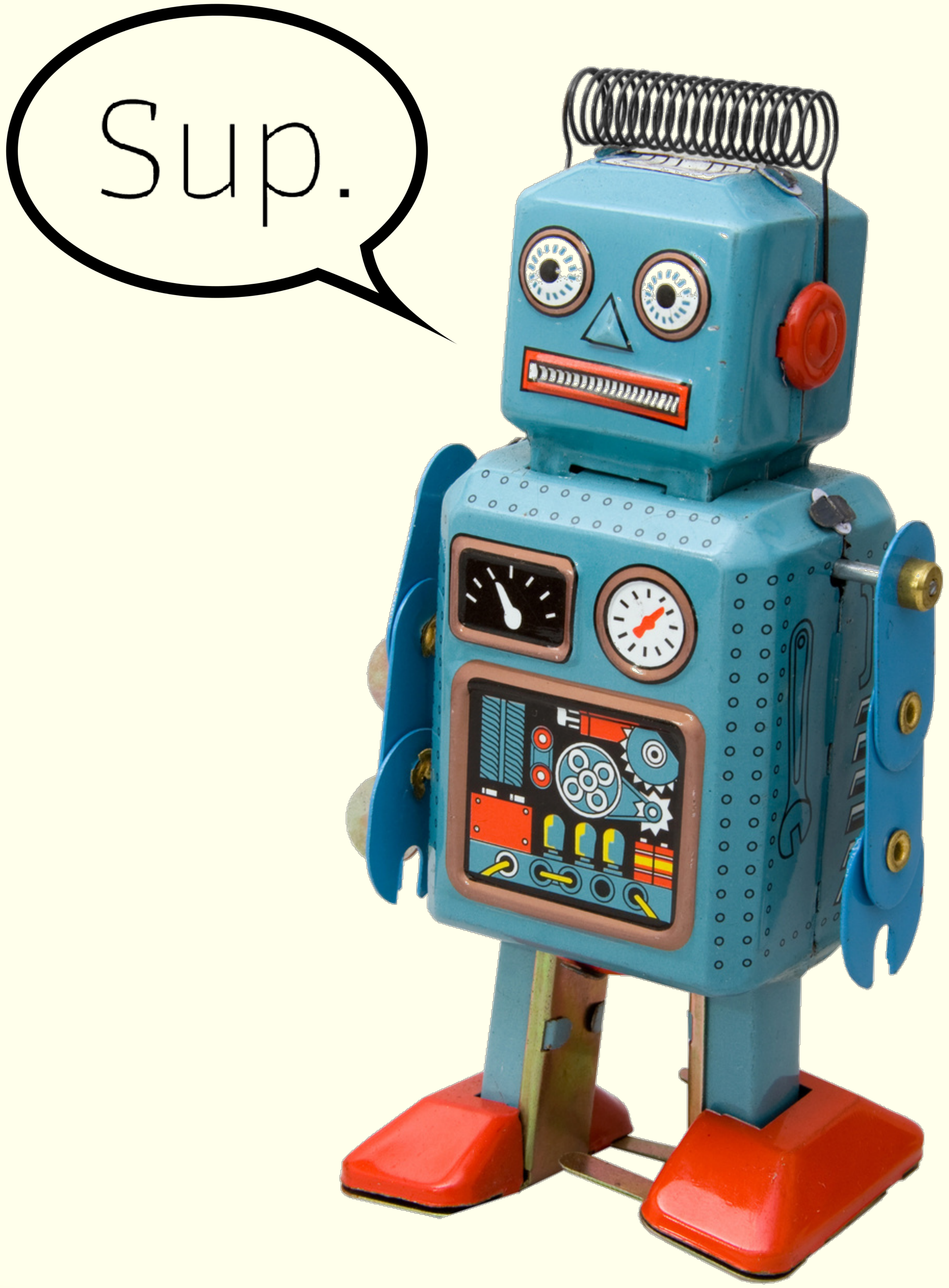
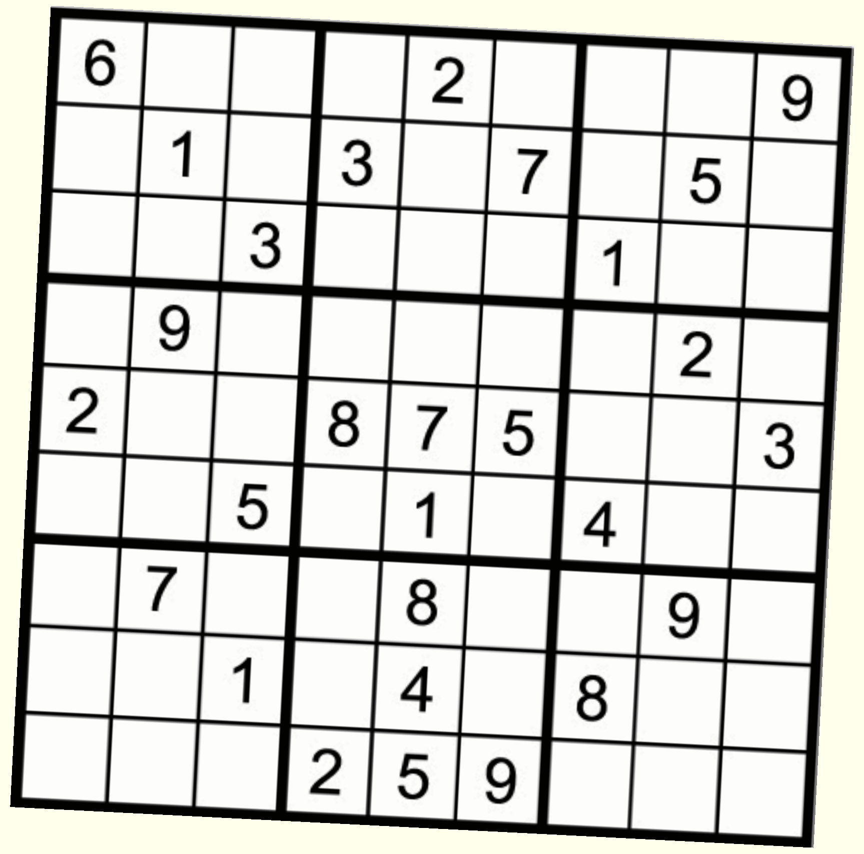
Hello, Sentient

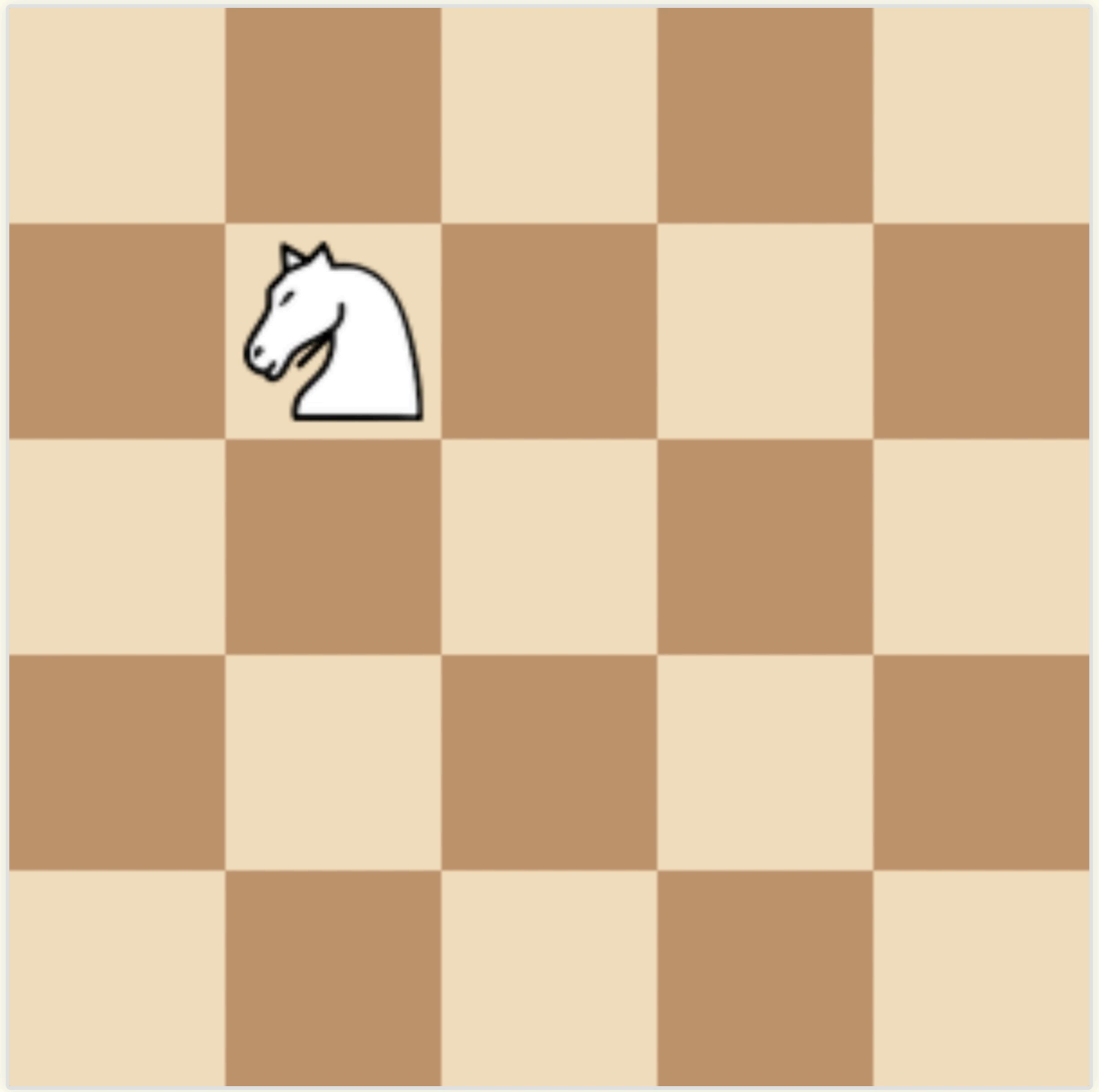


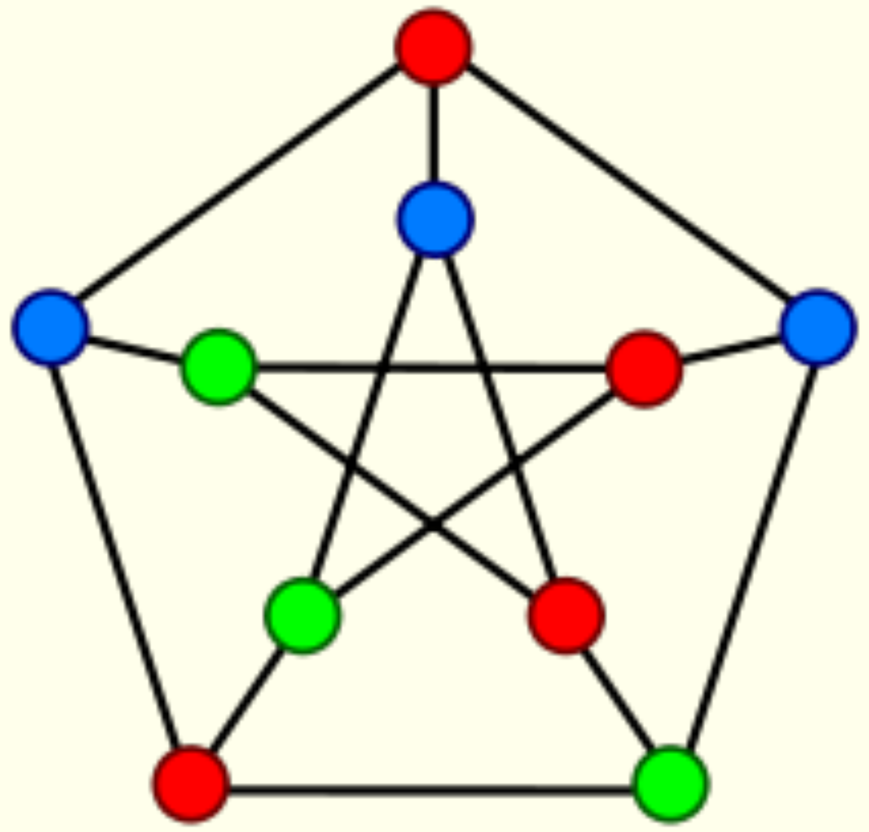
by Chris Patuzzo
Ember London, 2017-01-12
@cpatuzzo

