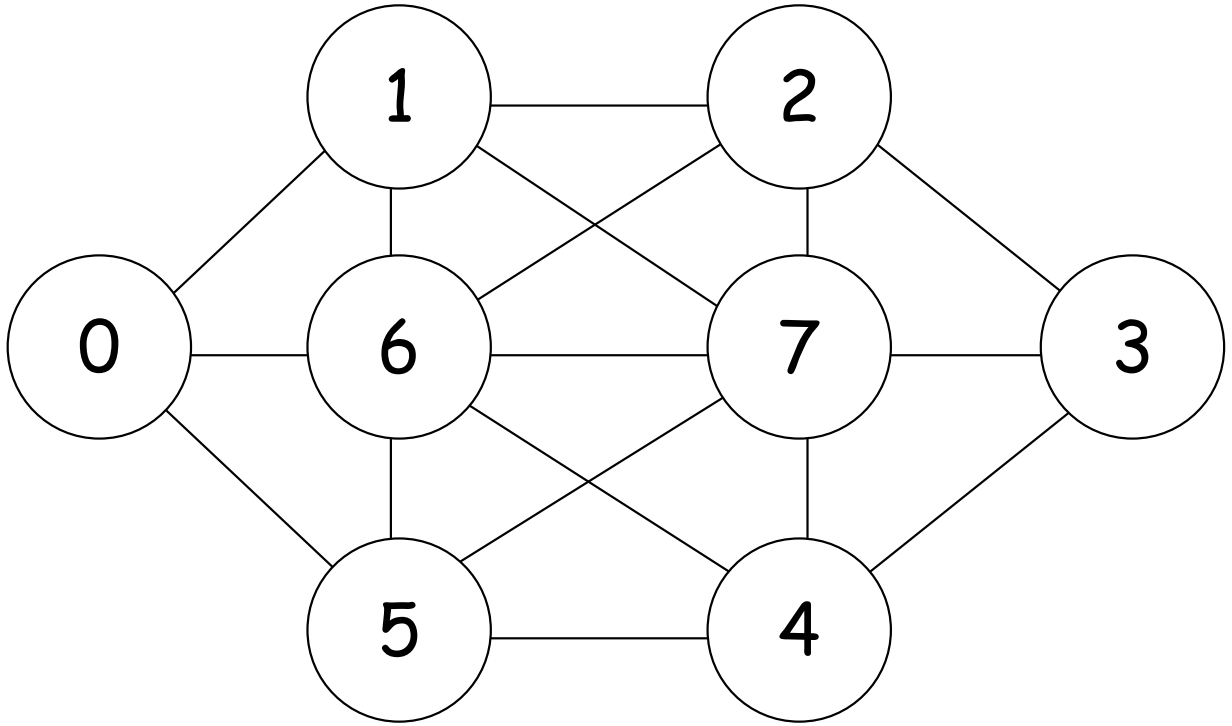
cm8.dzn — Untitled Project

File Edit MiniZinc View Help

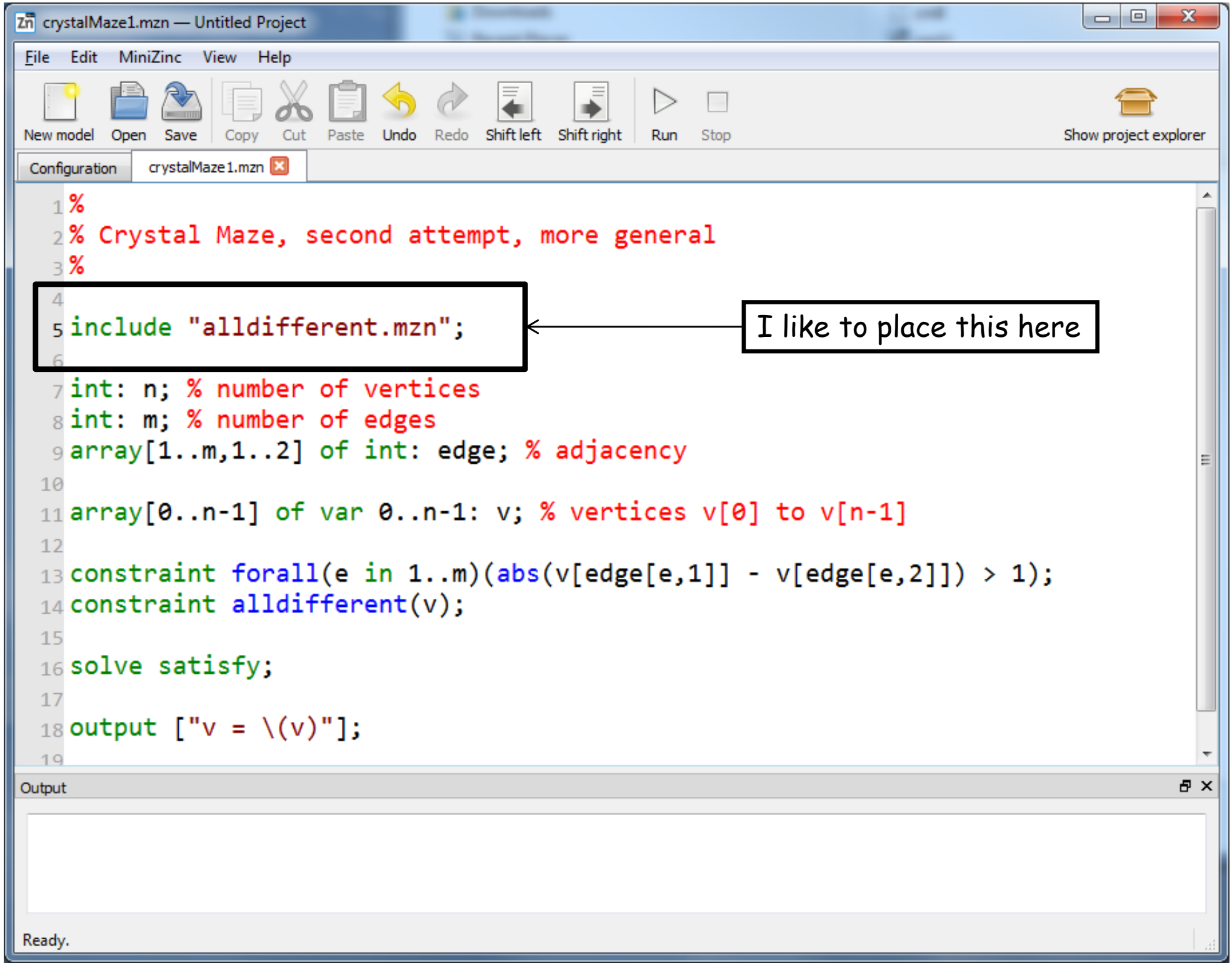
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer

