

# How to Use the System

## Read First: When Getting in Trouble with the System

Do not panic; stay calm. Raise your hand to call a staff member and tell them about your trouble. Here are a couple of phrases that may be useful in typical situations:

- “*We printed \_\_\_\_\_ (source code, debug output, ...) about \_\_\_\_ minutes ago but have not received the printout yet.*”
- “*\_\_\_\_\_ (browser, Eclipse, ...) does not start.*”
- “*Our \_\_\_\_\_ (computer, keyboard, mouse, ...) stopped working.*”

Staff will call the system team for you when they cannot help you by themselves.

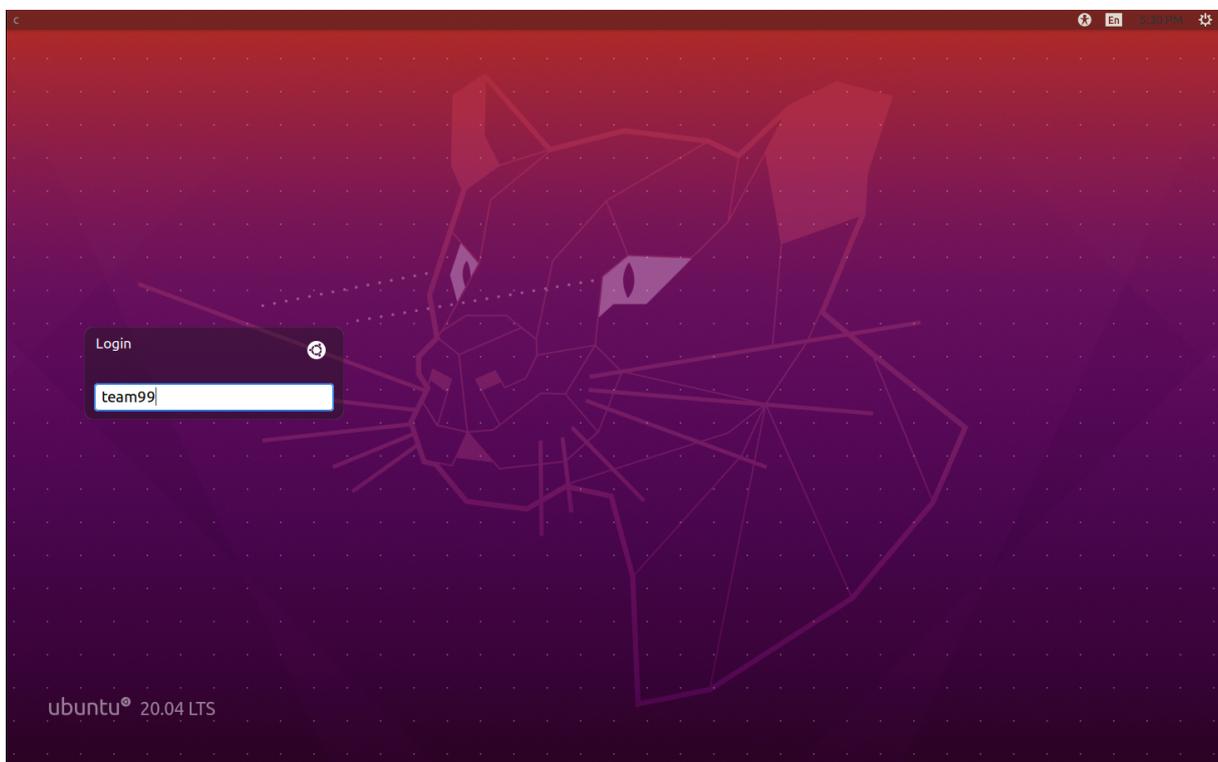
**Do not** use the Clarification for system troubles. You will be responded with “Contact the staff.”

## Logging In to Your Computer

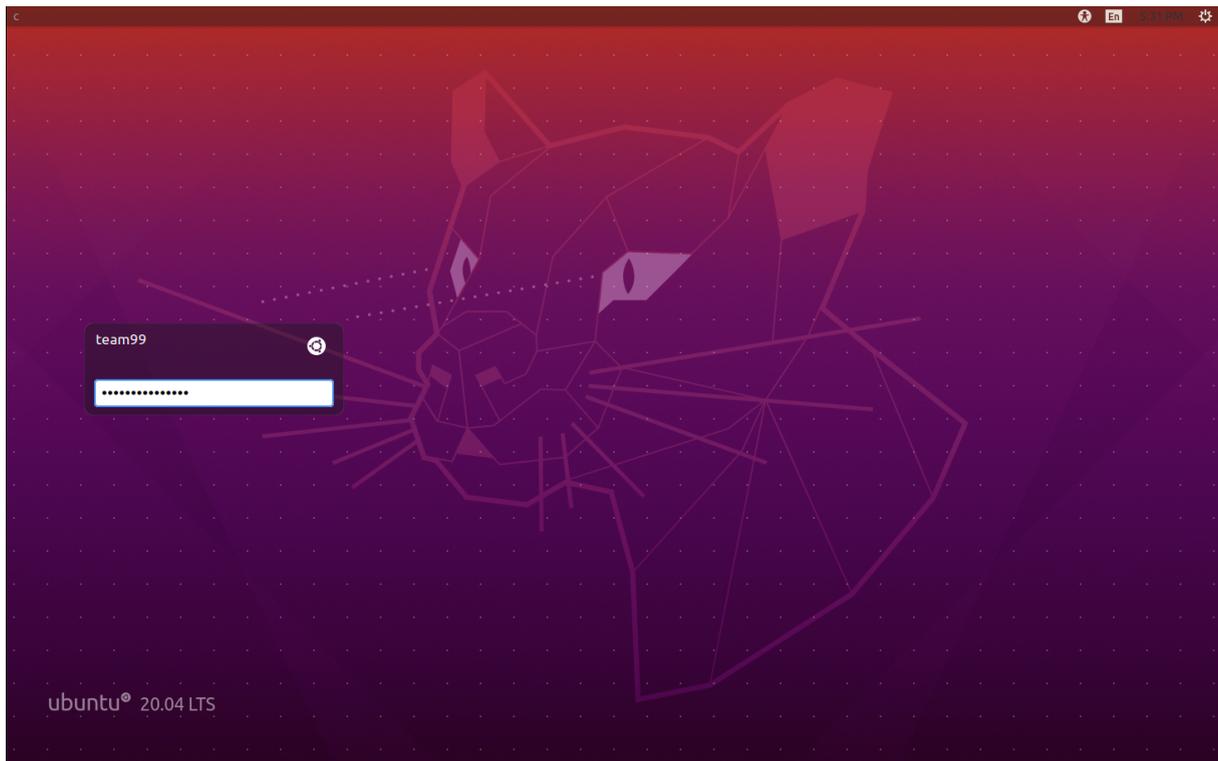
Your team ID and password are provided in a document in an envelope.

**Note:** You will use *the same team ID* but *different passwords* in the practice session (Dec. 27) and the main contest (Dec. 28). You will receive a document of login information each day.