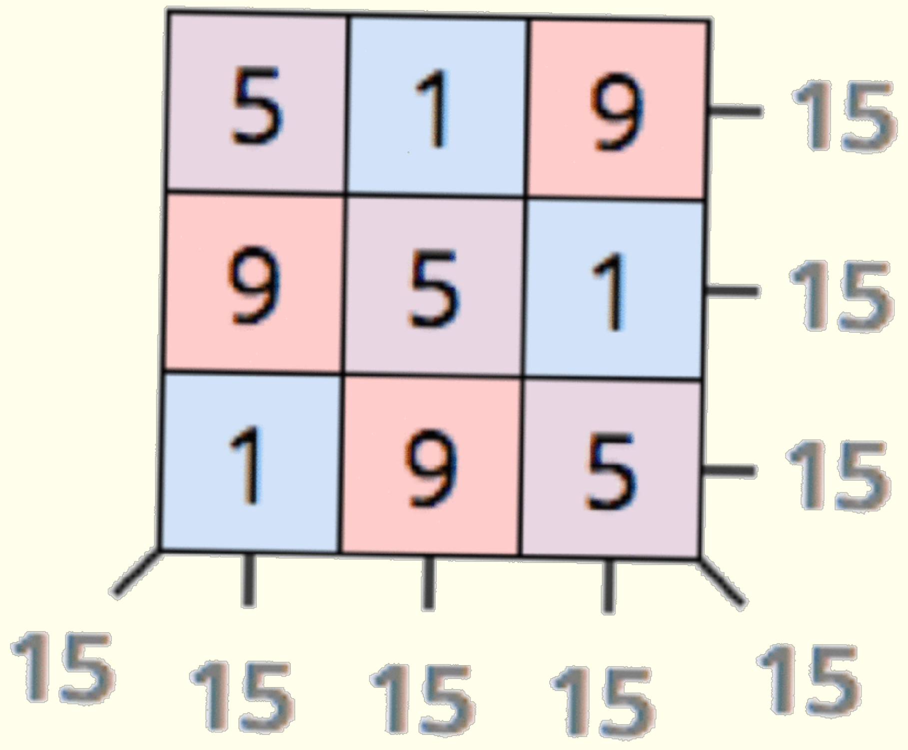
Firstly,

What is Sentient?









Most languages

6				2				9
	1		3		7		5	
		3				1		
	9						2	
2			8	7	5			3
		5		1		4		
	7			8			9	
		1		4		8		
			2	5	9			

Some algorithm

For each empty square

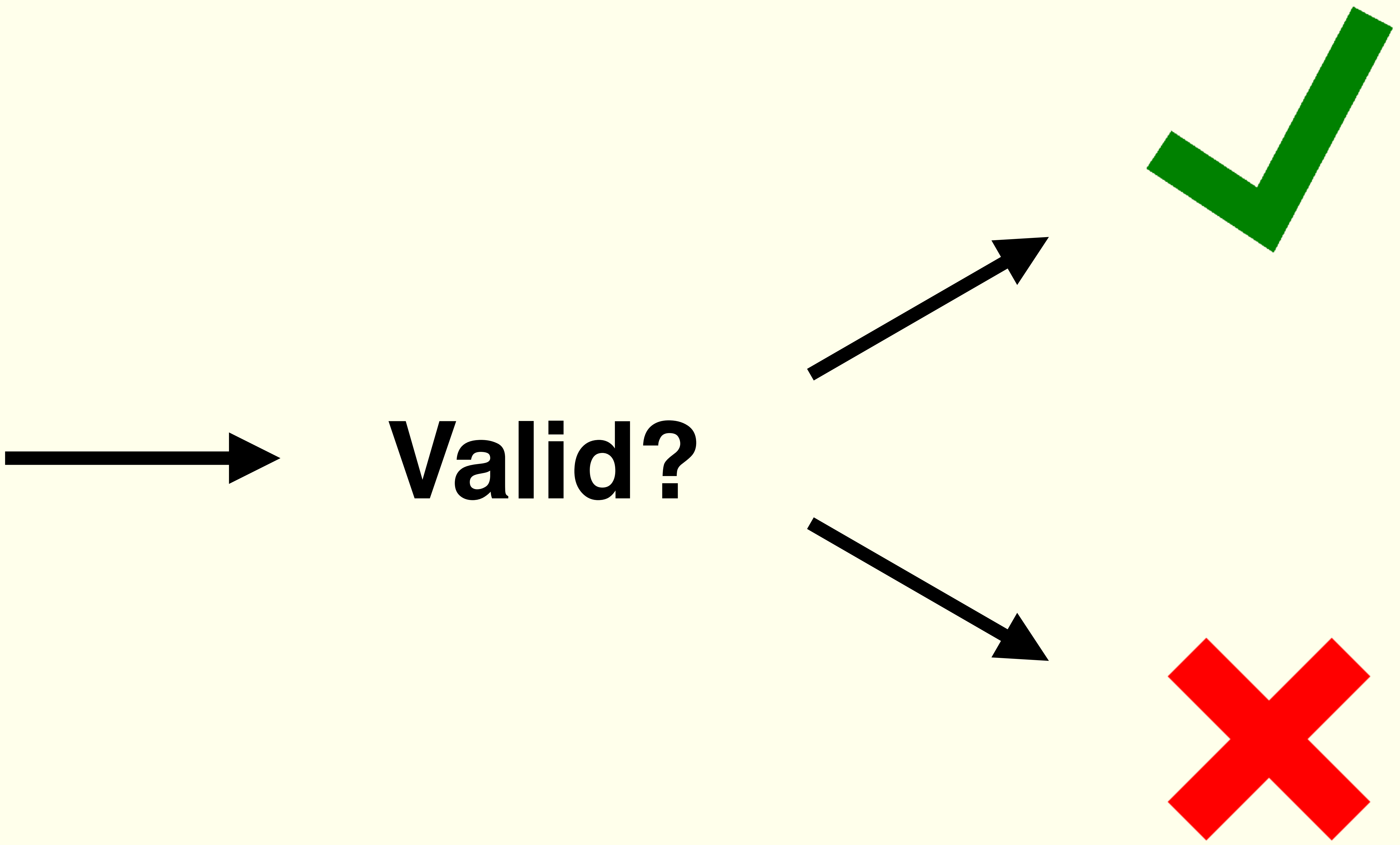
Try a number that fits

If no numbers fit

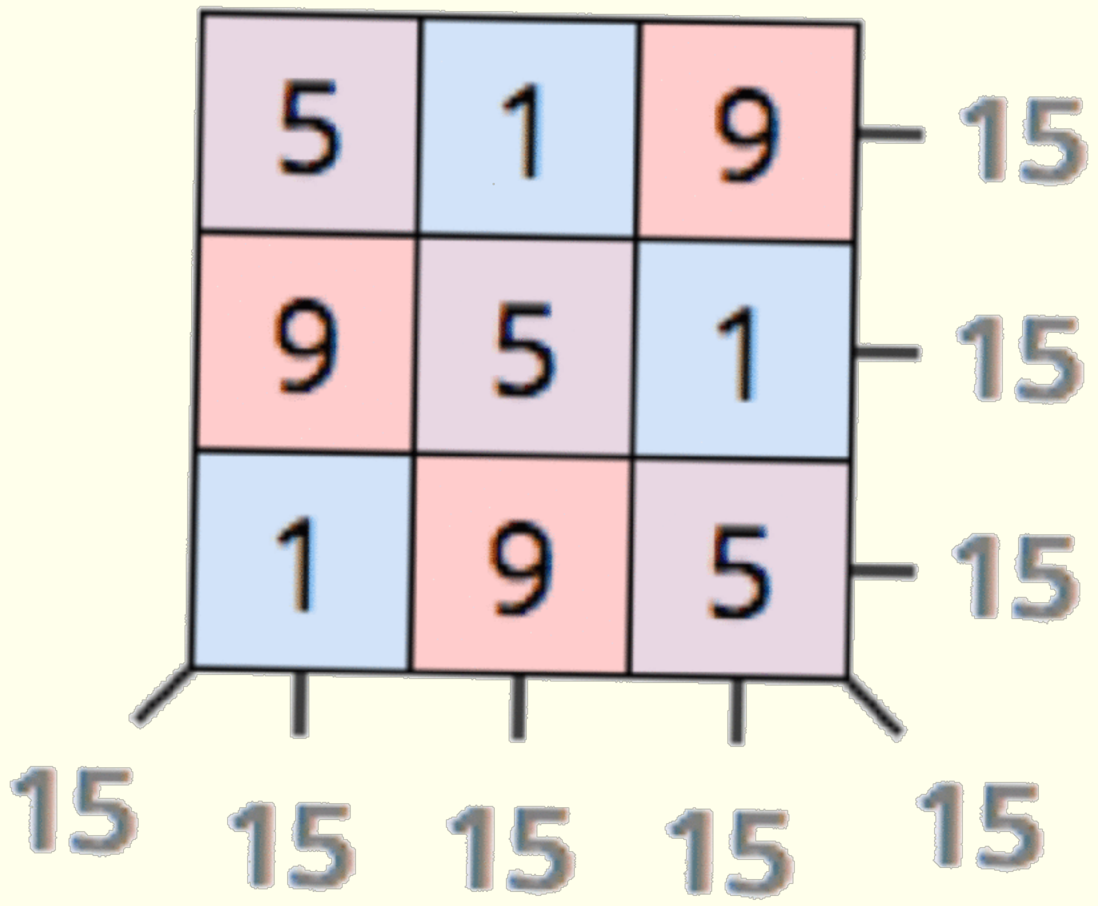
Backtrack

Sentient

6	7	8	3	5	9	4	1	2
4	2	1	7	8	6	5	3	9
5	3	9	2	1	4	6	8	7
9	4	2	6	7	3	1	5	8
1	5	3	4	2	8	9	7	6
8	6	7	1	9	5	2	4	3
7	8	6	5	4	2	3	9	1
3	1	4	9	6	7	8	2	5
2	9	5	8	3	1	7	6	4



