Hitori

Assignment 3: Grid Management Functions

The objective of this assignment is to develop more elaborate utility functions that will help in grid solving.

2. Color management functions and grid copy

1. Modify type `grid_t` such that each byte representing a cell does not only contain its value but also its color (`black, white`, or `grey`), defined by a new type `color_t.` Modify the grid parser such that the initial color of a cell is set to `grey`. Two bits per cell can be used for that purpose (or one bit per color if you prefer).
2. Implement the two following functions that get (respectively set) the color of a cell. The functions will check grid bounds and color validity.
   ```
   void set_color (int i, int j, grid_t *g, color_t c);
   char get_color (int i, int j, grid_t *g);
   ```
3. Implement the two following functions that get (respectively set) the value of a cell, with no modification of its color. The functions will check grid bounds and value validity.
   ```
   void set_value (int i, int j, grid_t *g, char c);
   char get_value (int i, int j, grid_t *g);
   ```
4. Modify the grid printing function `grid_print` such that it now handles the cell color. ANSI escape codes can be used for that purpose.

3. Grid validity control

1. Implement a function that returns true if a grid is *consistent*, i.e. meets the constraints of the Hitori game:
   a. At most one occurrence of any digit colored in `white` in any line/column
   b. No consecutive cells colored in black in any line/column
   c. No non-black cells are cut off from the rest of the grid (i.e. there exists a path from any non-black cell to any other one using elementary horizontal or vertical moves)

   This function should work even if the grid is not yet finished (i.e. still contains `grey` cells). The function will return `false` if the grid is not consistent, and `true` otherwise. The signature will be as follows:
   ```
   bool is_consistent(grid_t *g);
   ```
2. Implement a function that returns true if a grid is finished (no `grey` cells) and meets all the constraints of the Hitori game:
   ```
   bool is_valid(grid_t *g);
   ```

Intermediate functions can be developed to work on individual lines/columns.

To be able to automate testing, print a message on `stdout` to indicate if a grid is consistent and if it is valid. The format of the output will be as followed:

```
Input grid is consistent:     0/1
Input grid is valid:          0/1
```