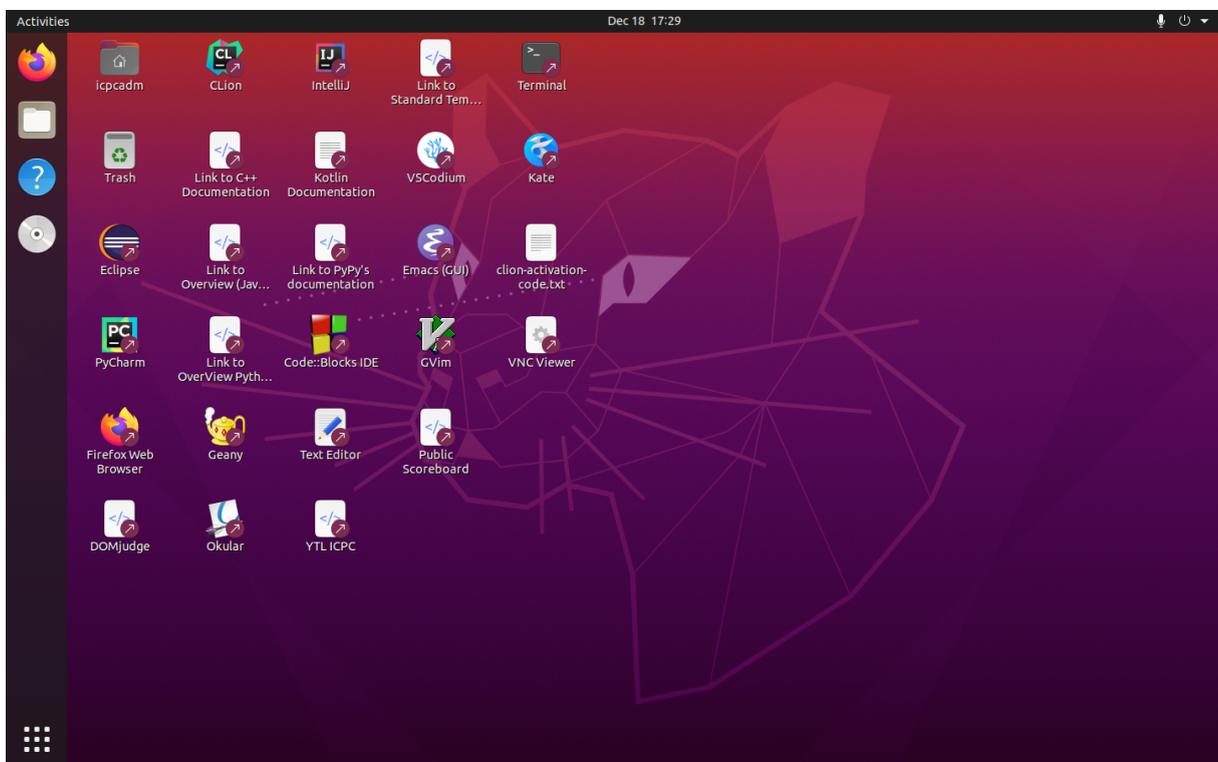
1. Type your team ID (`team01` or `team02` or ...) and press [Enter]:



2. Type your password and press [Enter]:



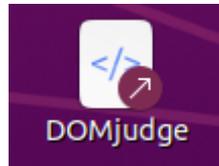
You should see a screen like this:



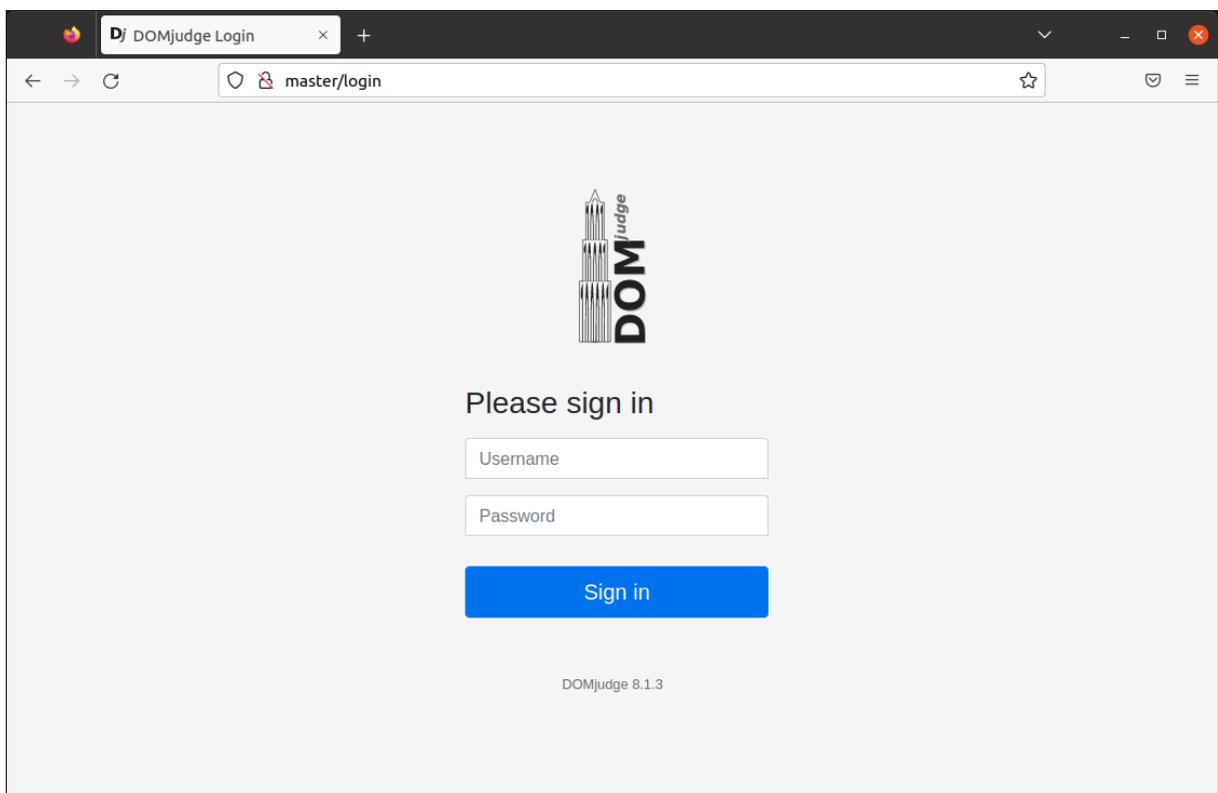
## Logging In to the Contest System

We use *DOMjudge 8.1.3* for the contest system this year.

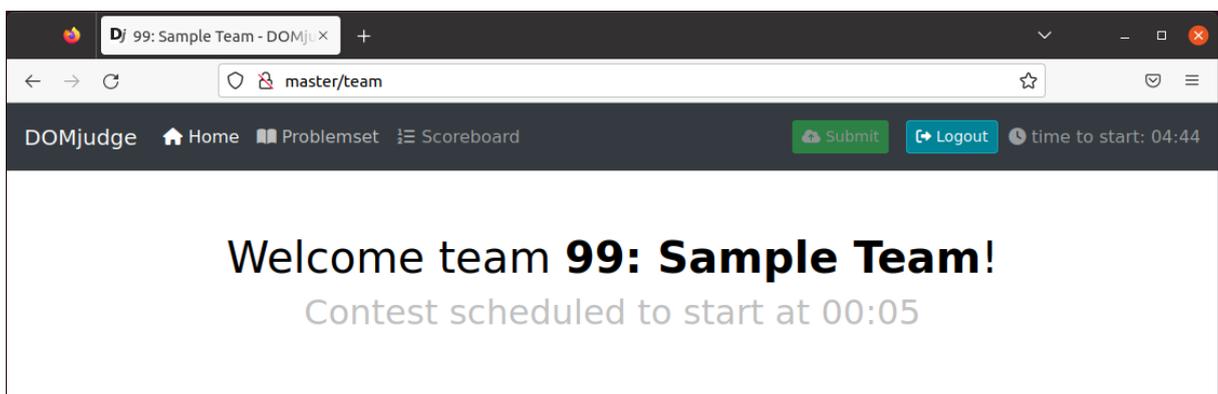
1. Double-click the DOMjudge shortcut on the desktop. It will open a new Firefox tab.