An example



1) The **rows** must sum to the same target

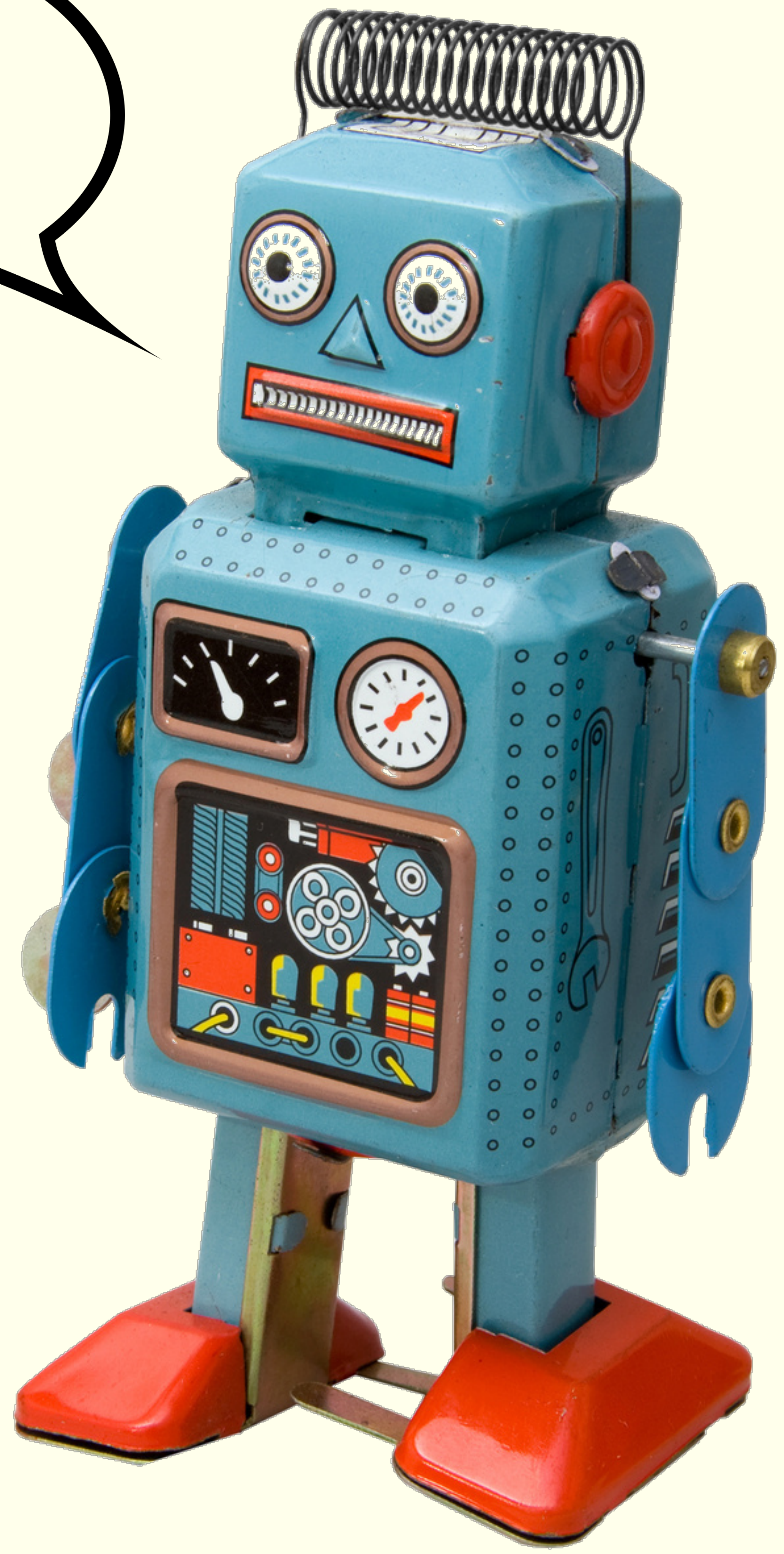
2) The **columns** must sum to the same target

3) The **diagonals** must sum to the same target

Here's how to check a magic square

Now go find some!

OK



```
array3<array3<int>> magic_square;
int target;

magic_square.each(function^ (row) {
  invariant row.sum == target;
  invariant row.all?(*positive?);
});

magic_square.transpose.each(function^ (column) {
  invariant column.sum == target;
});

left_diagonal = magic_square.map(function (row, index) {
  return row[index];
});

right_diagonal = magic_square.map(function (row, index) {
  return row.reverse[index];
});

invariant left_diagonal.sum == target;
invariant right_diagonal.sum == target;

expose magic_square, target;
```

```
array3<array3<int>> magic_square;
int target;
```

```
magic_square.each(function^ (row) {
  invariant row.sum == target;
  invariant row.all?(*positive?);
});
```

rows

```
magic_square.transpose.each(function^ (column) {
  invariant column.sum == target;
});
```

columns

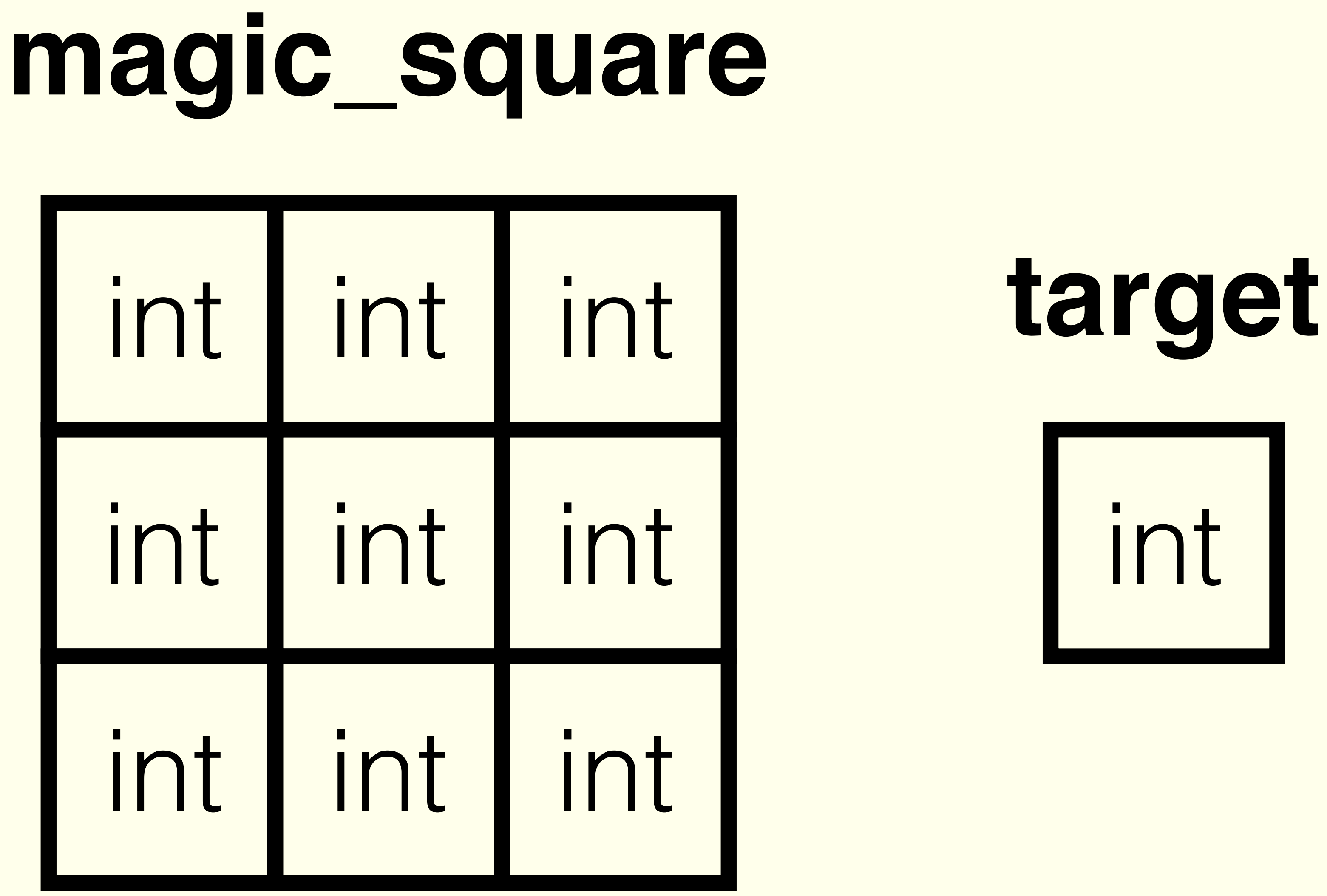
```
left_diagonal = magic_square.map(function (row, index) {
  return row[index];
});

right_diagonal = magic_square.map(function (row, index) {
  return row.reverse[index];
});

invariant left_diagonal.sum == target;
invariant right_diagonal.sum == target;
```

diagonals

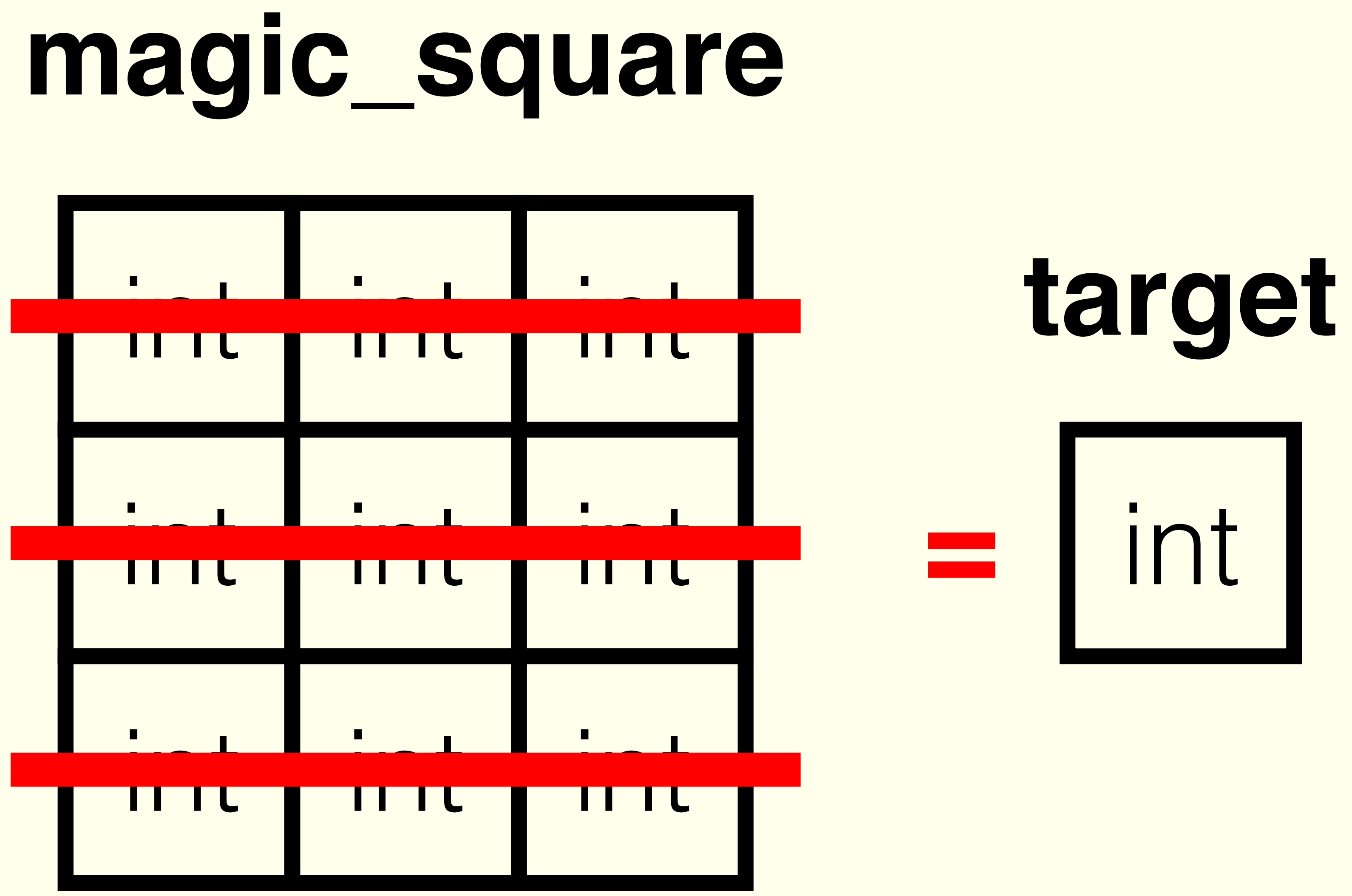
```
expose magic_square, target;
```

```
array3<array3<int>> magic_square;
int target;
```