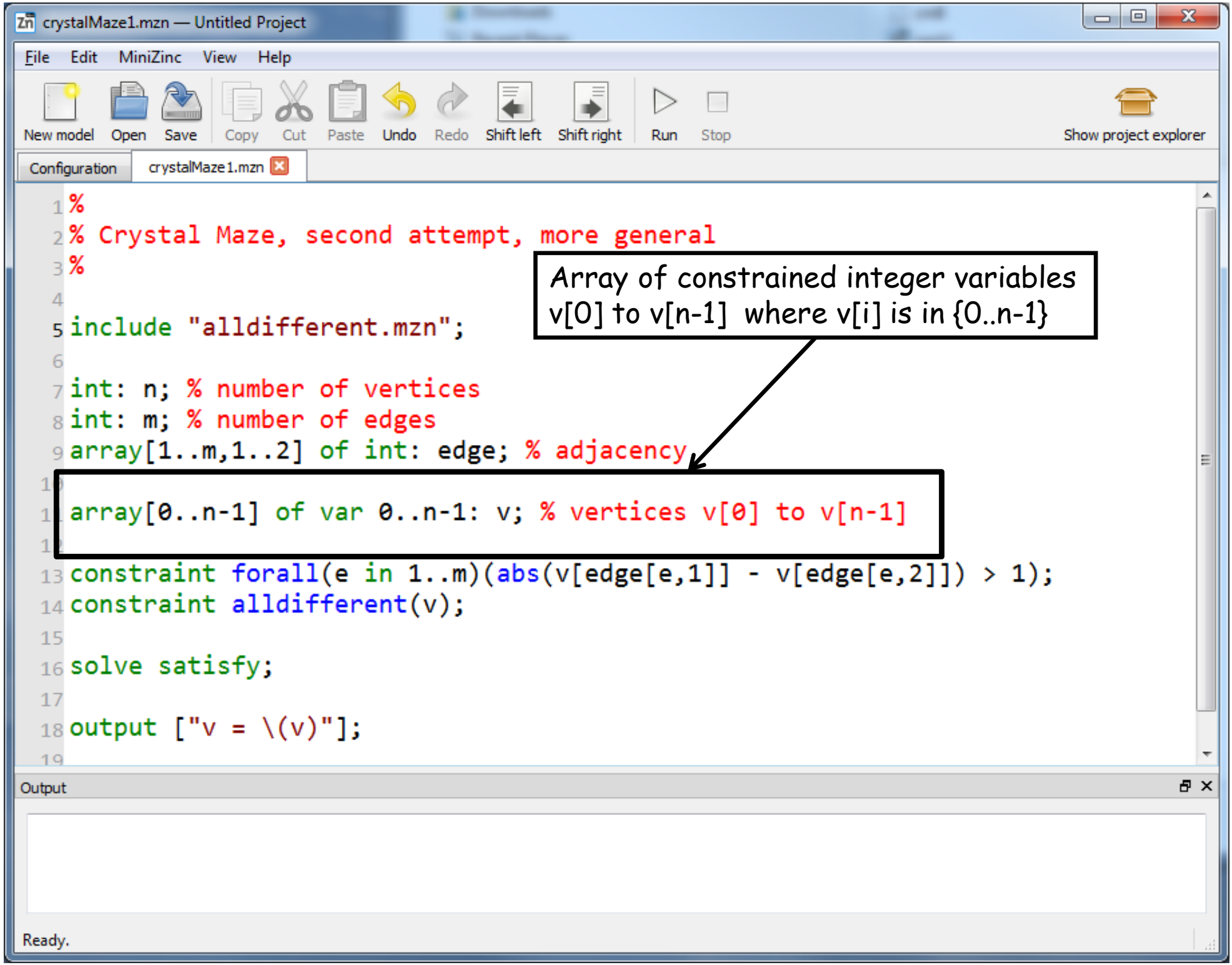
Configuration | crystalMaze1.mzn | cm8.dzn

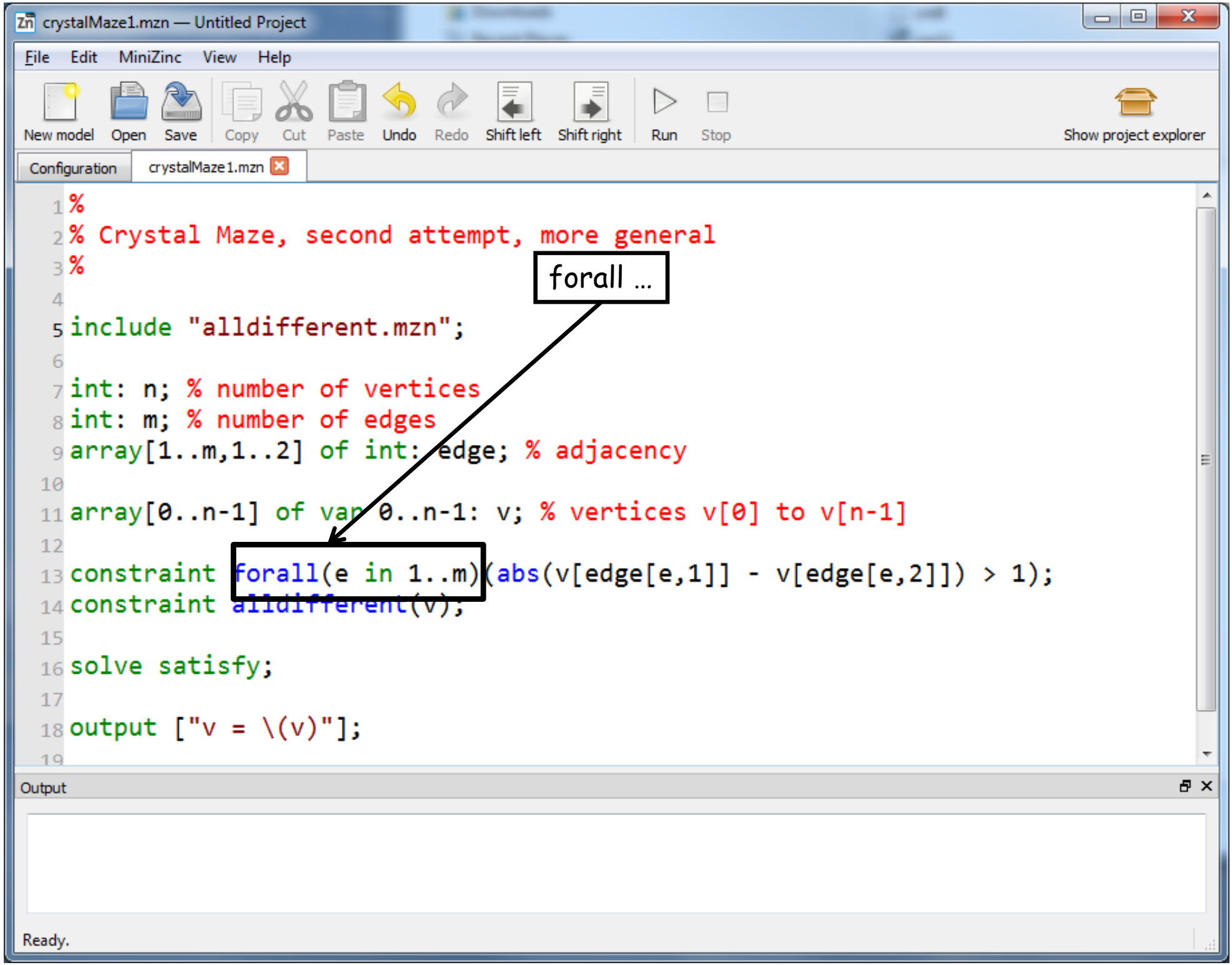
```
1  n = 8;
2  m = 17;
3  edge = [|0,1
4          |0,5
5          |0,6
6          |1,2
7          |1,6
8          |1,7
9          |2,3
10         |2,6
11         |2,7
12         |3,4
13         |3,7
14         |4,5
15         |4,6
16         |4,7
17         |5,6
18         |5,7
19         |6,7|]
20
```