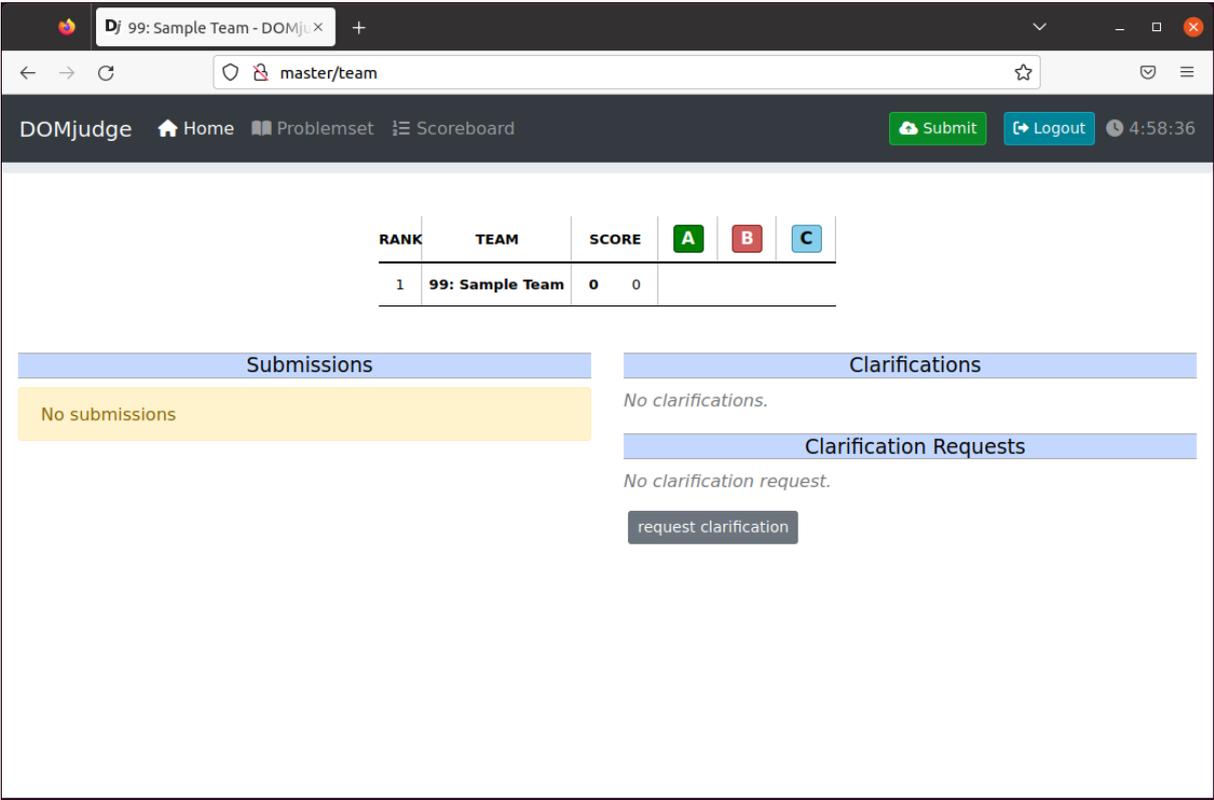
1. Type in your username and password, then click the [Sign in] button.



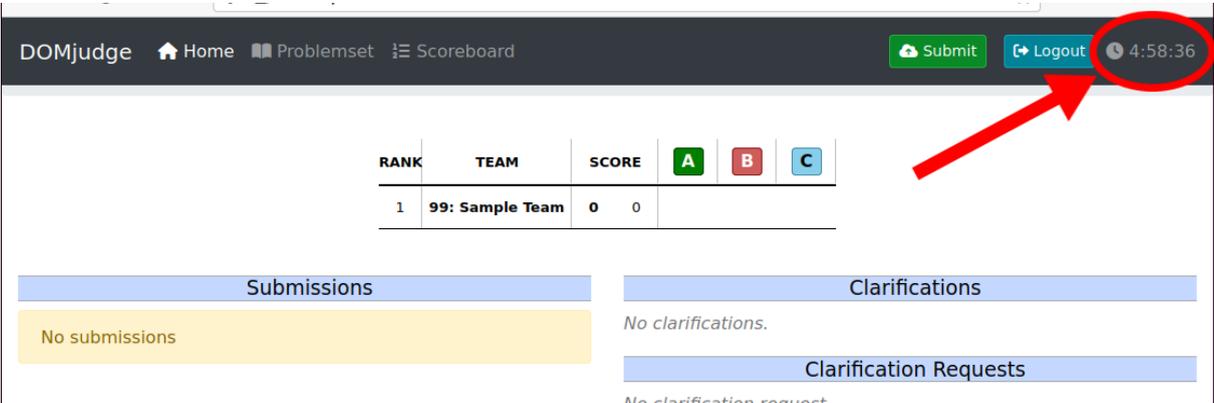
Once you get logged in, you will see a page like this:



Once the contest starts, you will see a page like this:

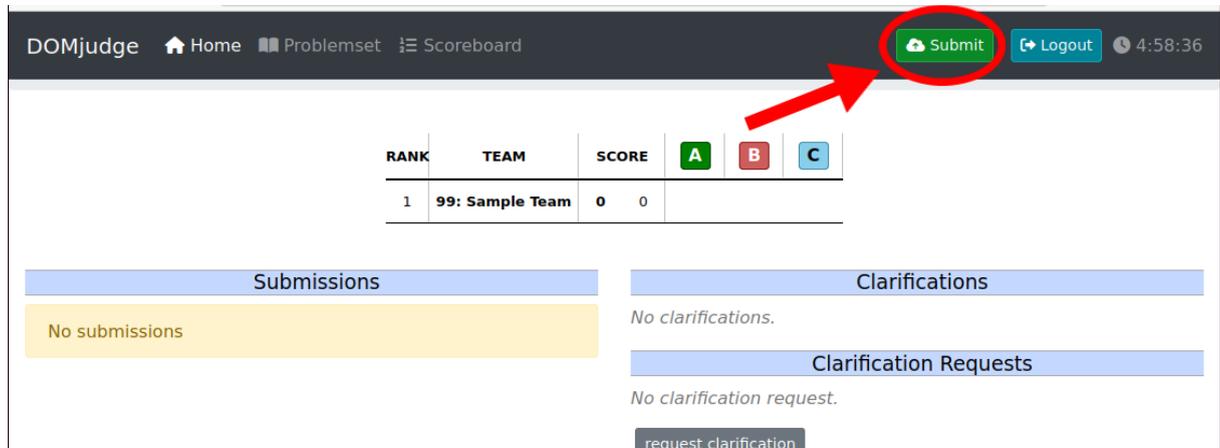


You can check the time to end at the upper-right corner:

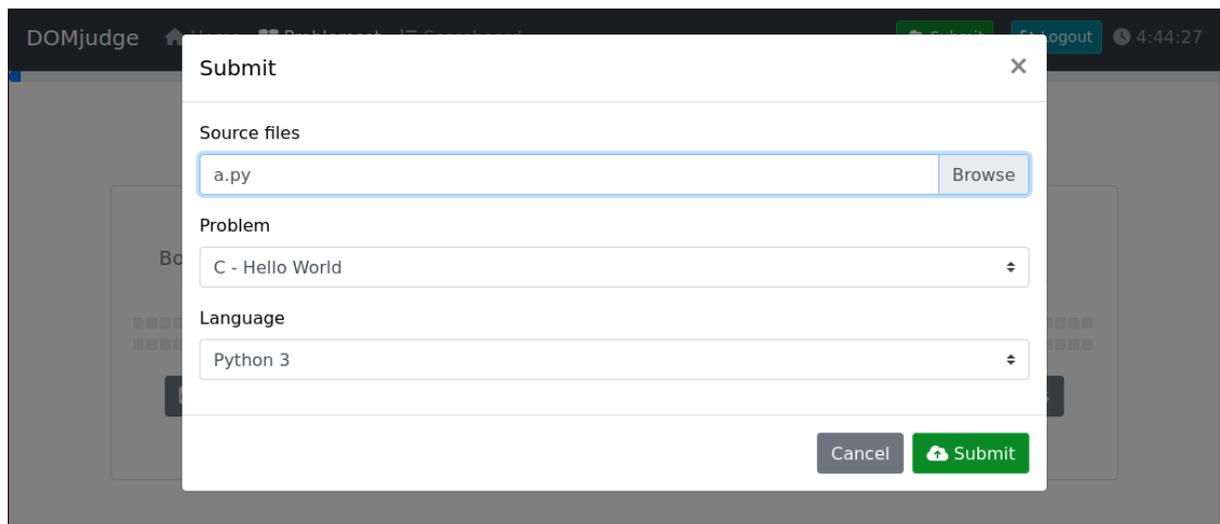


## Submitting Your Code

1. Click the [Submit] button near the upper-right corner. It will open the Submit dialog.



2. Choose your source files, problem, and language, then click the [Submit] button in the dialog.



Your submission results show up in the Submission section. You can see the details of each submission by clicking the corresponding row.

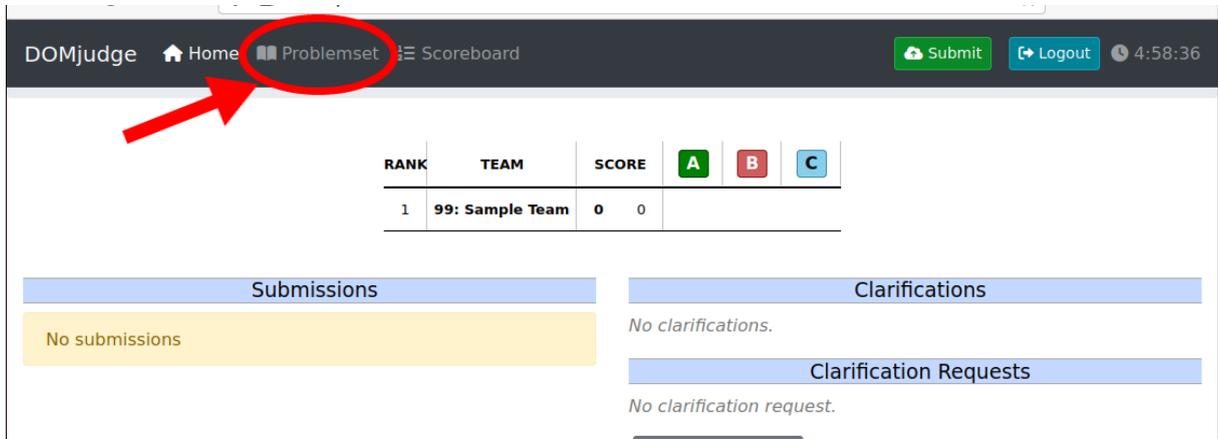
The Submission section also shows the submissions from your teammates. You can download the submitted code as a zip file there.

You can submit multiple source files: hold the [Ctrl] key down when choosing files.

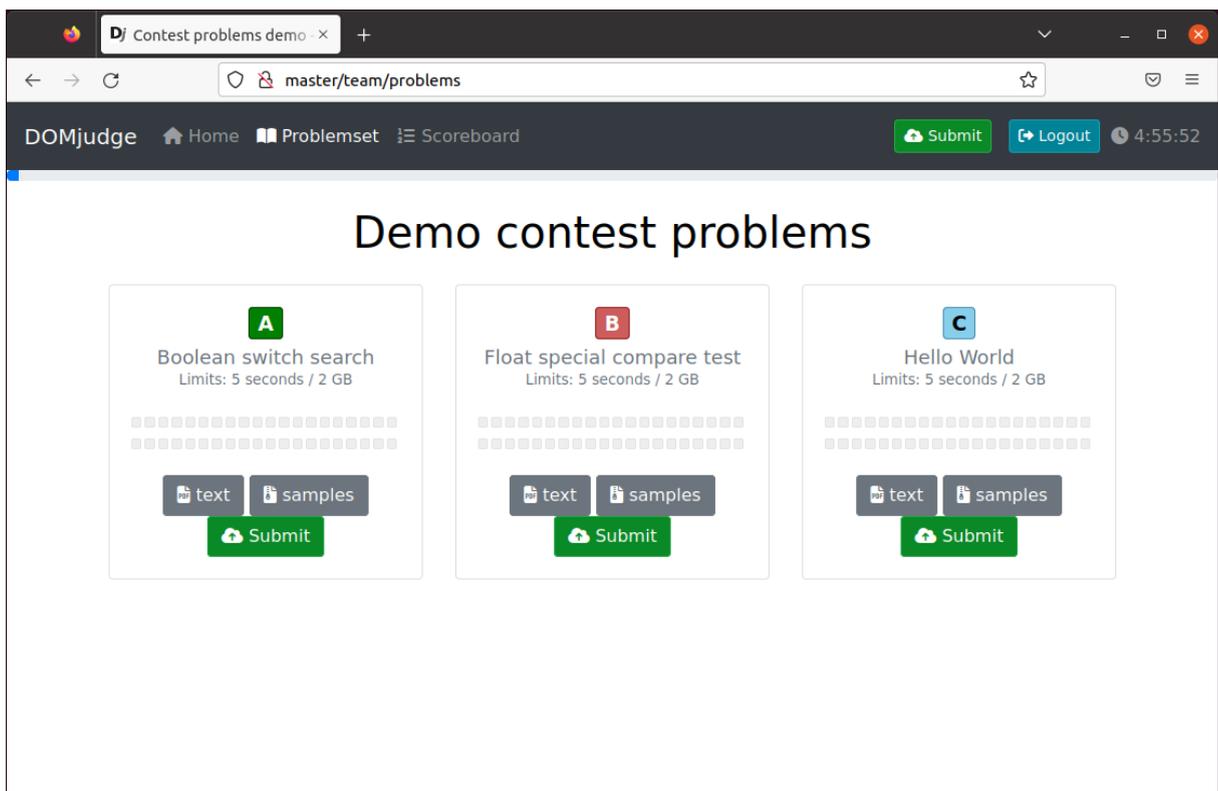
Filenames may contain only alphabets (A-Z, a-z), digits (0-9), and a few symbols (+, ., \_). They must start with an alphabet or a digit. In other words, they may not start with a symbol.