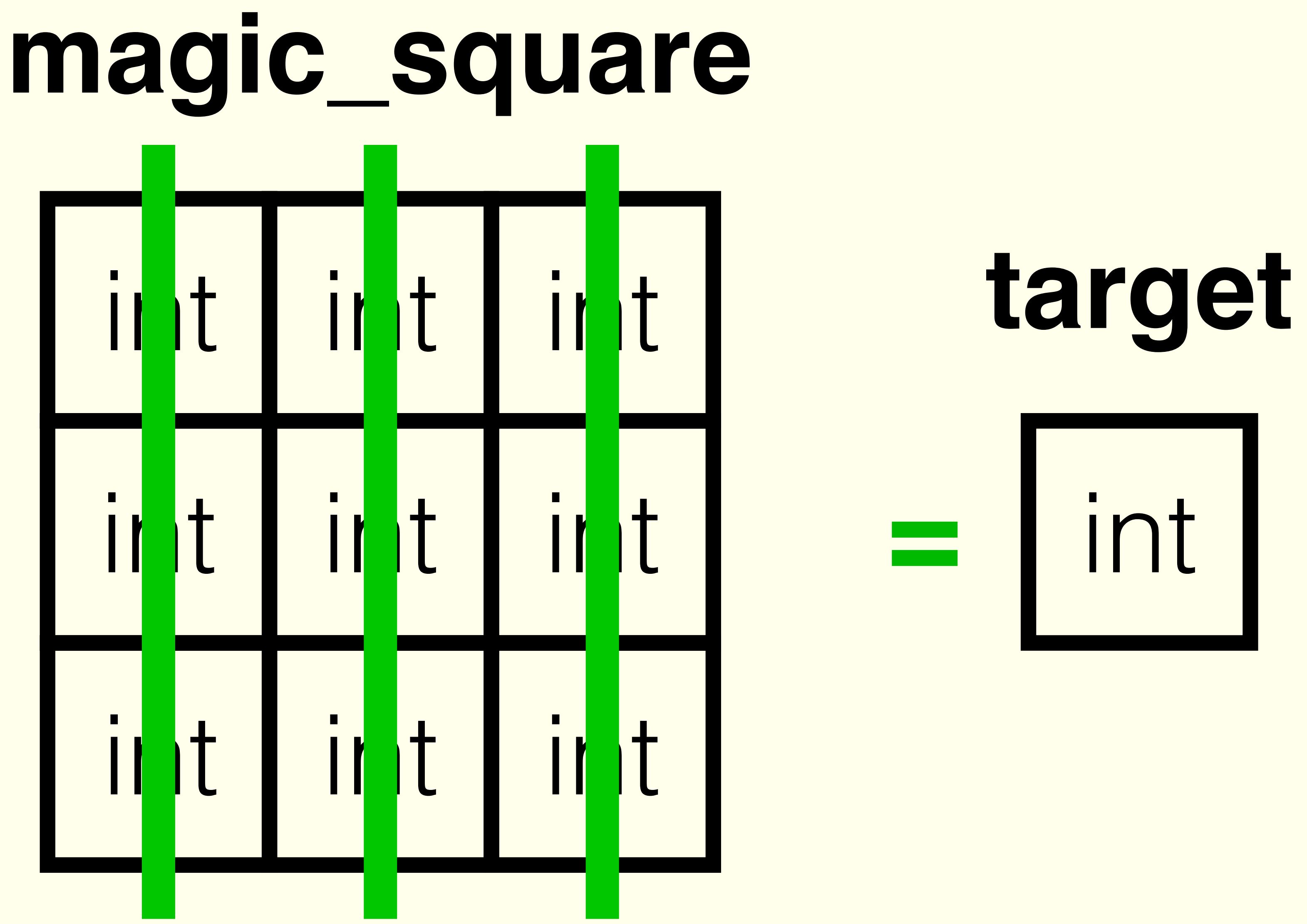
```
array3<array3<int>> magic_square;
int target;
```



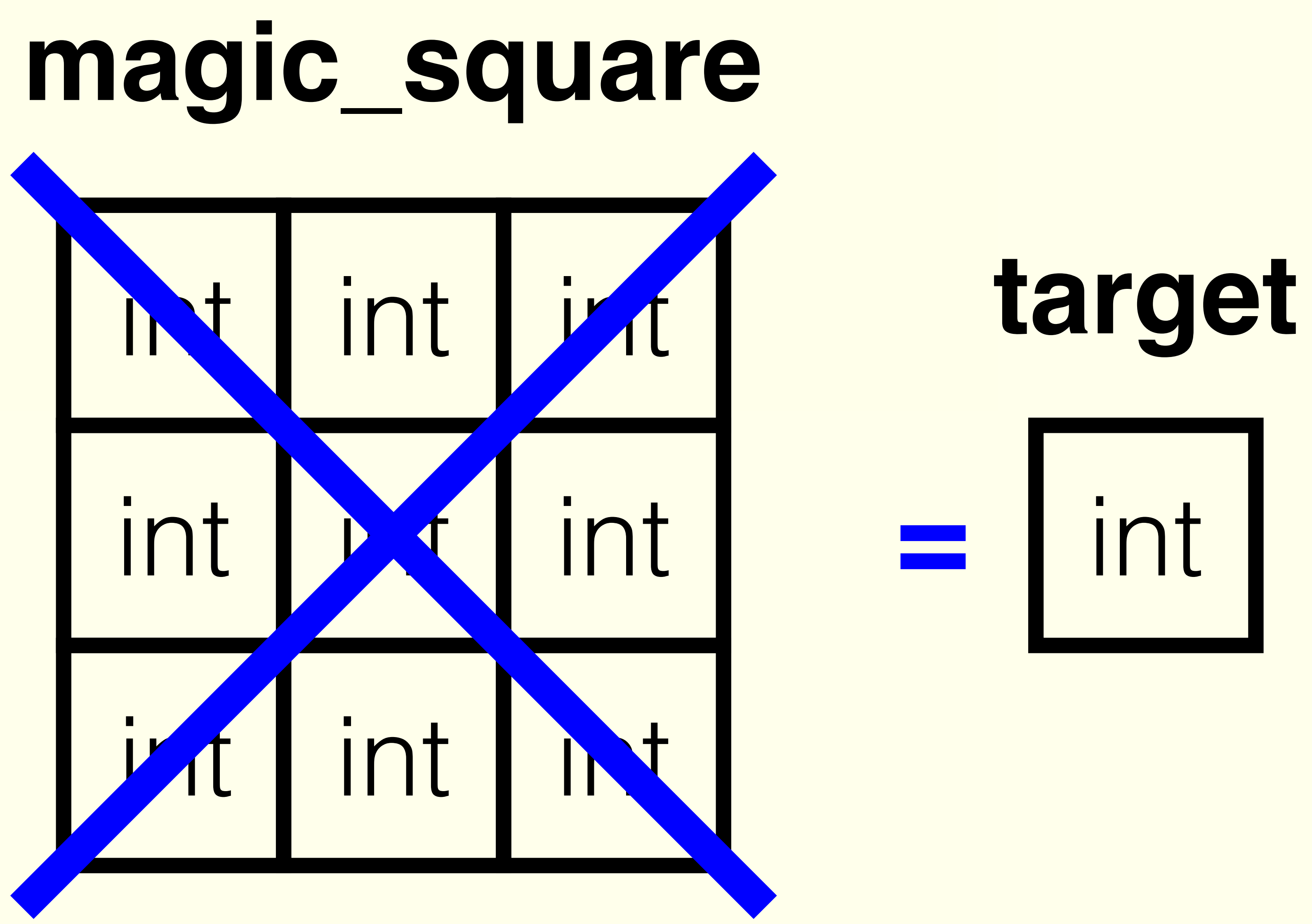
```
array3<array3<int>> magic_square;
int target;

magic_square.each(function^ (row) {
  invariant row.sum == target;
  invariant row.all?(*positive?);
});
```



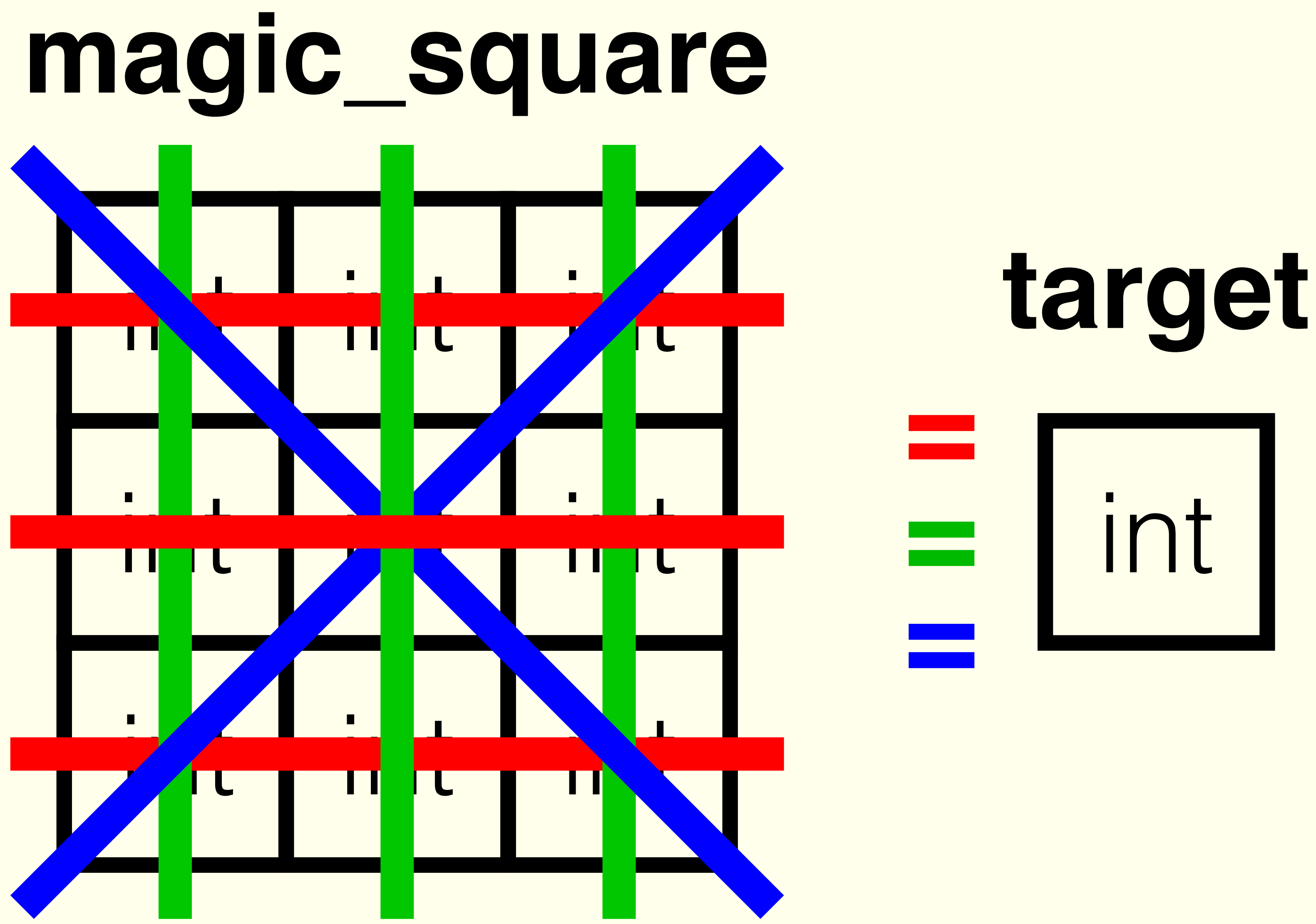
```
magic_square.transpose.each(function^ (column) {
  invariant column.sum == target;
});
```