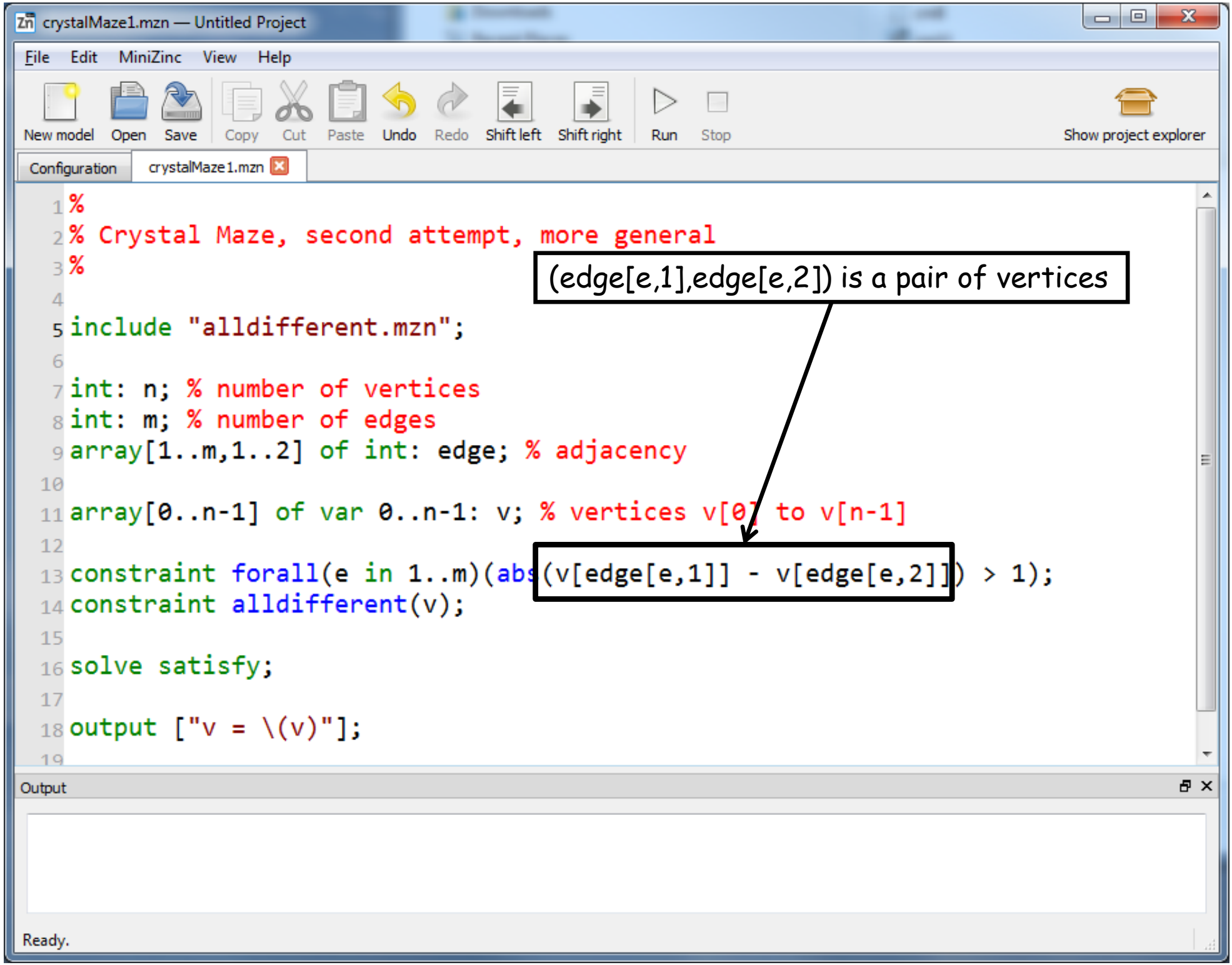


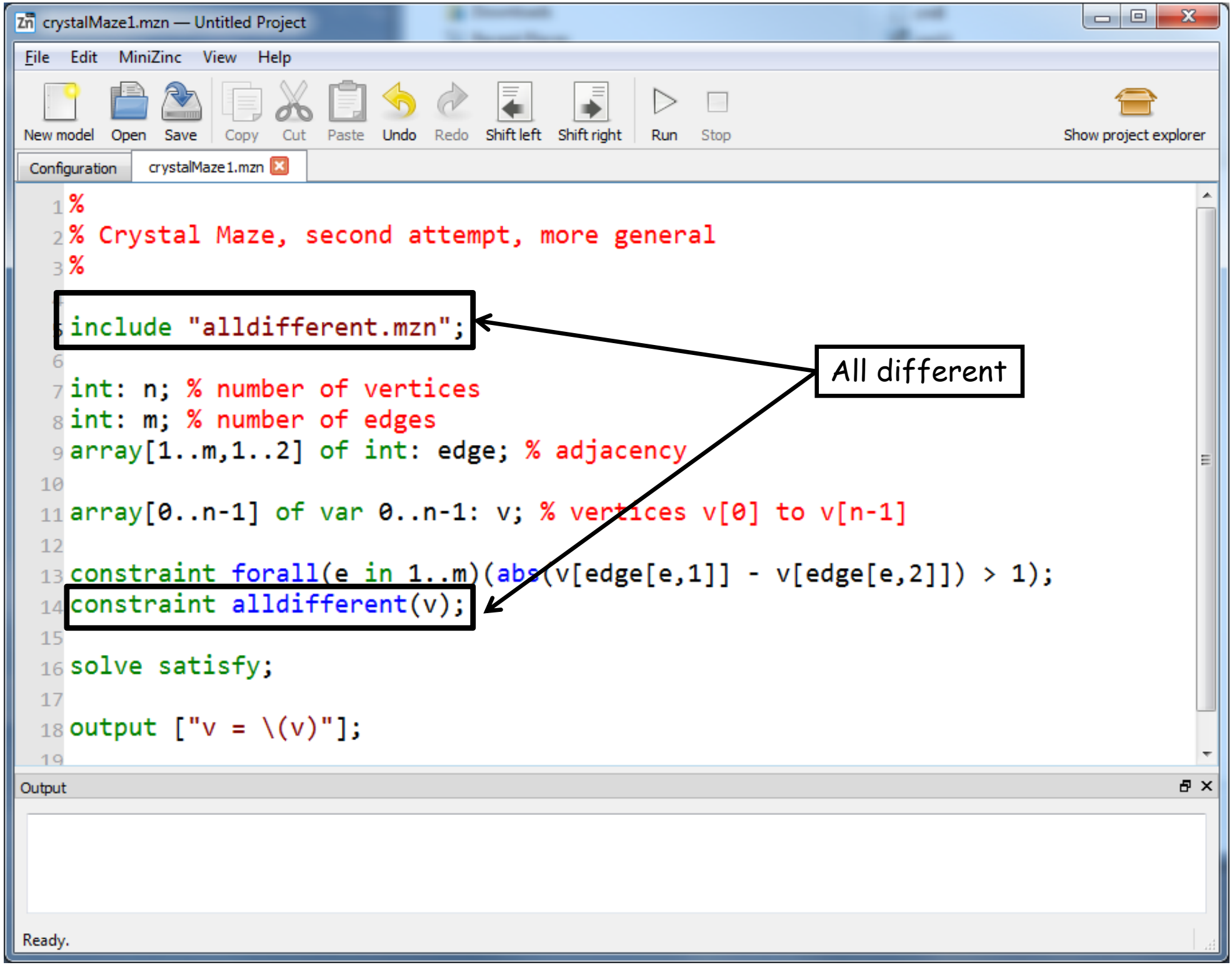
Output

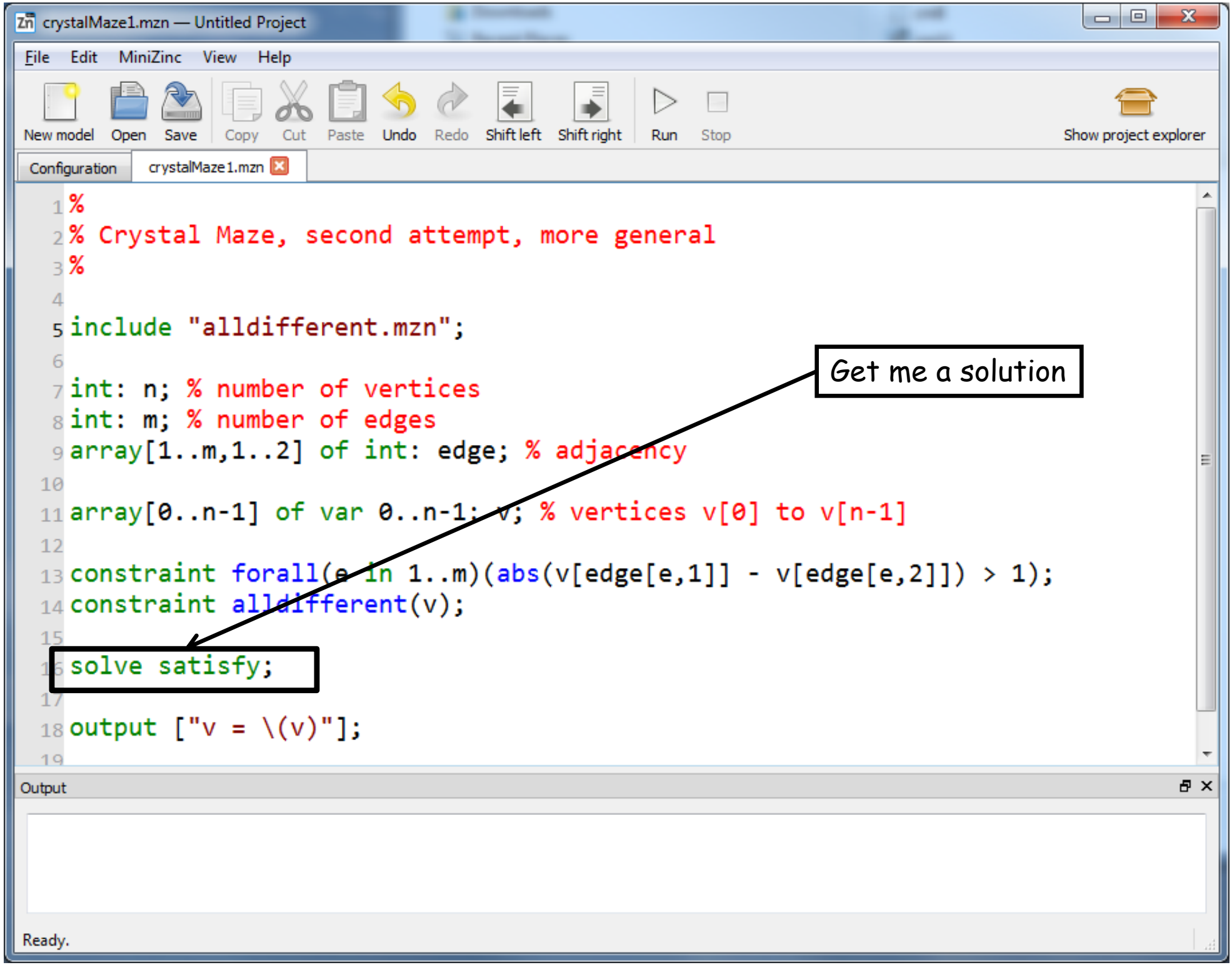

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
I like to place this here