## Downloading Sample Input/Output

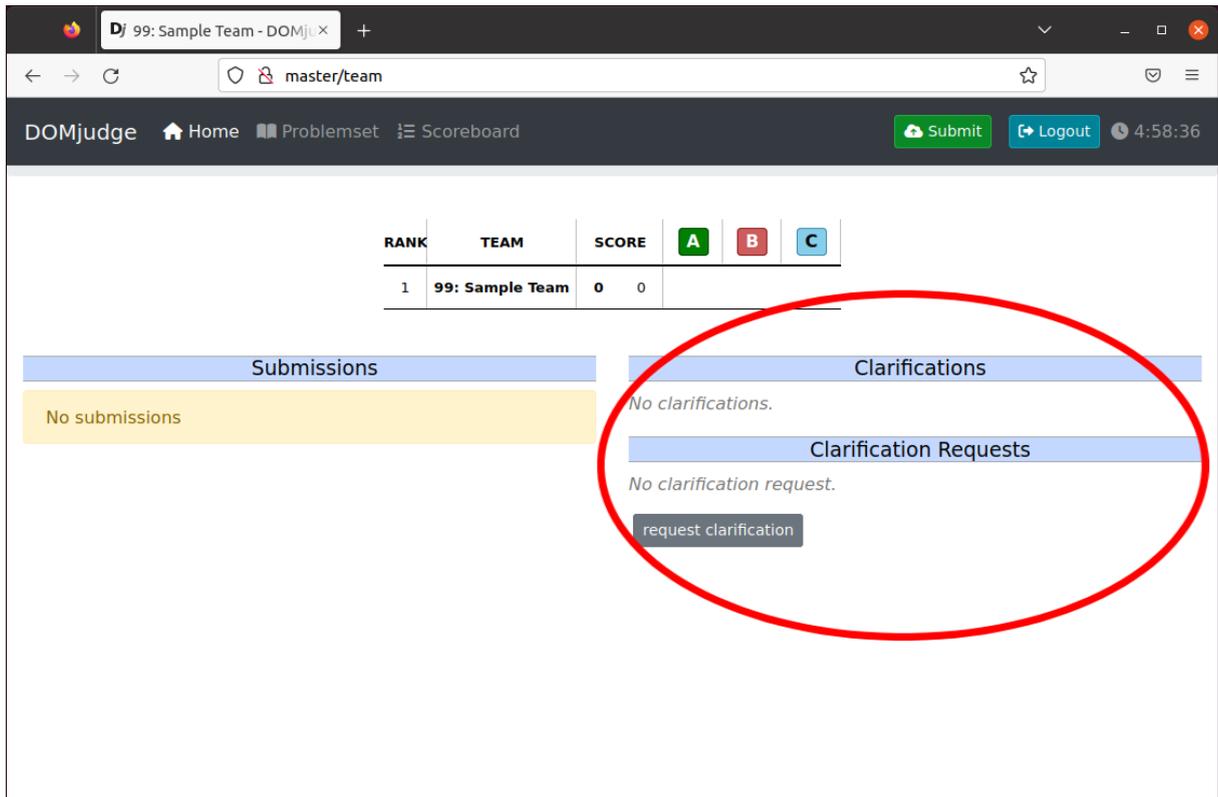
1. Click [Problemset] in the top navigation. It will open the Contest problems page.



2. Click the button of what you want to download.



## Communicating with the Judge



The screenshot shows the DOMjudge web interface. At the top, there is a navigation bar with 'DOMjudge', 'Home', 'Problemset', and 'Scoreboard' links, along with 'Submit' and 'Logout' buttons and a clock showing 4:58:36. Below the navigation bar is a scoreboard table:

RANK	TEAM	SCORE	A	B	C
1	99: Sample Team	0	0		

Below the scoreboard, there are three sections: 'Submissions' (showing 'No submissions'), 'Clarifications' (showing 'No clarifications.'), and 'Clarification Requests' (showing 'No clarification request.' and a 'request clarification' button). A red oval highlights the 'Clarifications' and 'Clarification Requests' sections.

On the right side is the only communication channel between you and the judge.

The Clarifications section shows messages from the judge. When the judge makes any announcements, they will show up in this section.

To ask a question to the judge:

1. Click the [request clarification] button. It will open the Send clarification request dialog.
2. Choose the Subject, write your question in the Message field, and click the [Send] button.

Your question will show up in the Clarification Requests section until it is answered. After you receive an answer, it will appear in the Clarifications section.

Be sure to write your question *in English* when you talk to the judge! Also, remember the judge **does not** answer system-related questions. If you have such questions, talk to the contest staff.

## Printing Your Code and Debug Output

To print a text file, such as your code and debug output, open a terminal and run:

```
print.sh $FILENAME
```

This command outputs “Print request was sent.” on success.

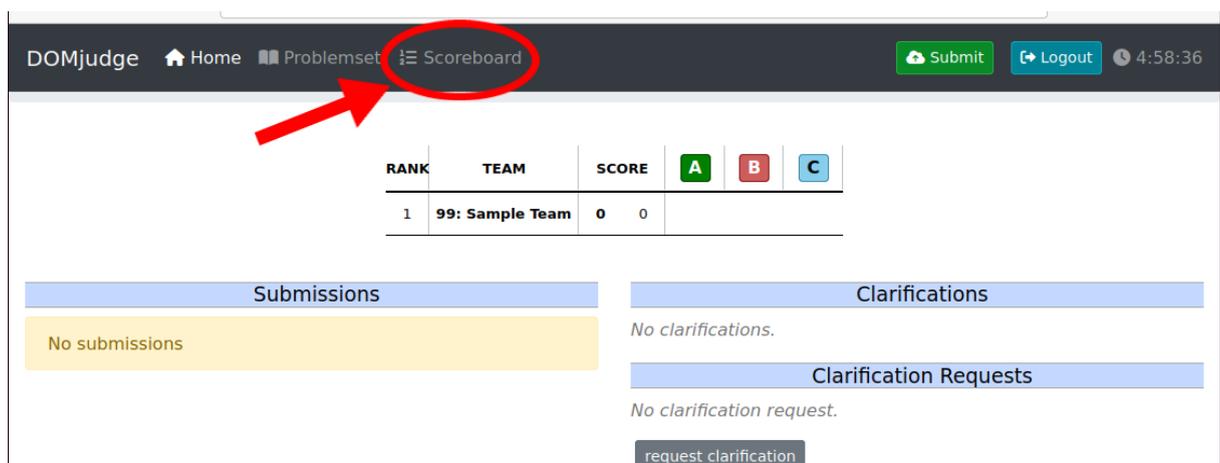
Your printouts will be delivered to your seat by staff. **Do not** attempt to pick them up by yourself; the printers are located in the staff area, which you cannot enter.

**Do not** use other methods, especially printing features in applications.

Only plain text is supported. **Do not** print anything else (e.g. PDF, HTML).

We do not have many printers. **Do not** attempt to print many pages. Printing the whole content of STL document or Java API document is a bad idea, for example.

## Viewing the Scoreboard



RANK	TEAM	SCORE	A	B	C
1	99: Sample Team	0 0			

Submissions: No submissions

Clarifications: No clarifications.

Clarification Requests: No clarification request.

request clarification

Click [Scoreboard] in the top navigation to show the current standings. Note the scoreboard will stop being updated (be “frozen”) 60 minutes before the contest ends.

## Writing and Compiling Your Code

The system is set up with a number of IDEs and editors. Use your favorite. There is a shortcut on the desktop for each of them.

You can test your programs with the following commands. Just pass your source file(s) or Java class file(s) as the arguments. See Appendix A for more detailed instructions.