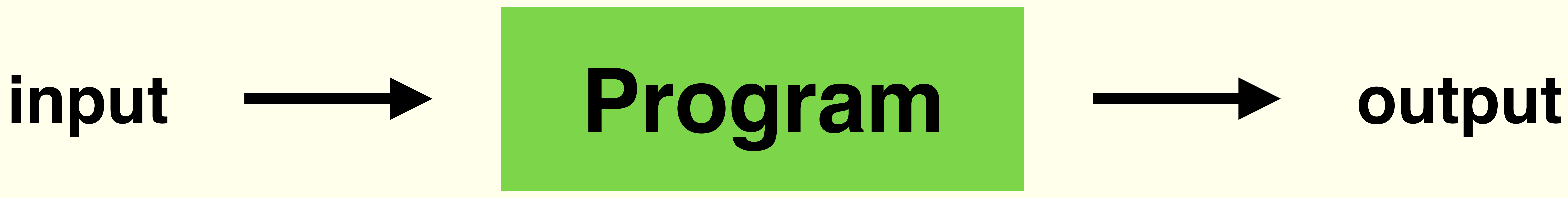


```
left_diagonal = magic_square.map(function (row, index) {
  return row[index];
});

right_diagonal = magic_square.map(function (row, index) {
  return row.reverse[index];
});

invariant left_diagonal.sum == target;
invariant right_diagonal.sum == target;
```



```
expose magic_square, target;
```



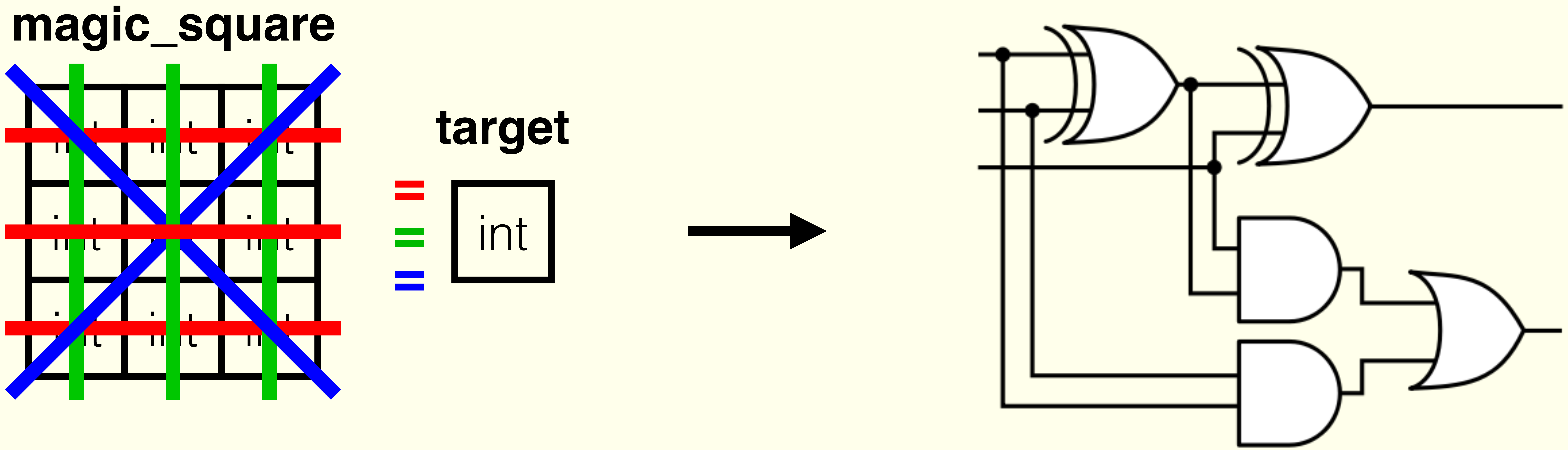

input
Program
output

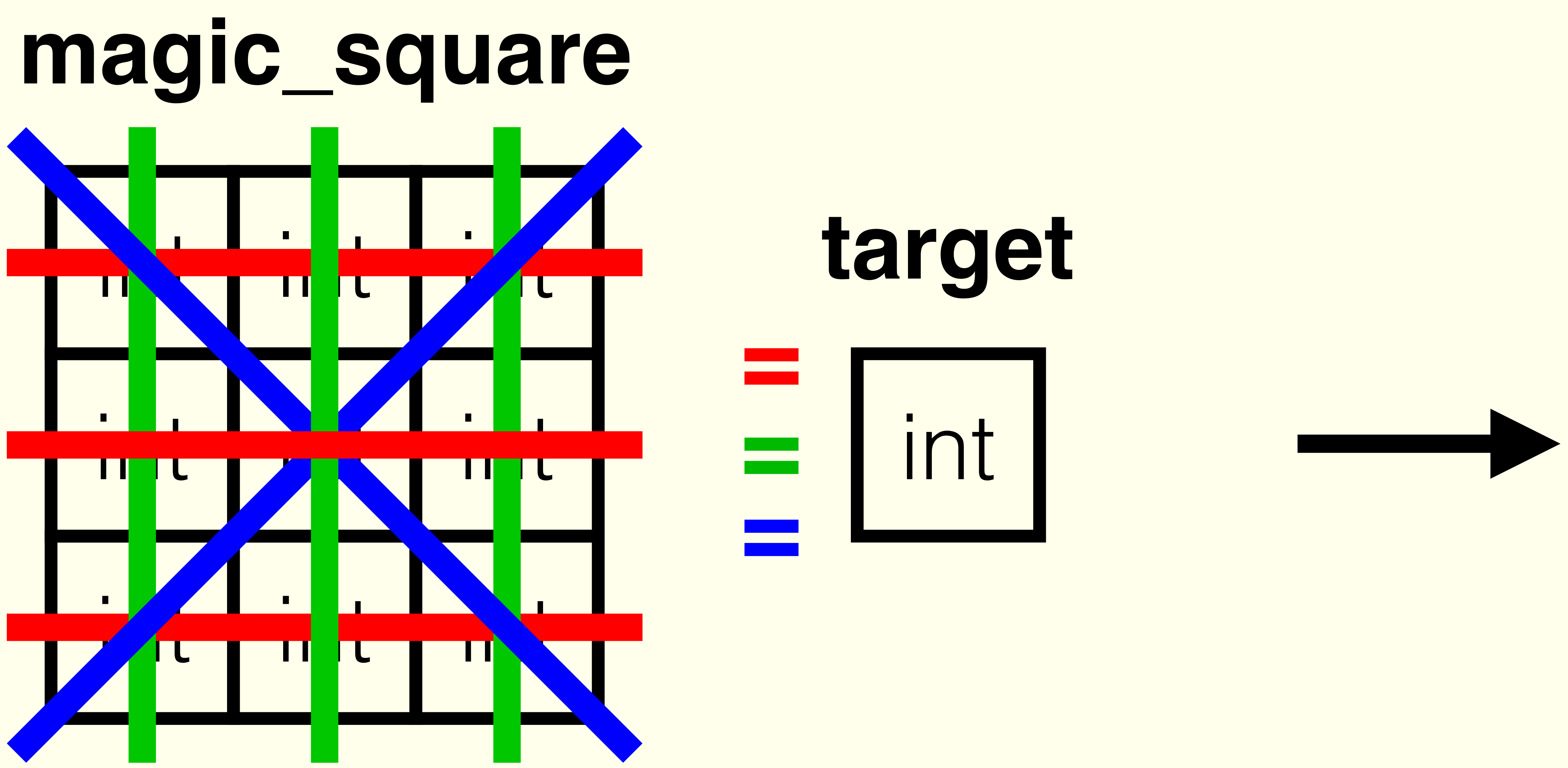

input
output
Program
input
output

```
chris:~ chris$
```


Google
Chris
Secure
https://www.google.co.uk
Mail
Images
C
Google
UK
Google Search
I'm Feeling Lucky
Advertising
Business
About
Privacy
Terms
Settings

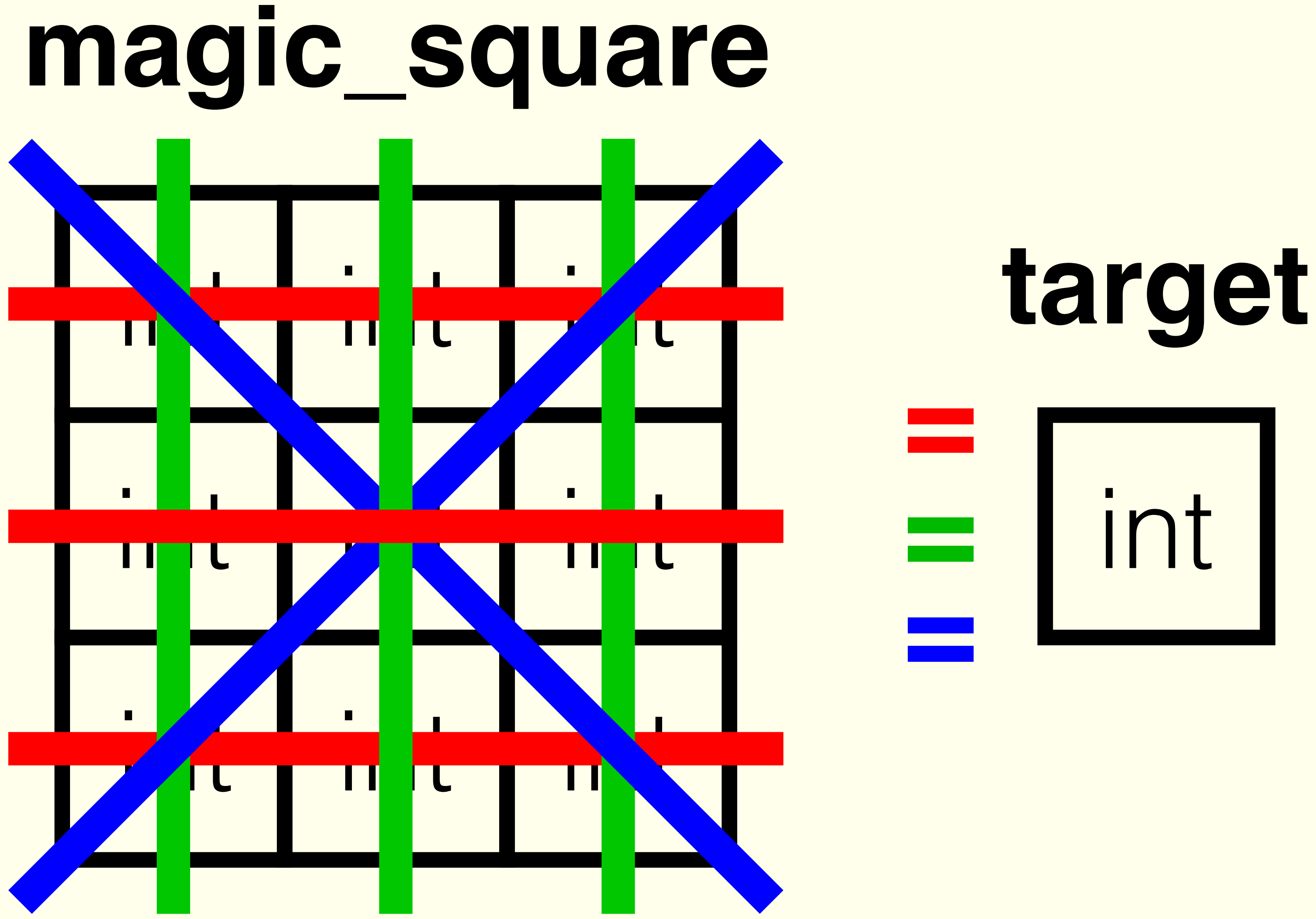
How does it work?