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Output
Ready.

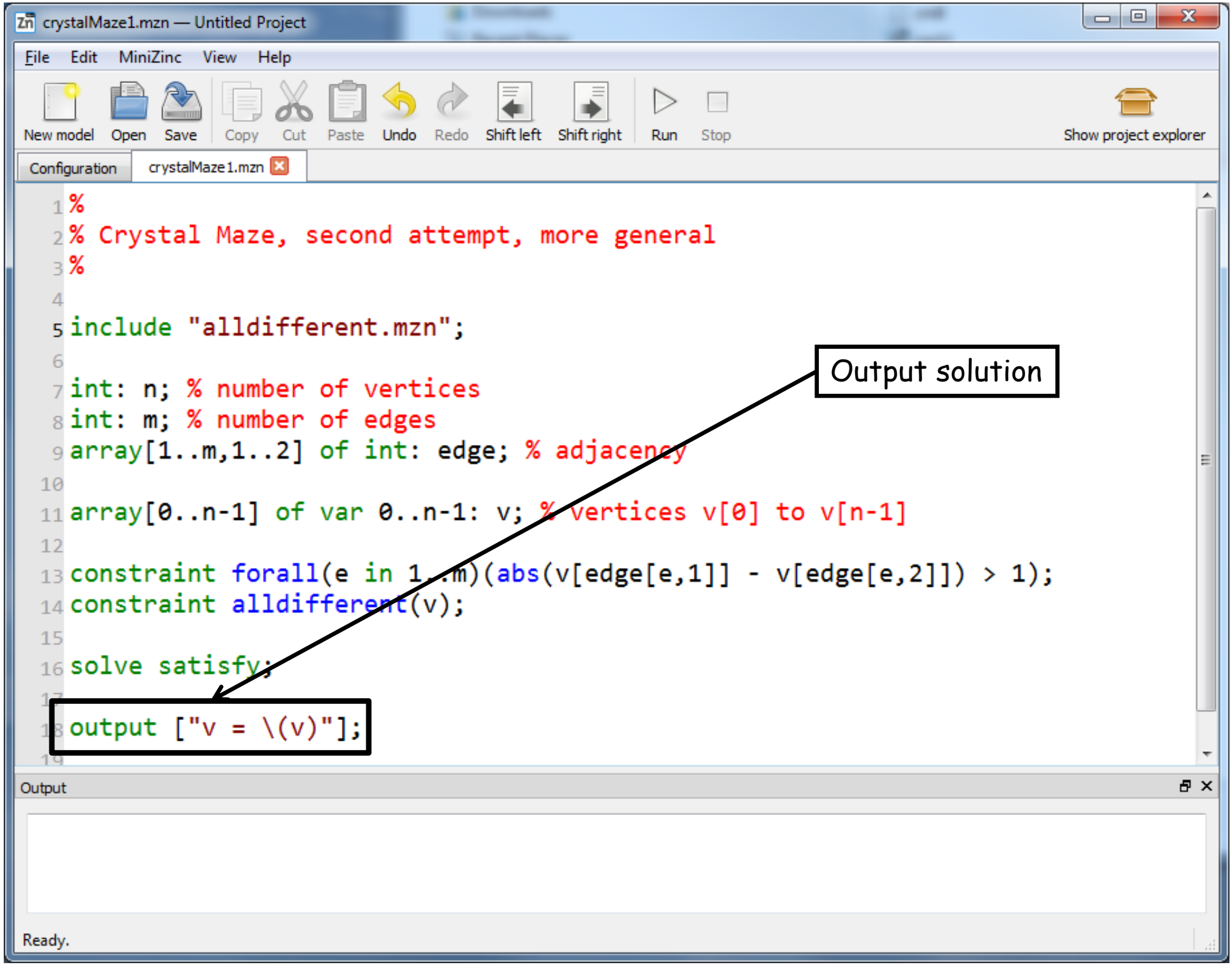

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Array of constrained integer variables
v[0] to v[n-1] where v[i] is in {0..n-1}
Output
Ready.

