

## DBOJ Problem Setting Instructions [deprecated]

**IMPORTANT: This method of problem setting is now deprecated on the judge. Head to <https://dboj.jimmyliu.dev/about/problem-setting> to see the latest problem setting instructions.**

### Setting a problem

1. Ask me (jiminycricket#2701) to give you access to the file database (send me your gmail)
2. Head to the Google Cloud database:  
<https://console.cloud.google.com/storage/browser/discord-bot-oj-file-storage>
3. Upload the problem testdata to the `TestData` folder as a folder named the same as the problem name. Data files should be named `data` followed by the batch number, a period, and case number and either `.in` or `.out` as the extension (example, `data2.3.in` would be the input file for batch 2 case 3)
4. There should also be a `cases.txt` file in the `TestData` folder specifying the settings of the problem. The format of the content should be as follows:

```
[# cases in batch 1] [#cases in batch 2] ...  
[# points for batch 1] [# points for batch 2]... (should total 100)  
[time limit for python] [time limit for java] [time limit for c/c++]
```

5. Upload the problem statement to the `ProblemStatements` folder as a `.txt` file named the same as the problem name (ex. `helloworld.txt`). The contents of the file can be written in the standard Discord messaging markdown scheme.
6. Head to this website to register your problem to the database:  
<https://DBOJ-Problem-Adding-Web-App.jimmyliu3.repl.co>

### Hosting a contest

To host a contest:

1. Make the contest problems (problem names should end with the associated problem number in the contest. For example, problem 3 of a contest could be named `xxxp3`)

2. Add the contest instructions to the `ContestInstructions` folder saved to a text file with the name `[contestName].txt`. The contents of this file will be shown to the user upon joining the contest.
3. Head to the [repl.it](#) link above to set up the contest in the judge database
4. Use the `-set [contest1] [contest2]...` command to setup live scoreboards.
5. Ask me for permissions to send an announcement in the discord server

### Custom checkers

To feature custom grading in your problem, add a `checker.py` python program in the folder with the testdata. This program needs to read the current case input data from the file `data.in` and read the output provided by the user's program from `data.out` after processing this information, if you deem the output as correct, print `AC` to the standard output stream. Otherwise, if it is incorrect, you don't need to print anything. If the judge encounters a runtime-error while running the checker program, it will give a `Wrong Answer` verdict for the test case. If you use a custom checker, you do not need to include output files in your test data; just input files is sufficient.

For an example of custom checkers, see problem [ec4p2](#).