magic_square
target
int

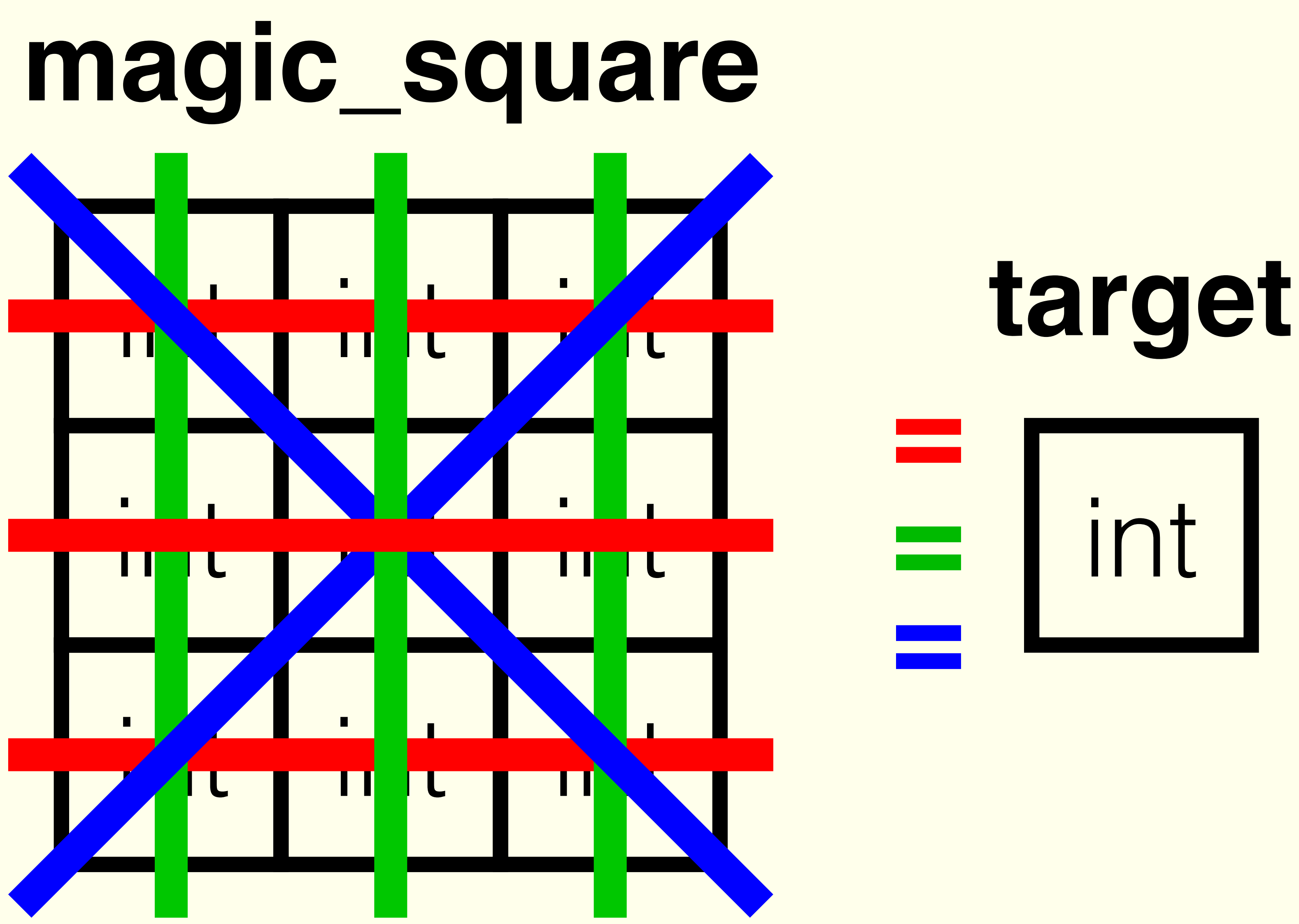


(a || b) && (!a || !c) && (a || !d)
&& (b || c) && (c || !d)

* This process is called a “Tseitin” transform which is how “Sentient” gets its name



$$(a \mid\mid b) \;\&\&\; (!a \mid\mid !c) \;\&\&\; (a \mid\mid !d) \;\&\&\; (b \mid\mid c) \;\&\&\; (c \mid\mid !d) \; \mathbf{== true}$$

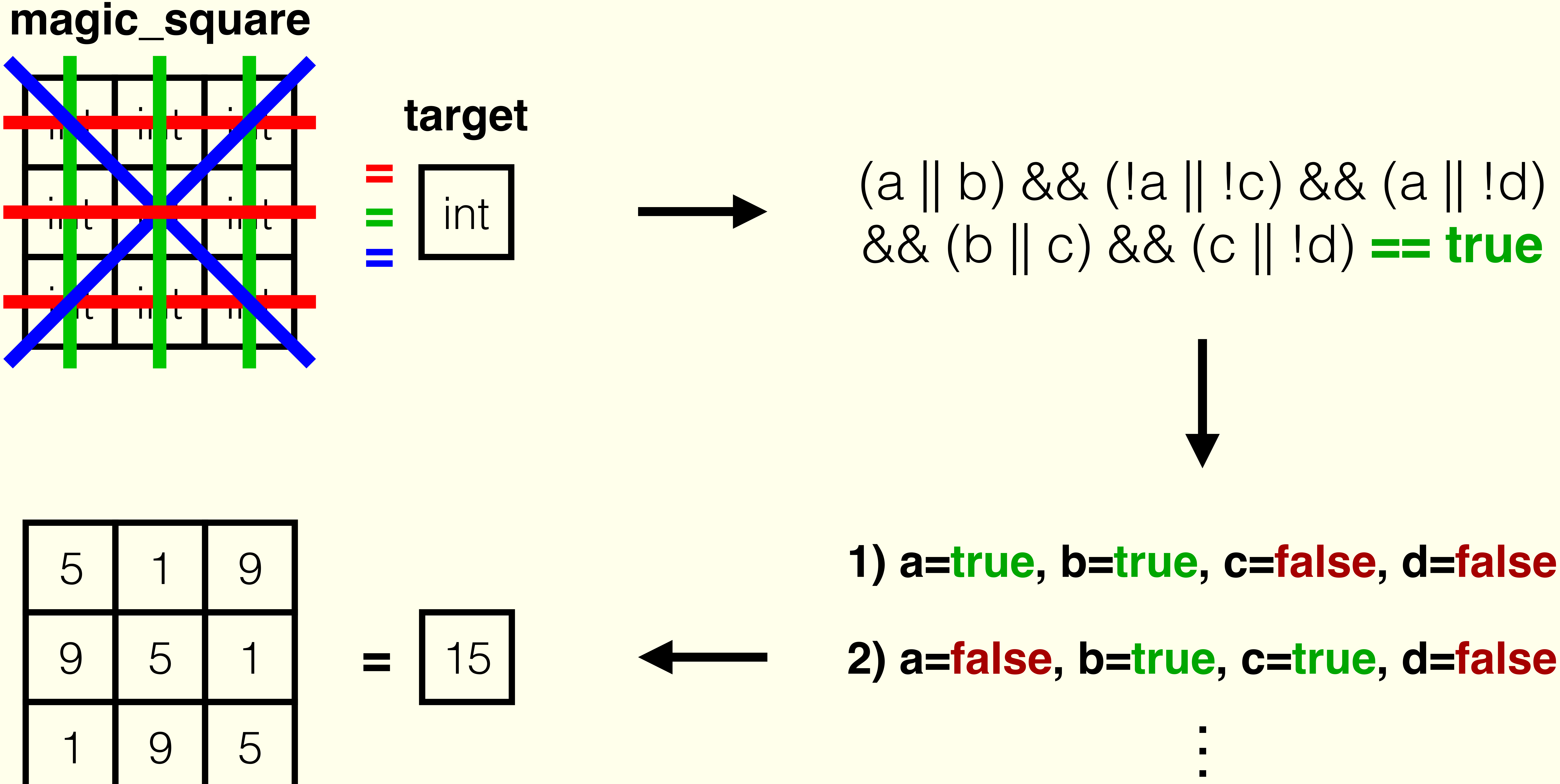


(a || b) && (!a || !c) && (a || !d) && (b || c) && (c || !d) **== true**

1) a=true, b=true, c=false, d=false

2) a=false, b=true, c=true, d=false

⋮


magic_square
target
int
=
=
=
int
(a || b) && (!a || !c) && (a || !d)
&& (b || c) && (c || !d) == true
1) a=true, b=true, c=false, d=false
2) a=false, b=true, c=true, d=false
⋮
5 1 9
9 5 1
1 9 5
=
15

