# SCALE FOR PROJECT COLLE 01 (/PROJECTS/BOOTCAMP-COLLE-01)

## Guidelines

Before inspecting and grading this project, make sure to read and apply preliminaries.

## Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/694/fc1b2b369a2444205385b7440bcb23ea-colle01.en.pdf)

## Preliminaries

*If cheating is suspected, the evaluation stops here. Tick the "Cheat" correction box at the top of this Scale page. You then have to let us know about it by opening a ticket on meta.intrav2.42.fr. In this ticket, you should indicate the Bootcamp day as well as the cheating student's login. => FYI : cheating means the student who is suspected cannot explain of recreate their own code, or he's used elements specifically forbidden in the subject or that go against school policies. Please do this calmy, wisely and with caution.*

**Simple preliminaries**

- Nothing submitted (or wrong file or directory) : 0, evaluation is over.
- Norm error : evaluation stops.
- As soon as you come accross an exercise that isn't fully fonctional, evaluation stops. The following exercises won't be evaluated.

☑ Yes ☒ No

**Tests éliminatoires**

First, run some elementary error handling tests. If you encounter any problem, discuss it with the students you're grading, but don't give them any points.

Carry out the following series of tests:

- A grid poorly shapped
$> ./sudoku "9...7...." "2...9..53" ".6..124.." "84...1.9." "5....8.." ".31..4..." "..37..68.." ".9..5.741" "47......."

- A wrong grid
$> ./sudoku "9...7...." "2...9..53" ".6..124.." "84...1.9." "5.....8.." ".31..4..." "..37..68." ".9..5.741" "48......."

- A grid with wrong character (, in this case)
$> ./sudoku "9...7.,.." "2...9..53" ".6..124.." "84...1.9." "5.....8.." ".31..4..." "..37..68." ".9..5.741" "48......."

- A grid with not enough information
$> ./sudoku "9...7...." "2...9..53" ".6..124.." "84...1.9." "5.....8.." ".31..4..." "........." ".9..5.741"

Yes  No

# Features

**Features testing**

Check out with the following sudokus the difficulty level:

$> ./sudoku "62.....45" ".5.32..1." "...4.8..." "78.5.2.63" "..4.8.5.." "56.9.3.82" "...2.9..." ".1..65.7." "87.....29"
$> ./sudoku "95..31.6." ".1....5.9" ".4.5....." "..1.683.." "........." "..314.8.." ".....6.3." "7.4....1." ".3.27..46"
$> ./sudoku "3..49...." ".82....." "7.....15" "..417268." "........." ".138465.." "83.....9" ".....73." "....61..8"
$> ./sudoku ".6.1..9.." "....5...2" "...2....4" "61.5..3.." ".3..6..5." "..2..8.71" "3....2..." "9...3...." "..1..5.9."
$> ./sudoku ".1...4..." "...5..9.." "68..1..2." ".....48." "578...263" ".24....." ".5..3..96" "..1..6..." "...2...1."

**Rate it from 0 (failed) through 5 (excellent)**

**Explanation**

Ask the shyest team member to explain how the code works.

If he can't explain it, no matter how other team members reply, leave No in the scale and carry on with the evaluation.

Proceed to asking a series of questions (try grilling the student who seems the most confused) : :

- Ask them about the algorythm they've used, without them being able to look at their code.
- Look for lines with bits transfers (<< and >>) and ask what they are.
- Look for complex lines and ask what it does.

- Ask them to explain recursivity.
- Which data structures ? And why ?
- Ask them what the return value is when the soduko has been filled ? What happens in the case of an impossible grid ?
- Check malloc returns and break their program if possible => if it breaks, select Crash.

- Any other question that comes to mind, be creative ! :)

If a student cannot answer a question at any given time, it's considered cheating, select Cheat.

If explanations are satisfactory, tick Yes.

If even one student cannot answer a question, tick No.

☑ Yes ☒ No

# Bonus

**Bonuses**

Error management 2.0

Check out the following errors:

- A grid with more than one solution (should return an error)

$> ./sudoku "9...7...." "2...9..53" ".6..124.." "84...1.9." "5.....8.." ".31..4..." "........." ".9..5.741" "47......."
-> If the program returns an error give 2 points, else stop here.

- A full grid (the program must react with reason)
$> ./sudoku "914375268" "287496153" "365812479" "846521397" "529637814" "731984526" "153749682" "698253741" "472168935"
-> If the program crashes, give 0 else give 2 points.
-> If the program returns an error stop here.
-> If the program returns a full grid, try with a full wrong grid.

$> ./sudoku "914375268" "287496153" "365812479" "846521397" "529637814" "731984526" "153749682" "698253742" "472168935"
-> If the program returns an error, give the last point.

**Rate it from 0 (failed) through 5 (excellent)**

## Ratings

**Don't forget to check the flag corresponding to the defense**

⊘ Forbidden function

## Conclusion

**Leave a comment on this evaluation**

Preview!!!