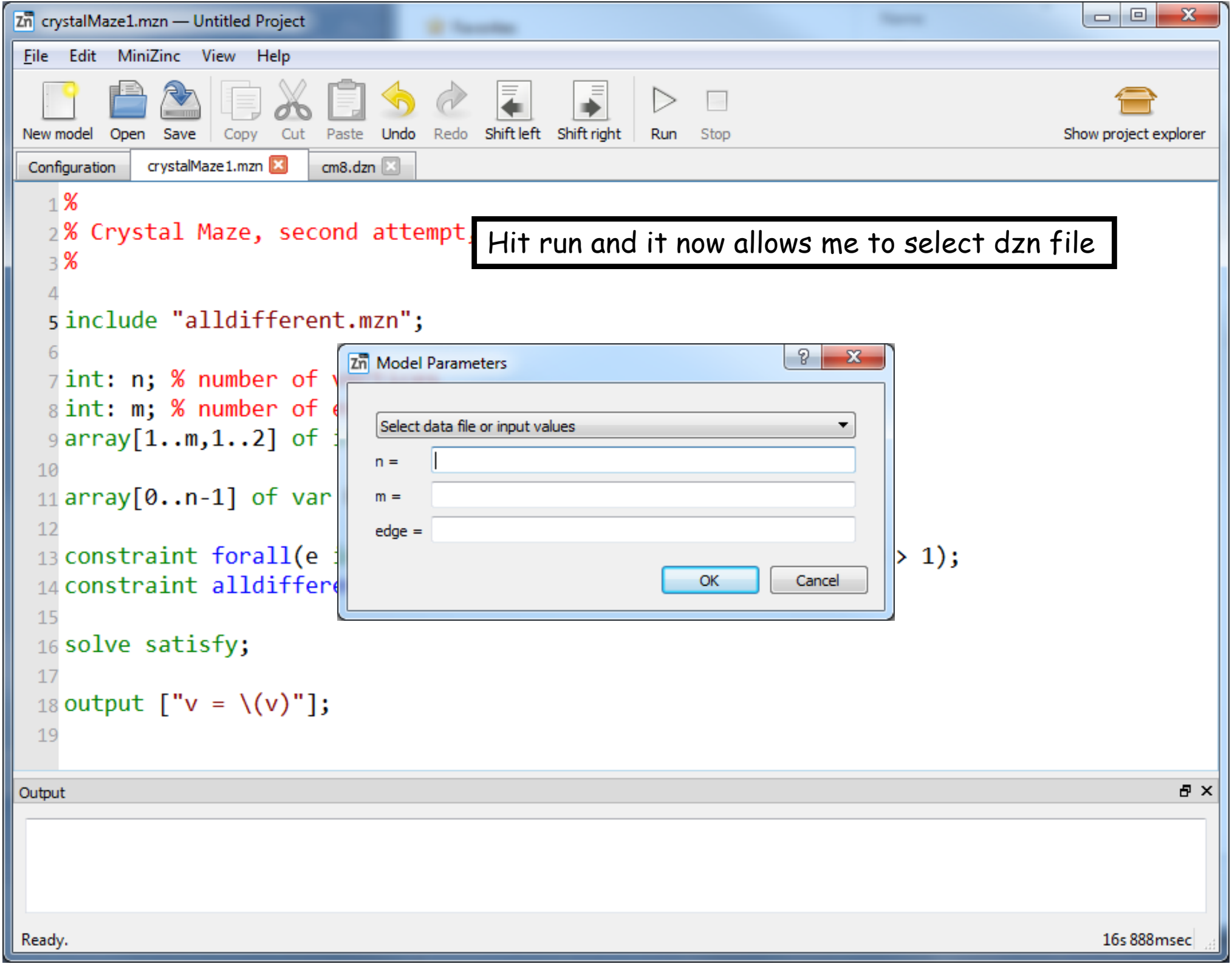

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
forall ...
Output
Ready.

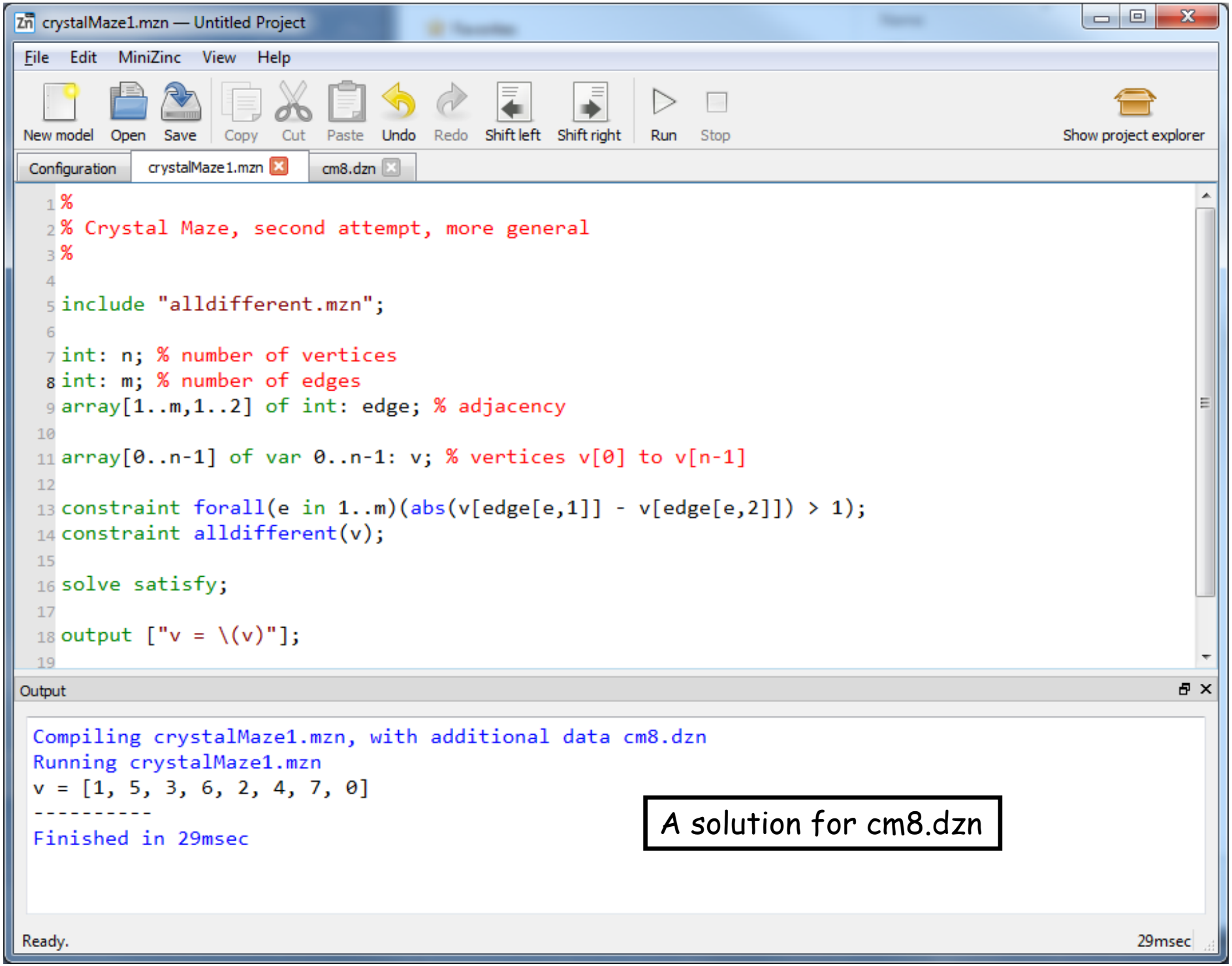

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
(edge[e,1],edge[e,2]) is a pair of vertices
Output
Ready.


crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
All different
Output
Ready.


crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Get me a solution
Output
Ready.