(!x49) && (!x33) && (!x17) && (!x73) && (!x65) && (!x1) && (!x41) && (!x57) && (!x9) && (!x25) && (x8 || !x16 || x81) && (x8 || x16 || !x81) && (!x8 || !x16 || !x81) && (!x8 || x16 || x81) && (!x8 || !x16 || x82) && (x8 || !x82) && (x16 || !x82) && (x7 || !x15 || x83) && (x7 || x15 || !x83) && (!x7 || !x15 || !x83) && (!x7 || x15 || x83) && (!x82 || x83 || x84) && (x82 || x83 || !x84) && (!x82 || !x83 || !x84) && (x82 || !x83 || x84) && (x6 || !x14 || x86) && (x6 || x14 || !x86) && (!x6 || !x14 || !x86) && (!x6 || x14 || x86) && (!x85 || x86 || x87) && (x85 || x86 || !x87) && (!x85 || !x86 || !x87) && (x85 || !x86 || x87) && (x5 || !x13 || x89) && (x5 || x13 || !x89) && (!x5 || !x13 || !x89) && (!x5 || x13 || x89) && (!x88 || x89 || x90) && (x88 || x89 || !x90) && (!x88 || !x89 || !x90) && (x88 || !x89 || x90) && (x4 || !x12 || x92) && (x4 || x12 || !x92) && (!x4 || !x12 || !x92) && (!x4 || x12 || x92) && (!x91 || x92 || x93) && (x91 || x92 || !x93) && (!x91 || !x92 || !x93) && (x91 || !x92 || x93) && (x3 || !x11 || x95) && (x3 || x11 || !x95) && (!x3 || !x11 || !x95) && (!x3 || x11 || x95) && (!x94 || x95 || x96) && (x94 || x95 || !x96) && (!x94 || !x95 || !x96) && (x94 || !x95 || x96) && (!x94 || !x95 || x97) && (x95 || !x97) && (x94 || !x97) && (!x10 || x98) && (x2 || x10 || !x98) && (!x2 || x98) && (!x98 || x99) && (!x2 || !x10) && (!x24 || x80 || x81) && (x24 || !x80 || x81) && (!x24 || !x80 || !x81) && (x24 || x80 || !x81) && (!x24 || !x81 || x100) && (x81 || !x100) && (x24 || !x100) && (!x23 || x84 || x101) && (x23 || x84 || !x101) && (!x23 || !x84 || !x101) && (x23 || !x84 || x101) && (x79 || !x100 || x101) && (!x79 || x100 || x101) && (!x79 || !x100 || !x101) && (x79 || x100 || !x101) && (!x22 || x87 || x103) && (x22 || x87 || !x103) && (!x22 || !x87 || !x103) && (x22 || !x87 || x103) && (x78 || !x102 || x103) && (!x78 || x102 || x103) && (!x78 || !x102 || !x103) && (x78 || x102 || !x103) && (!x21 || x90 || x105) && (x21 || x90 || !x105) && (!x21 || !x90 || !x105) && (x21 || !x90 || x105) && (x77 || !x104 || x105) && (!x77 || x104 || x105) && (!x77 || !x104 || !x105) && (x77 || x104 || !x105) && (!x20 || x93 || x107) && (x20 || x93 || !x107) && (!x20 || !x93 || !x107) && (x20 || !x93 || x107) && (x76 || !x106 || x107) && (!x76 || x106 || x107) && (!x76 || !x106 || !x107) && (x76 || x106 || !x107) && (!x19 || x96 || x109) && (x19 || x96 || !x109) && (!x19 || !x96 || !x109) && (x19 || !x96 || x109) && (x75 || !x108 || x109) && (!x75 || x108 || x109) && (!x75 || !x108 || !x109) && (x75 || x108 || !x109) && (!x108 || !x109 || x110) && (x109 || !x110) && (x108 || !x110) && (!x18 || x111) && (x18 || x99 || !x111) && (!x99 || x111) && (x74 || !x111) && (!x18 || !x99) && (x32 || !x40 || x112) && (x32 || x40 || !x112) && (!x32 || !x40 || !x112) && (!x32 || x40 || x112) && (!x32 || !x40 || x113) && (x32 || !x113) && (x40 || !x113) && (x31 || !x39 || x114) && (x31 || x39 || !x114) && (!x31 || !x39 || !x114) && (!x31 || x39 || x114) && (!x113 || x114 || x115) && (x113 || x114 || !x115) && (!x113 || !x114 || !x115) && (x113 || !x114 || x115) && (x30 || !x38 || x117) && (x30 || x38 || !x117) && (!x30 || !x38 || !x117) && (!x30 || x38 || x117) && (!x116 || x117 || x118) && (x116 || x117 || !x118) && (!x116 || !x117 || !x118) && (x116 || !x117 || x118) && (x29 || !x37 || x120) && (x29 || x37 || !x120) && (!x29 || !x37 || !x120) && (!x29 || x37 || x120) && (!x119 || x120 || x121) && (x119 || x120 || !x121) && (!x119 || !x120 || !x121) && (x119 || !x120 || x121) && (x28 || !x36 || x123) && (x28 || x36 || !x123) && (!x28 || !x36 || !x123) && (!x28 || x36 || x123) && (!x122 || x123 || x124) && (x122 || x123 || !x124) && (!x122 || !x123 || !x124) && (x122 || !x123 || x124) && (x27 || !x35 || x126) && (x27 || x35 || !x126) && (!x27 || !x35 || !x126) && (!x27 || x35 || x126) && (!x125 || x126 || x127) && (x125 || x126 || !x127) && (!x125 || !x126 || !x127) && (x125 || !x126 || x127) && (!x27 || !x35 || x128) && (x27 || !x128) && (x35 || !x128) && (!x34 || x129) && (x26 || x34 || !x129) && (!x26 || x129) && (!x129 || x130) && (!x26 || !x34) && (!x48 || x80 || x112) && (x48 || !x80 || x112) && (!x48 || !x80 || !x112) && (x48 || x80 || !x112) && (!x48 || !x112 || x131) && (x112 || !x131) && (x48 || !x131) && (!x47 || x115 || x132) && (x47 || x115 || !x132) && (!x47 || !x115 || !x132) && (x47 || !x115 || x132) && (x79 || !x131 || x132) && (!x79 || x131 || x132) && (!x79 || !x131 || !x132) && (x79 || x131 || !x132) && (!x46 || x118 || x134) && (x46 || x118 || !x134) && (!x46 || !x118 || !x134) && (x46 || !x118 || x134) && (x78 || !x133 || x134) && (!x78 || x133 || x134) && (!x78 || !x133 || !x134) && (x78 || x133 || !x134) && (!x45 || x121 || x136) && (x45 || x121 || !x136) && (!x45 || !x121 || !x136) && (x45 || !x121 || x136) && (x77 || !x135 || x136) && (!x77 || x135 || x136) && (!x77 || !x135 || !x136) && (x77 || x135 || !x136) && (!x44 || x124 || x138) && (x44 || x124 || !x138) && (!x44 || !x124 || !x138) && (x44 || !x124 || x138) && (x76 || !x137 || x138) && (!x76 || x137 || x138) && (!x76 || !x137 || !x138) && (x76 || x137 || !x138) && (!x43 || x127 || x140) && (x43 || x127 || !x140) && (!x43 || !x127 || !x140) && (x43 || !x127 || x140) && (x75 || !x139 || x140) && (!x75 || x139 || x140) && (!x75 || !x139 || !x140) && (x75 || x139 || !x140) && (!x139 || !x140 || x141) && (x140 || !x141) && (x139 || !x141) && (!x42 || x142) && (x42 || x130 || !x142) && (!x130 || x142) && (x74 || !x142) && (!x42 || !x130) && (x56 || !x64 || x143) && (x56 || x64 || !x143) && (!x56 || !x64 || !x143) && (!x56 || x64 || x143) && (!x56 || !x64 || x144) && (x56 || !x144) && (x64 || !x144) && (x55 || !x63 || x145) && (x55 || x63 || !x145) && (!x55 || !x63 || !x145) && (!x55 || x63 || x145) && (!x144 || x145 || x146) && (x144 || x145 || !x146) && (!x144 || !x145 || !x146) && (x144 || !x145 || x146) && (x54 || !x62 || x148) && (x54 || x62 || !x148) && (!x54 || !x62 || !x148) && (!x54 || x62 || x148) && (!x147 || x148 || x149) && (x147 || x148 || !x149) && (!x147 || !x148 || !x149) && (x147 || !x148 || x149) && (x53 || !x61 || x151) && (x53 || x61 || !x151) && (!x53 || !x61 || !x151) && (!x53 || x61 || x151) && (!x150 || x151 || x152) && (x150 || x151 || !x152) && (!x150 || !x151 || !x152) && (x150 || !x151 || x152) && (x52 || !x60 || x154) && (x52 || x60 || !x154) && (!x52 || !x60 || !x154) && (!x52 || x60 || x154) && (!x153 || x154 || x155) && (x153 || x154 || !x155) && (!x153 || !x154 || !x155) && (x153 || !x154 || x155) && (x51 || !x59 || x157) && (x51 || x59 || !x157) && (!x51 || !x59 || !x157) && (!x51 || x59 || x157) && (!x156 || x157 || x158) && (x156 || x157 || !x158) && (!x156 || !x157 || !x158) && (x156 || !x157 || x158) && (!x51 || !x59 || x159) && (x51 || !x159) && (x59 || !x159) && (!x58 || x160) && (x50 || x58 || !x160) && (!x50 || x160) && (!x160 || x161) && (!x50 || !x58) && (!x72 || x80 || x143) && (x72 || !x80 || x143) && (!x72 || !x80 || !x143) && (x72 || x80 || !x143) && (!x72 || !x143 || x162) && (x143 || !x162) && (x72 || !x162) && (!x71 || x146 || x163) && (x71 || x146 || !x163) && (!x71 || !x146 || !x163) && (x71 || !x146 || x163) && (x79 || !x162 || x163) && (!x79 || x162 || x163) && (!x79 || !x162 || !x163) && (x79 || x162 || !x163) && (!x70 || x149 || x165) && (x70 || x149 || !x165) && (!x70 || !x149 || !x165) && (x70 || !x149 || x165) && (x78 || !x164 || x165) && (!x78 || x164 || x165) && (!x78 || !x164 || !x165) && (x78 || x164 || !x165) && (!x69 || x152 || x167) && (x69 || x152 || !x167) && (!x69 || !x152 || !x167) && (x69 || !x152 || x167) && (x77 || !x166 || x167) && (!x77 || x166 || x167) && (!x77 || !x166 || !x167) && (x77 || x166 || !x167) && (!x68 || x155 || x169) && (x68 || x155 || !x169) && (!x68 || !x155 || !x169) && (x68 || !x155 || x169) && (x76 || !x168 || x169) && (!x76 || x168 || x169) && (!x76 || !x168 || !x169) && (x76 || x168 || !x169) && (!x67 || x158 || x171) && (x67 || x158 || !x171) && (!x67 || !x158 || !x171) && (x67 || !x158 || x171) && (x75 || !x170 || x171) && (!x75 || x170 || x171) && (!x75 || !x170 || !x171) && (x75 || x170 || !x171) && (!x67 || !x158 || x172) && (x158 || !x172) && (x67 || !x172) && (!x66 || x173) && (x66 || x161 || !x173) && (!x161 || x173) && (x74 || !x173) && (!x66 || !x161) && (x8 || !x32 || x174) && (x8 || x32 || !x174) && (!x8 || !x32 || !x174) && (!x8 || x32 || x174) && (!x8 || !x32 || x175) && (x8 || !x175) && (x32 || !x175) && (x7 || !x31 || x176) && (x7 || x31 || !x176) && (!x7 || !x31 || !x176) && (!x7 || x31 || x176) && (!x175 || x176 || x177) && (x175 || x176 || !x177) && (!x175 || !x176 || !x177) && (x175 || !x176 || x177) && (x6 || !x30 || x179) && (x6 || x30 || !x179) && (!x6 || !x30 || !x179) && (!x6 || x30 || x179) && (!x178 || x179 || x180) && (x178 || x179 || !x180) && (!x178 || !x179 || !x180) && (x178 || !x179 || x180) && (x5 || !x29 || x182) && (x5 || x29 || !x182) && (!x5 || !x29 || !x182) && (!x5 || x29 || x182) && (!x181 || x182 || x183) && (x181 || x182 || !x183) && (!x181 || !x182 || !x183) && (x181 || !x182 || x183) && (x4 || !x28 || x185) && (x4 || x28 || !x185) && (!x4 || !x28 || !x185) && (!x4 || x28 || x185) && (!x184 || x185 || x186) && (x184 || x185 || !x186) && (!x184 || !x185 || !x186) && (x184 || !x185 || x186) && (x3 || !x27 || x188) && (x3 || x27 || !x188) && (!x3 || !x27 || !x188) && (!x3 || x27 || x188) && (!x187 || x188 || x189) && (x187 || x188 || !x189) && (!x187 || !x188 || !x189) && (x187 || !x188 || x189) && (!x187 || !x188 || x190) && (x188 || !x190) && (x187 || !x190) && (!x26 || x191) && (x2 || x26 || !x191) && (!x2 || x191) && (!x191 || x192) && (!x2 || !x26) && (!x56 || x80 || x174) && (x56 || !x80 || x174) && (!x56 || !x80 || !x174) && (x56 || x80 || !x174) && (!x56 || !x174 || x193) && (x174 || !x193) && (x56 || !x193) && (!x55 || x177 || x194) && (x55 ||

My favourite puzzle

This pangram contains ___ a's, ___ b's, ___ c's, ___ d's, ___ e's, ___ f's, ___ g's, ___ h's, ___ i's, ___ j’s, ___ k’s, ___ l's, ___ m's, ___ n's, ___ o's, ___ p’s, ___ q’s, ___ r's, ___ s's, ___ t's, ___ u's, ___ v's, ___ w's, ___ x's, ___ y’s and ___ z’s.

