

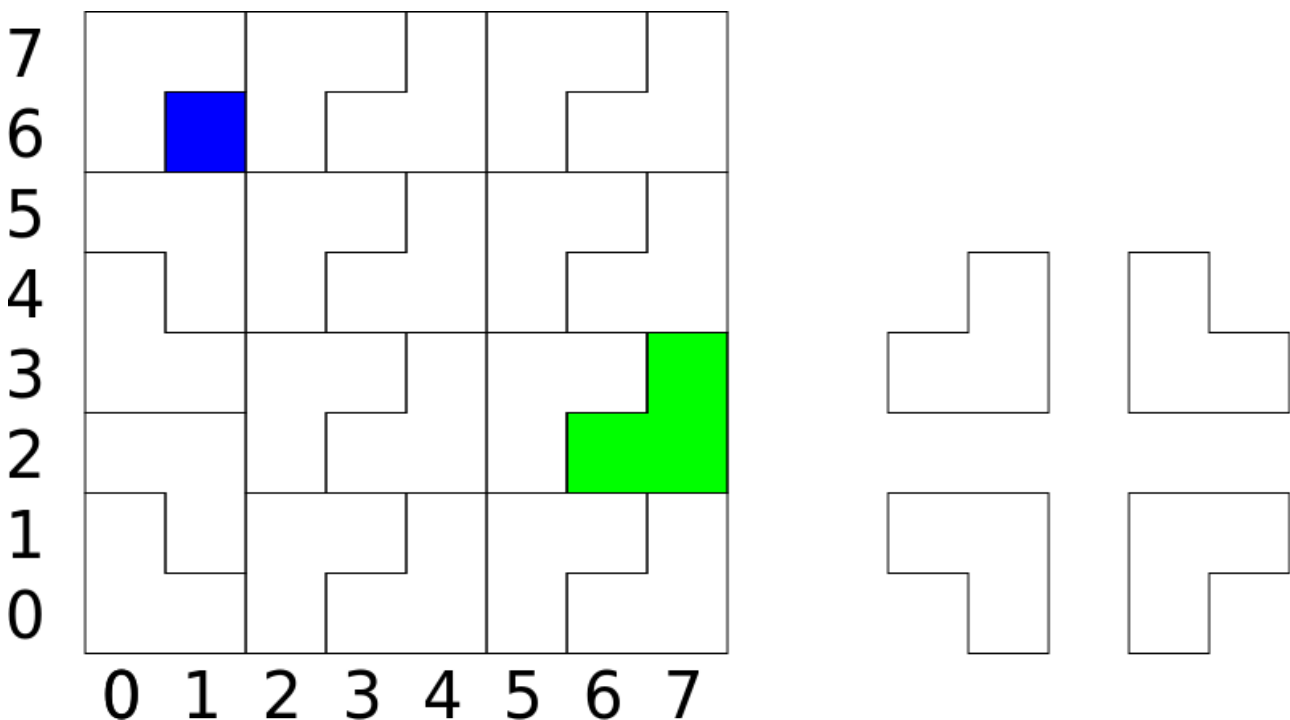
Tile

Place tiles on the floor

We have a 2^n by 2^n floor with 4^n unit squares. Each square can be identified by its x and y coordinates (both from 0 to $2^n - 1$). We want to cover this floor with four types of 2 by 2 L-shaped tiles, as shown in the following figure. The tiles cannot overlap, and they can only be placed on grid boundary. In addition, there is exactly a square that we cannot cover with tiles. Please find a way to cover the entire floor with tiles.

Example

In the following example we have $n = 3$ and 64 squares. The blue square at (1, 6) cannot be covered by any tile. This figure shows a possible way to cover the floor. To identify a tile we will use the x and y coordinates. For example, the green tile in the figure can be identified by (6, 2), (7, 2), and (7, 3).



Statement

Write a function `placeTile` to cover the floor with tiles.

- `placeTile(n, x, y, tile)`
 - n indicates that the size of the floor is 2^n by 2^n .
 - x , and y are the coordinates of the square that cannot be covered with tiles.
 - `tile` is a two-dimensional array to place the tile positions. For example, if we want to place the green tile as the first tile, we can set `tile[0][0]`, `tile[0][1]`, ... to `tile[0][5]`, as 6, 2, 7, 2, 7, and 3 respectively. The three squares of a tile can appear in any order, so we can also set them as 7, 2, 7, 3, and 6, 2. The tile can also appear in any order.

Subtask 1 (5 points)

- n is 1.

Subtask 2 (15 points)

- $1 \leq n \leq 2$.

Subtask 3 (20 points)

- $1 \leq n \leq 8$, and $x = y = 0$.

Subtask 4 (60 points)

- $1 \leq n \leq 8$.

Implementation details

You have to submit exactly one file, called `tile.c`, `tile.cpp` or `tile.pas`. This file implements the subprogram described above using the following signatures. You also need to include a header file `tile.h`.

C/C++ program

```
#include "tile.h"
void placeTile(int n, int x, int y, int tile[][6]);
```

Pascal program

```
type
  tileType = array[0..21844][0..5] of longint;
procedure placeTile(n, x, y: longint; var tile: tileType);
```

This subprogram must behave as described above. Of course you are free to implement other subprograms for their internal use. Your submissions must not interact in any way with standard input/output, nor with any other file.

Sample grader

The sample grader reads the input from the standard input with the following format:

- line 1: n, x, and y.

The sample grader writes "correct." to the standard output if your program is correct, "incorrect." otherwise.