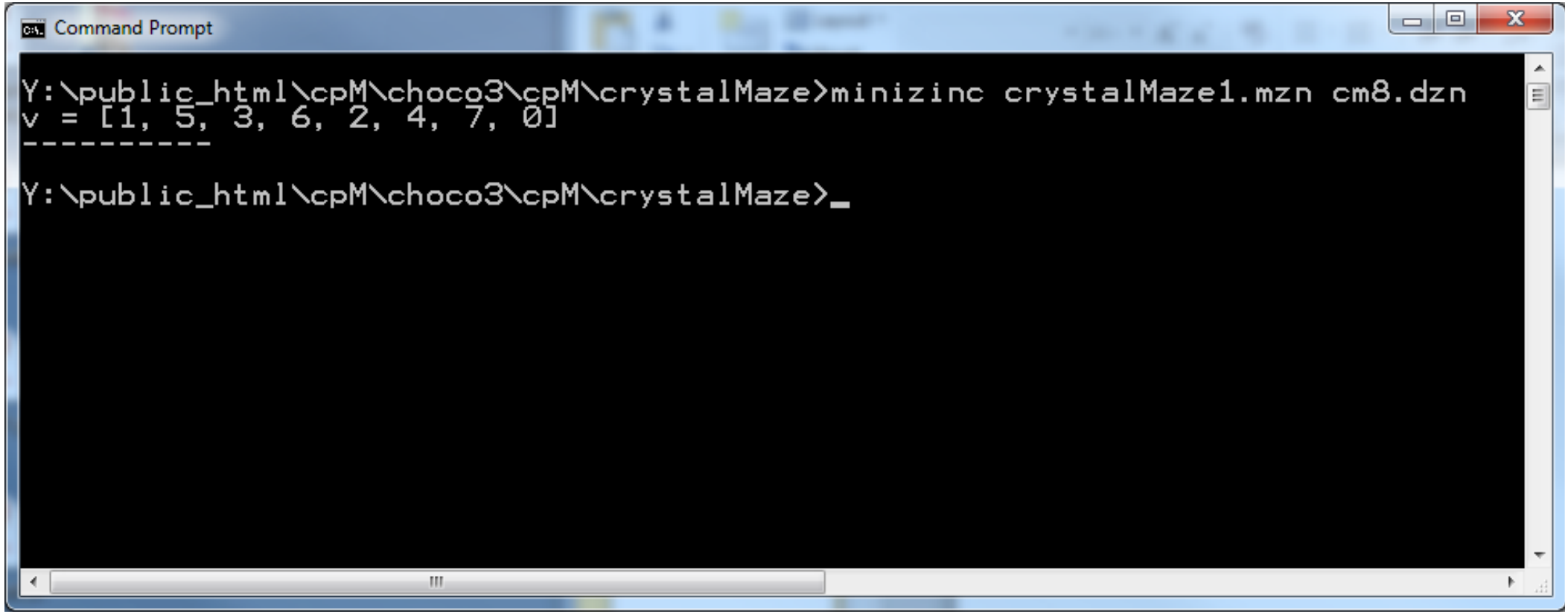

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration
crystalMaze1.mzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Output solution
Output
Ready.


crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze1.mzn cm8.dzn
% Crystal Maze, second attempt
%
include "alldifferent.mzn";
int: n; % number of
int: m; % number of
array[1..m,1..2] of
array[0..n-1] of var
constraint forall(e
constraint alldiffere
> 1);
solve satisfy;
output ["v = \(v)"];
Hit run and it now allows me to select dzn file
Model Parameters
Select data file or input values
n =
m =
edge =
OK
Cancel
Output
Ready.
16s 888msec

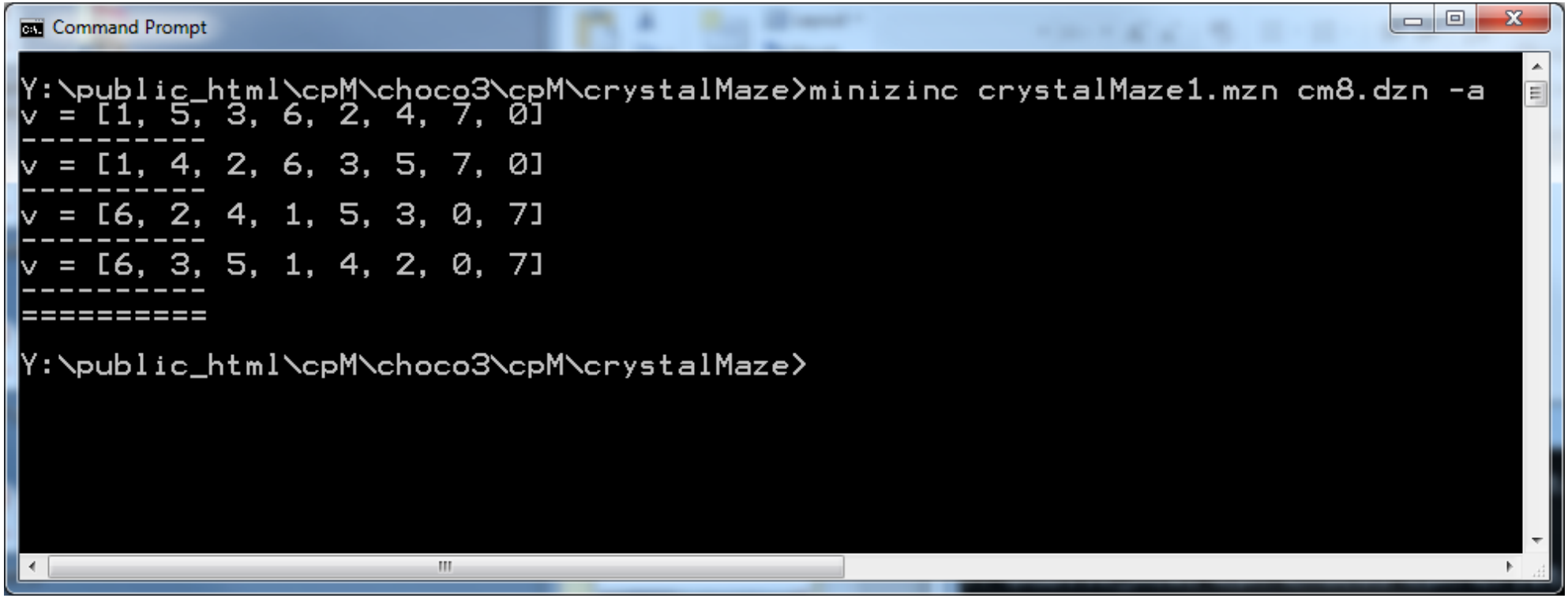

crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze1.mzn cm8.dzn
%
% Crystal Maze, second attempt, more general
%
include "alldifferent.mzn";
int: n; % number of vertices
int: m; % number of edges
array[1..m,1..2] of int: edge; % adjacency
array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
constraint alldifferent(v);
solve satisfy;
output ["v = \(v)"];
Output
Compiling crystalMaze1.mzn, with additional data cm8.dzn
Running crystalMaze1.mzn
v = [1, 5, 3, 6, 2, 4, 7, 0]