This pangram contains **four** a's, **two** b's, **one** c,
___ d's, ___ e's, ___ f's, ___ g's, ___ h's, ___ i's,
___ j’s, ___ k’s, ___ l's, ___ m's, ___ n's, ___ o's,
___ p’s, ___ q’s, ___ r's, ___ s's, ___ t's, ___ u's,
___ v's, ___ w's, ___ x's, ___ y’s and ___ z’s.

$40^{26} =$

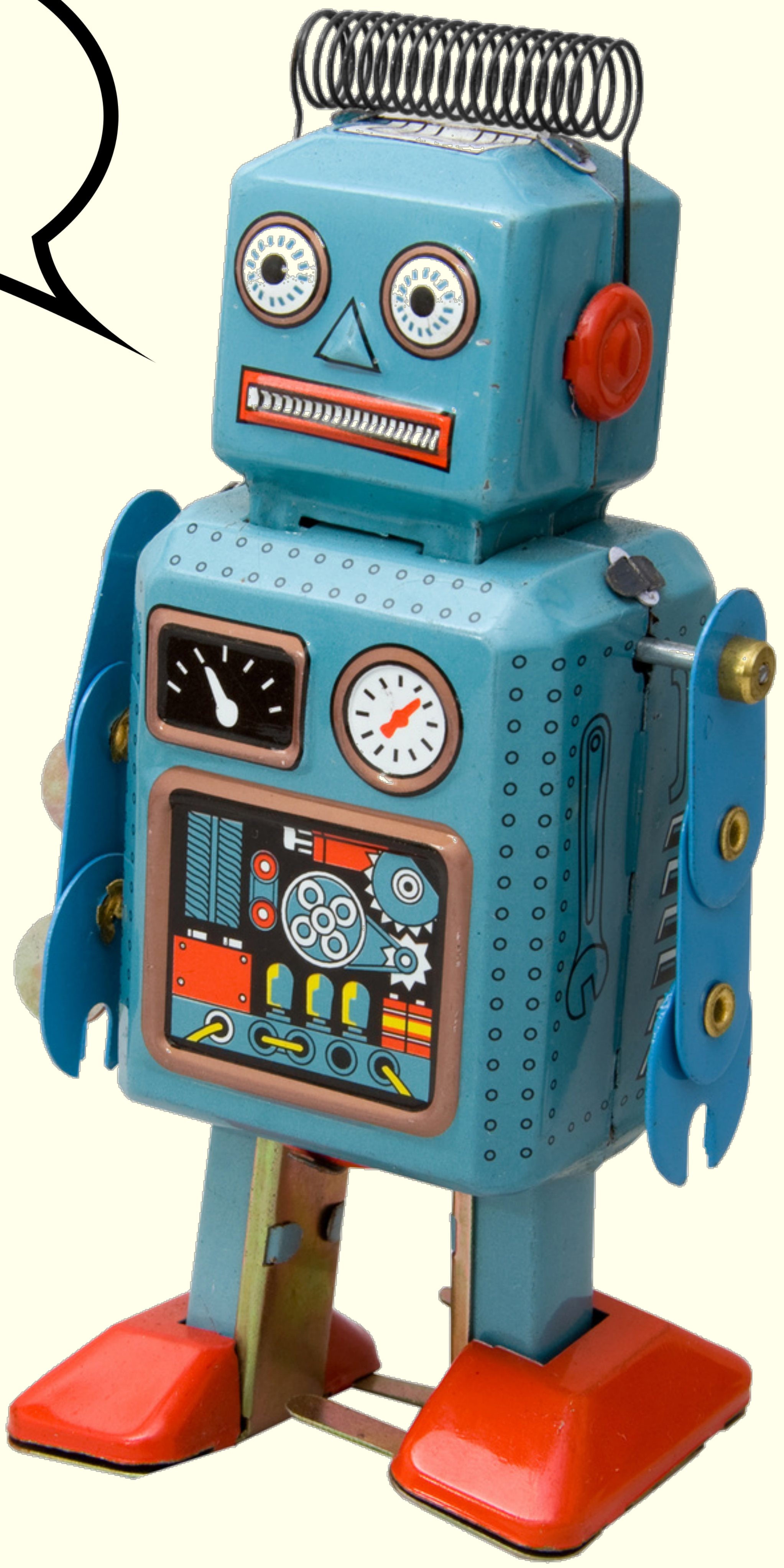
450,359,962,737,049,600,000,
000,000,000,000,000,000,000

This pangram contains ___ a's, ___ b's, ___ c's, ___ d's, ___ e's, ___ f's, ___ g's, ___ h's, ___ i's, ___ j’s, ___ k’s, ___ l's, ___ m's, ___ n's, ___ o's, ___ p’s, ___ q’s, ___ r's, ___ s's, ___ t's, ___ u's, ___ v's, ___ w's, ___ x's, ___ y’s and ___ z’s.

Here's how to check a pangram

Now go find some!

OK



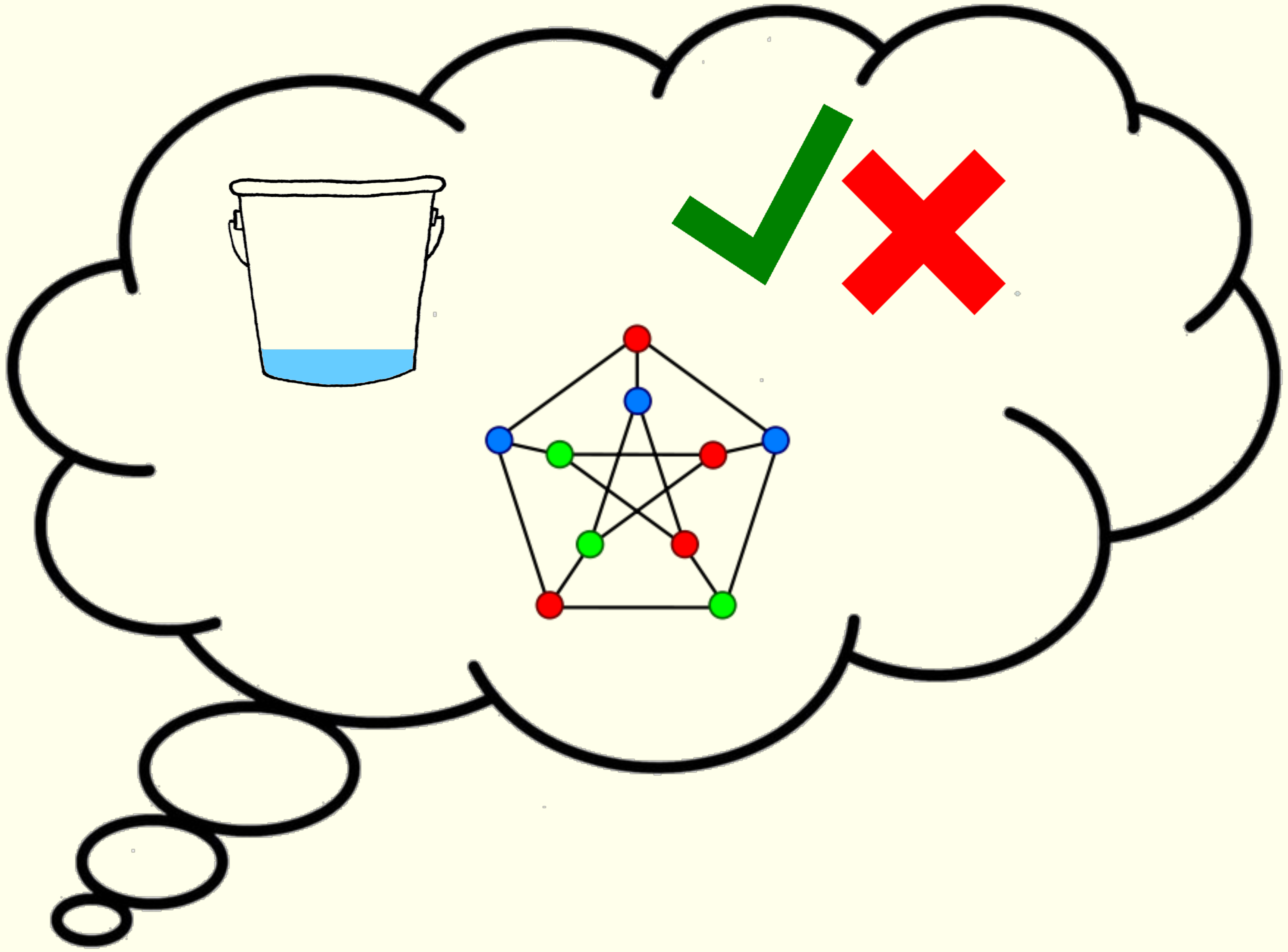
2 minutes

This sentence is dedicated to Ember London and it contains five a's, two b's, four c's, seven d's, thirty-four e's, eight f's, four g's, eight h's, sixteen i's, one j, one k, two l's, two m's, twenty-three n's, eighteen o's, one p, one q, nine r's, thirty s's, twenty-four t's, five u's, seven v's, seven w's, two x's, five y’s and one z.

This sentence is dedicated to Ember London and it contains five a's, two b's, four c's, seven d's, thirty-four e's, eight f's, four g's, eight h's, sixteen i's, one j, one k, two l's, two m's, twenty-three n's, eighteen o's, one p, one q, nine r's, thirty s's, twenty-four t's, five u's, seven v's, seven w's, two x's, five y’s and one z.

This sentence is dedicated to Em**b**er London and it contains five a's, two **b**'s, four c's, seven d's, thirty-four e's, eight f's, four g's, eight h's, sixteen i's, one j, one k, two l's, two m's, twenty-three n's, eighteen o's, one p, one q, nine r's, thirty s's, twenty-four t's, five u's, seven v's, seven w's, two x's, five y’s and one z.

To wrap up



sentient-lang.org

whyarecomputers.com/4

github.com/sentient-lang

Thanks!

by Chris Patuzzo
Ember London, 2017-01-12
@cpatuzzo