- **C**                    `compilegcc` (then run `./a.out`)
- **C++**                  `compileg++` (then run `./a.out`)
- **Java**                 `compilejava / runjava`
- **Python 3**            `compilepython3 / runpython3`
- **Kotlin**              `compilekotlin / runkotlin`

## Customizing the System

We strongly discourage you from changing system settings of the computer. If you choose to do so, you are responsible for all troubles it could cause, i.e. you will be doing it at your own risk.

Also be advised the system will be reset to the original state between the practice and the main contest. All files and customizations made during the practice will be lost.

## Taking Out Your Files after the Contest

If you would like to take your files out of the computer after the contest, please put them in:

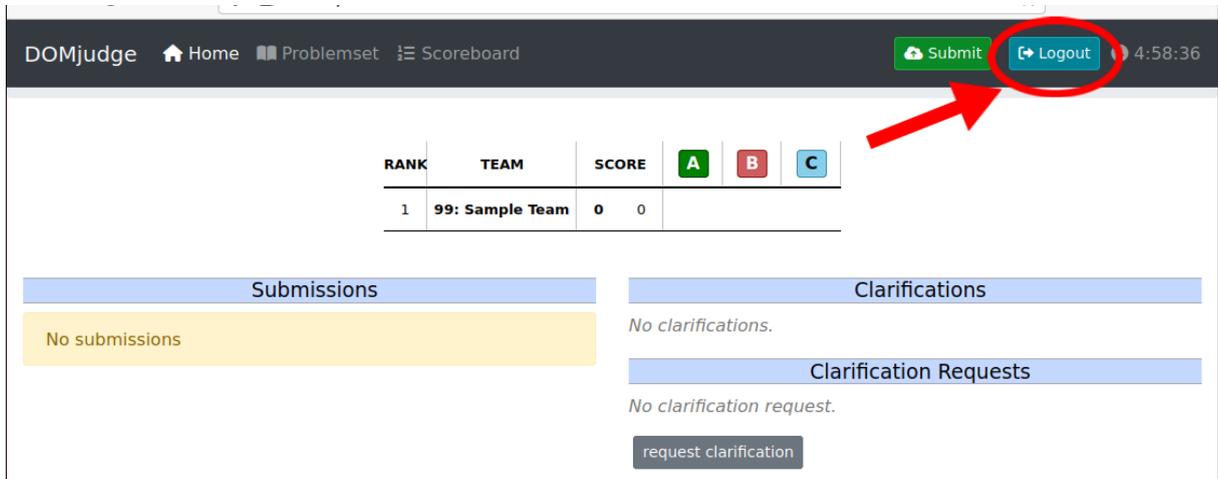
```
~/takeout
```

We will pick up the files there and send them to you by email. We will give you some time to copy the files after the contest is over, so that you do not have to care about it during the contest.

See Appendix B if you need step-by-step instructions.

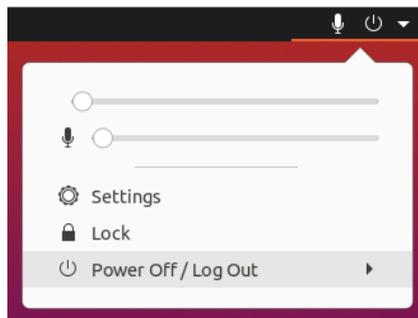
## Logging Out from the Contest System

1. Click the [Logout] button at the upper-right corner of the page.

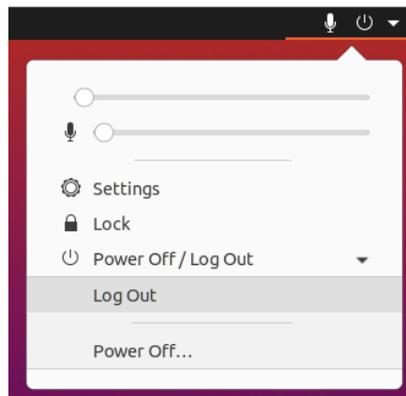


## Logging Out from Your Computer

1. Click  (down-pointing triangle) at the upper-right corner of the screen.
2. Click your team name to open the submenu.



3. Select [Log Out] from the shown menu.

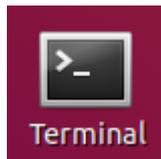


## Appendix A: How to Compile and Run Your Code

### C/C++

We assume that you save your source file as `A.c` or `A.cpp` and input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.



2. Type as follows in the Terminal window (`↵` denotes [Enter]), depending on whether you are using C or C++. It compiles your code and generates a binary named `a.out`.

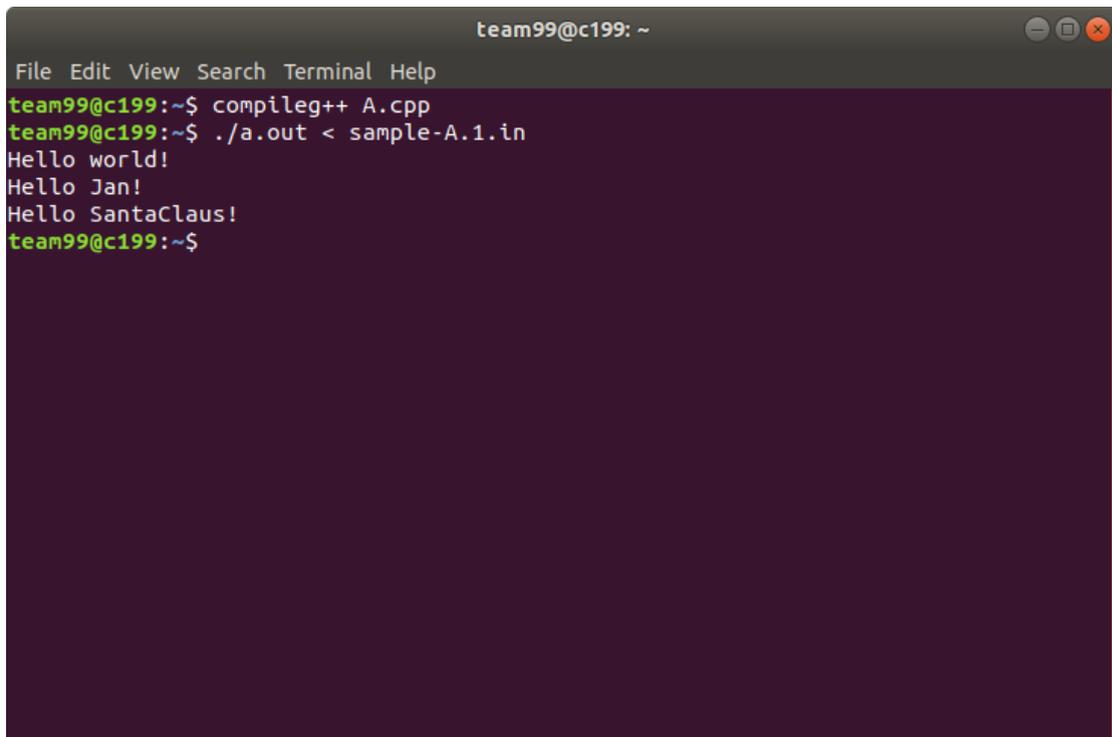
```
C:    compilegcc A.c↵  
C++:  compileg++ A.cpp↵
```

3. Type as follows to run your program:

```
./a.out < sample-A.1.in↵
```

If you would like to save the output into a file named `my-A.1.out`:

```
./a.out < sample-A.1.in > my-A.1.out↵
```

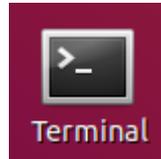
A screenshot of a terminal window titled "team99@c199: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal shows the following commands and output:

```
team99@c199:~$ compileg++ A.cpp  
team99@c199:~$ ./a.out < sample-A.1.in  
Hello world!  
Hello Jan!  
Hello SantaClaus!  
team99@c199:~$
```

## Java

We assume that you save your source file as `A.java` and input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.