Finished in 29msec
A solution for cm8.dzn
Ready.
29msec

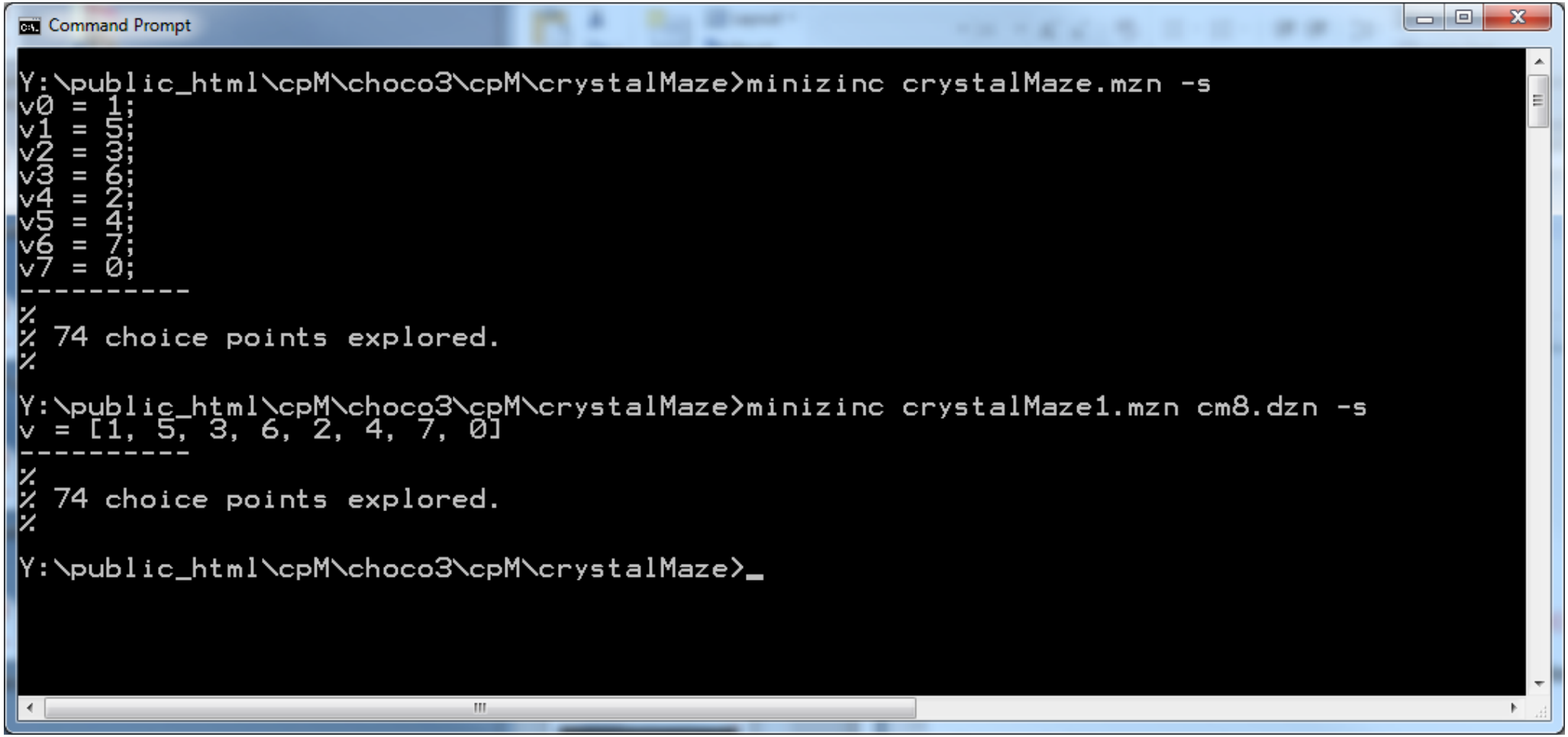
Also on command line



Also on command line (all solutions)



Compare two models … cool



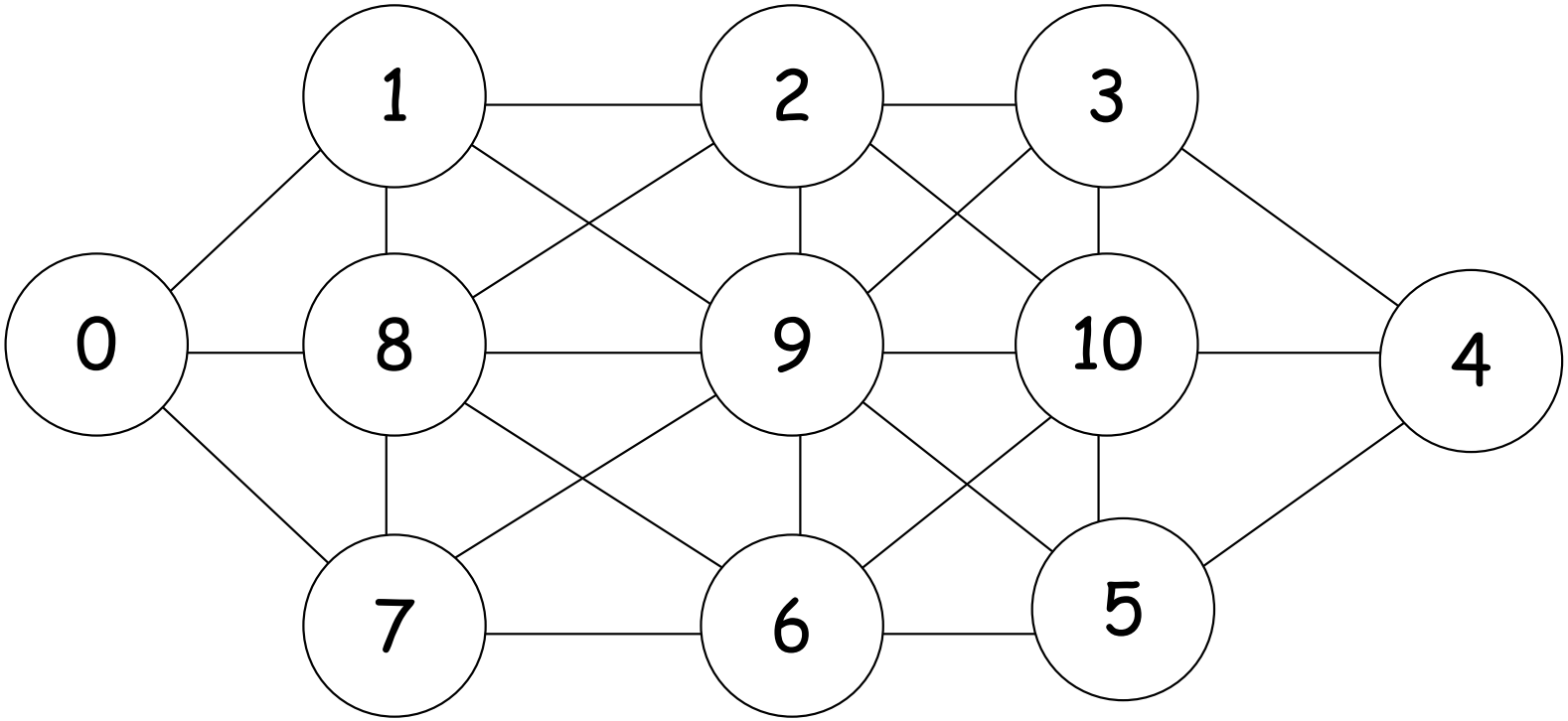
cm11.dzn — Untitled Project

File Edit MiniZinc View Help

New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer

Configuration | crystalMaze1.mzn | cm8.dzn | cm11.dzn

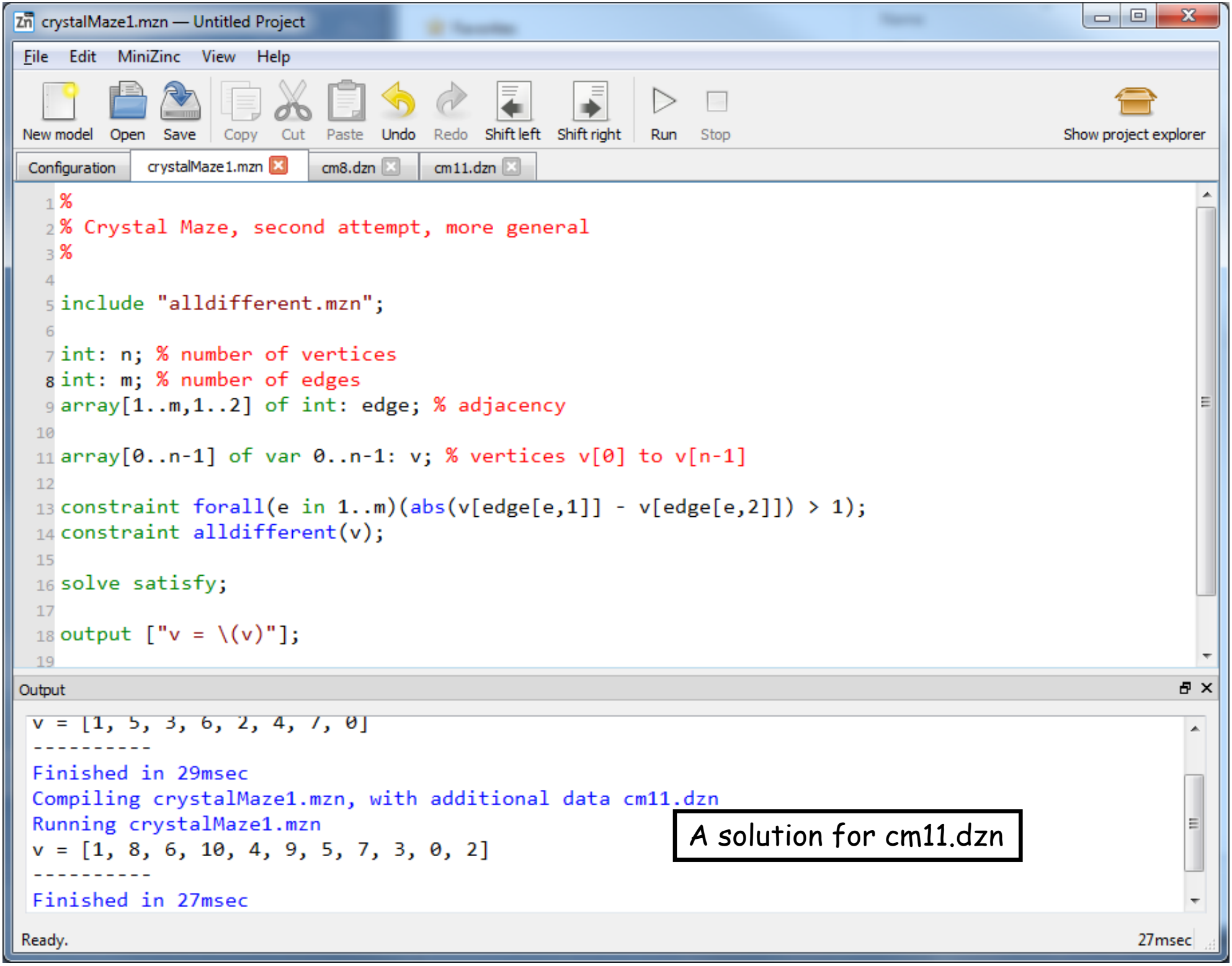
```
n = 11;
m = 26;
edge = [|0,1
        |0,8
        |0,7
        |1,2
        |1,9
        |1,8
        |2,3
        |2,10
        |2,9
        |2,8
        |3,4
        |3,10
        |3,9
        |4,10
        |4,5
        |5,10
        |5,9
        |5,6
        |6,10
        |6,9
        |6,8
        |6,7
        |7,9
        |7,8
        |8,9
        |9,10|]
```



Output

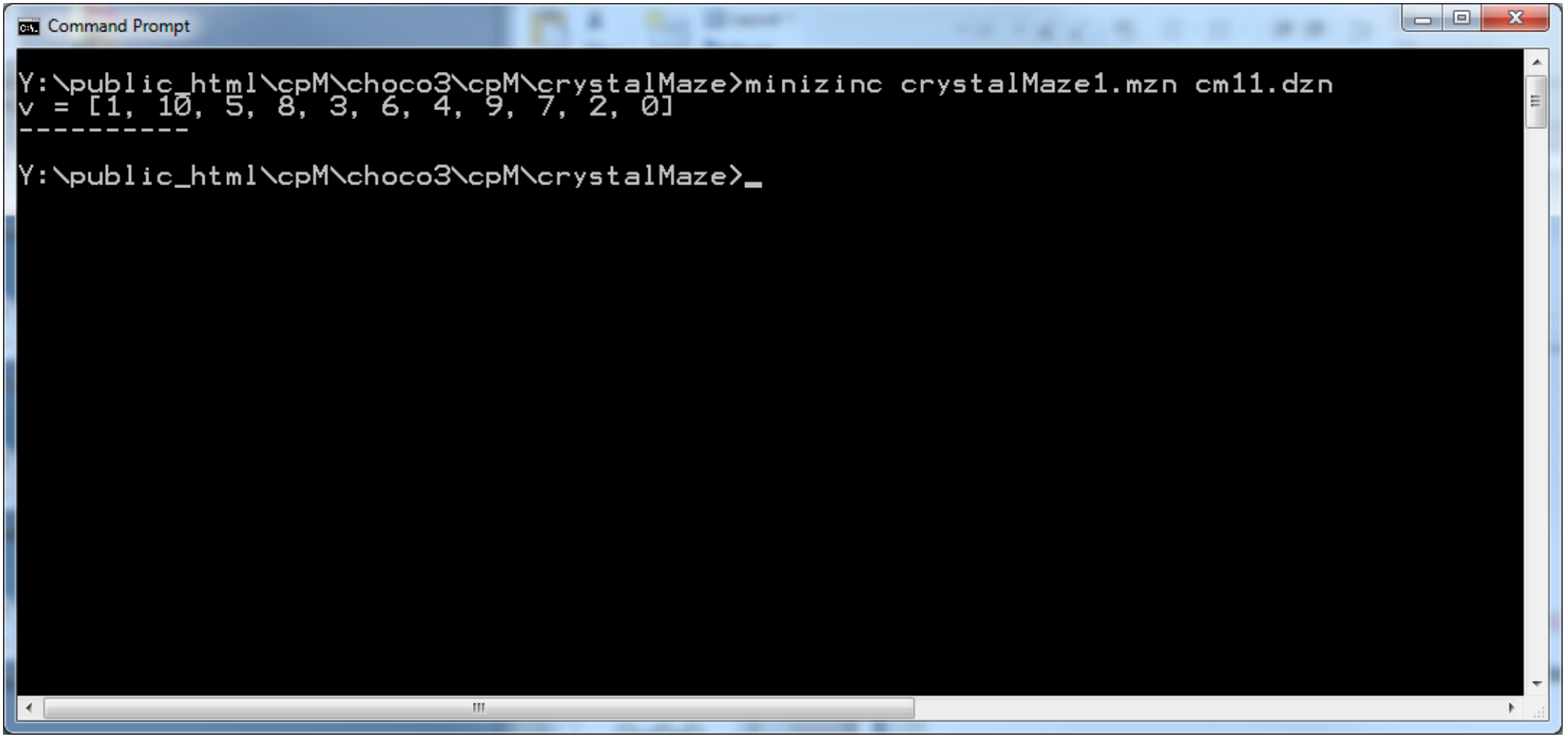
```
----------
Finished in 29msec
Compiling crystalMaze1.mzn, with additional data cm11.dzn
Running crystalMaze1.mzn
```

Ready. 27msec


crystalMaze1.mzn — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run Stop Show project explorer
Configuration crystalMaze1.mzn cm8.dzn cm11.dzn
1 %
2 % Crystal Maze, second attempt, more general
3 %
4
5 include "alldifferent.mzn";
6
7 int: n; % number of vertices
8 int: m; % number of edges
9 array[1..m,1..2] of int: edge; % adjacency
10
11 array[0..n-1] of var 0..n-1: v; % vertices v[0] to v[n-1]
12
13 constraint forall(e in 1..m)(abs(v[edge[e,1]] - v[edge[e,2]]) > 1);
14 constraint alldifferent(v);
15
16 solve satisfy;
17
18 output ["v = \(v)"];
19
Output
v = [1, 5, 3, 6, 2, 4, 7, 0]

Finished in 29msec
Compiling crystalMaze1.mzn, with additional data cm11.dzn
Running crystalMaze1.mzn
v = [1, 8, 6, 10, 4, 9, 5, 7, 3, 0, 2]

Finished in 27msec
A solution for cm11.dzn
Ready.
27msec


Command Prompt
Y:\public_html\cpM\choco3\cpM\crystalMaze>minizinc crystalMaze1.mzn cm11.dzn
v = [1, 10, 5, 8, 3, 6, 4, 9, 7, 2, 0]

Y:\public_html\cpM\choco3\cpM\crystalMaze>

So, what IS a constraint program?

Possible answers

It's a program that generates variables and constraints to represent a problem

It's a program that creates a model of a problem and then uses search and heuristics to solve the problem

It's a program that compiles some problem into a representation as CSP


That's all Folks!