2. Type as follows in the Terminal window (`\n` denotes [Enter]). It compiles your code into Java class files (e.g. `A.class`).

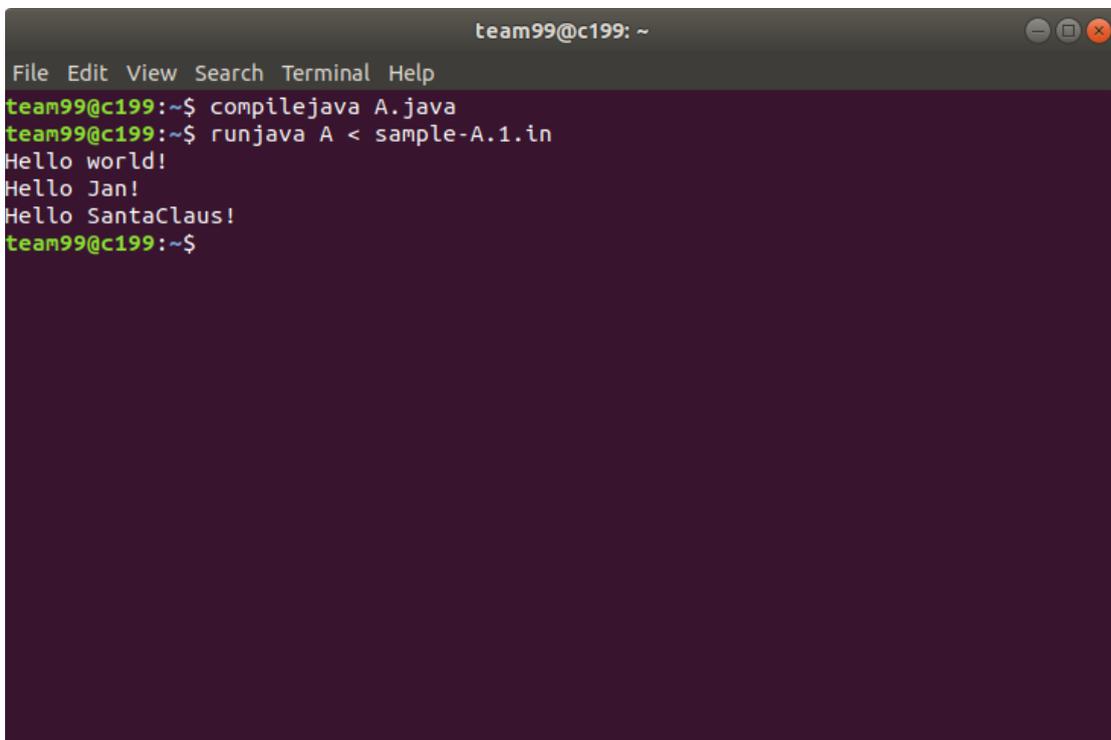
```
compilejava A.java\n
```

3. Type as follows to run your program:

```
runjava A < sample-A.1.in\n
```

If you would like to save the output into a file named `my-A.1.out`:

```
runjava A < sample-A.1.in > my-A.1.out\n
```

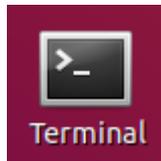
A screenshot of a terminal window titled "team99@c199: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal shows the following commands and output:

```
team99@c199:~$ compilejava A.java
team99@c199:~$ runjava A < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

## Python

We assume that you save your source file as `A.py` and input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

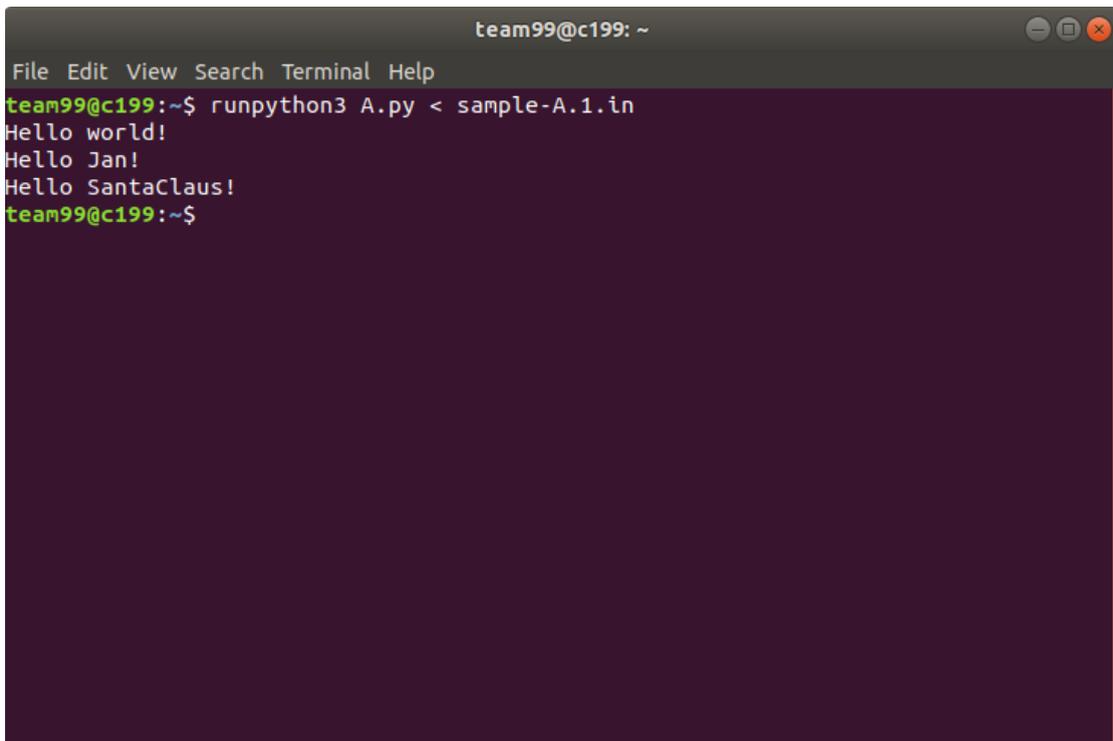
1. Double-click the Terminal shortcut on the desktop.



2. Type as follows in the Terminal window to run your program (`↵` denotes [Enter]):  
`runpython3 A.py < sample-A.1.in↵`

If you would like to save the output into a file named `my-A.1.out`:

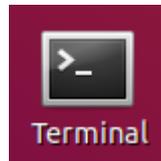
`runpython3 A.py < sample-A.1.in > my-A.1.out↵`

A screenshot of a Linux terminal window. The title bar shows "team99@c199: ~" and standard window controls. The menu bar includes "File Edit View Search Terminal Help". The terminal content shows the command `runpython3 A.py < sample-A.1.in` being executed, resulting in three lines of output: "Hello world!", "Hello Jan!", and "Hello SantaClaus!". The prompt `team99@c199:~$` is visible at the end of the output.

## Kotlin

We assume that you save your source file as `A.kt` and input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.



2. Type as follows in the Terminal window (`\n` denotes [Enter]). It compiles your code into Java class files (e.g. `AKt.class`).

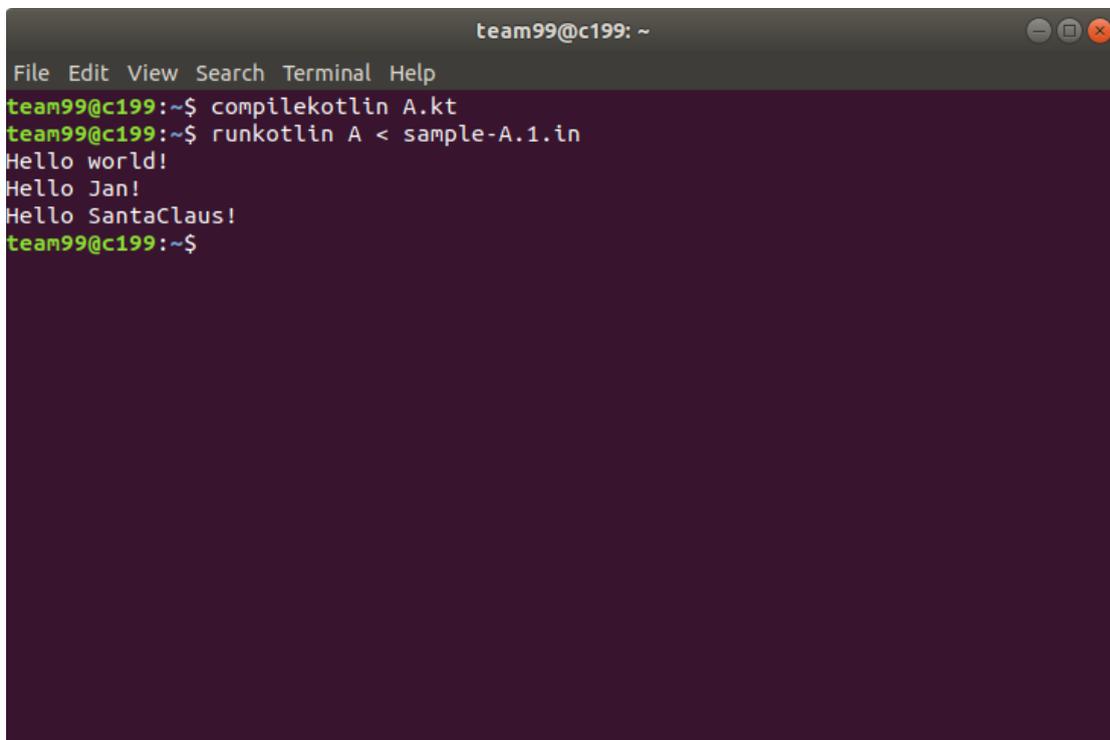
```
compilekotlin A.kt\n
```

3. Type as follows to run your program:

```
runkotlin A < sample-A.1.in\n
```

If you would like to save the output into a file named `my-A.1.out`:

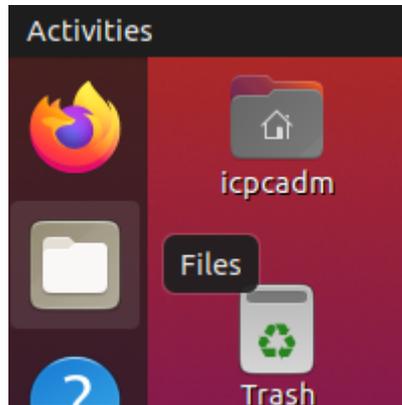
```
runkotlin A < sample-A.1.in > my-A.1.out\n
```

A screenshot of a terminal window titled 'team99@c199: ~'. The window has a dark grey title bar with standard window controls (minimize, maximize, close) on the right. The terminal content shows the following commands and output:

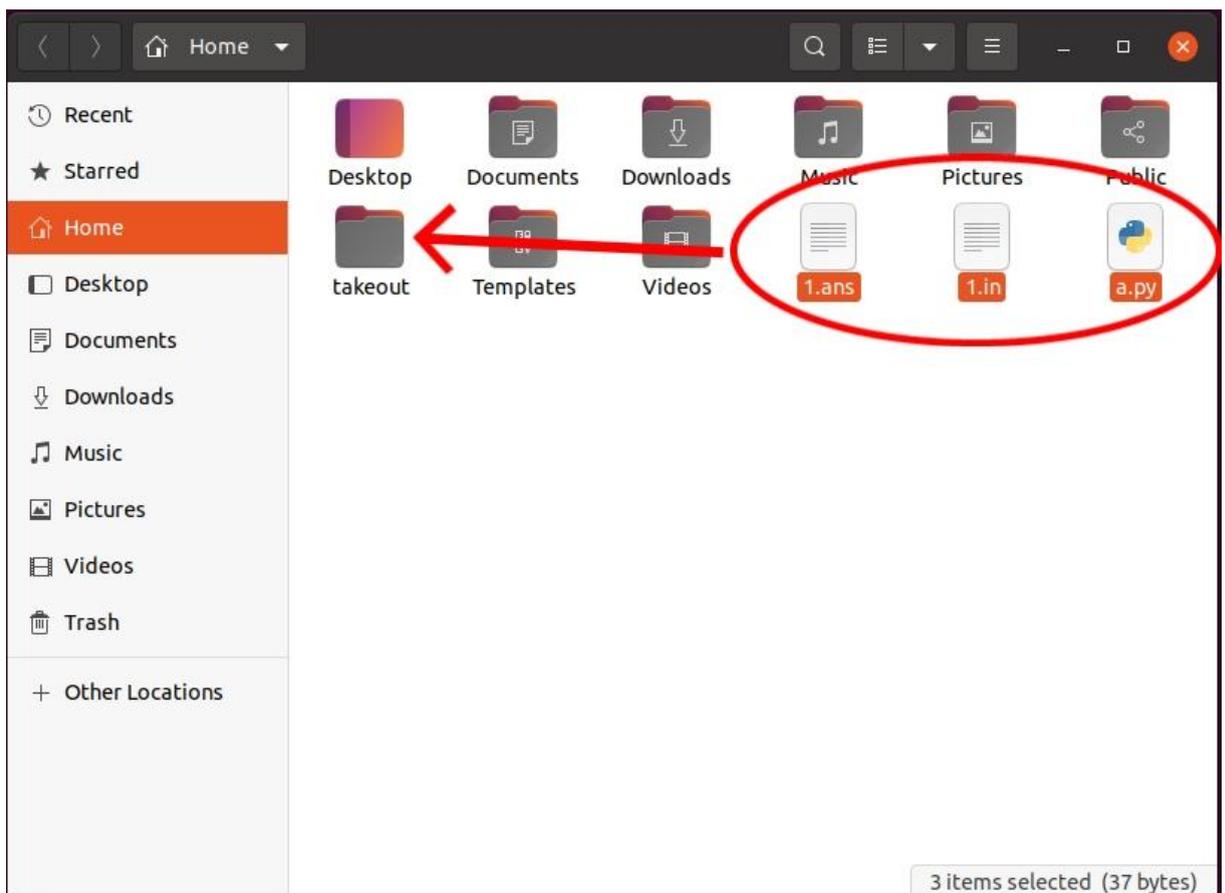
```
File Edit View Search Terminal Help
team99@c199:~$ compilekotlin A.kt
team99@c199:~$ runkotlin A < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

## Appendix B: How to Move Your Files to Take Out

1. Click [Files] in the launcher (on the left side of the screen) to open your home folder.



2. Select your files to copy. You can use [Shift] and [Ctrl] keys as you do on Windows/Mac.<sup>1</sup>
3. Drag and drop them into the takeout folder.



<sup>1</sup> For Mac users: [Ctrl] behaves like [command] here.