



<https://onlinejudge.u-aizu.ac.jp/>

## Tutorial

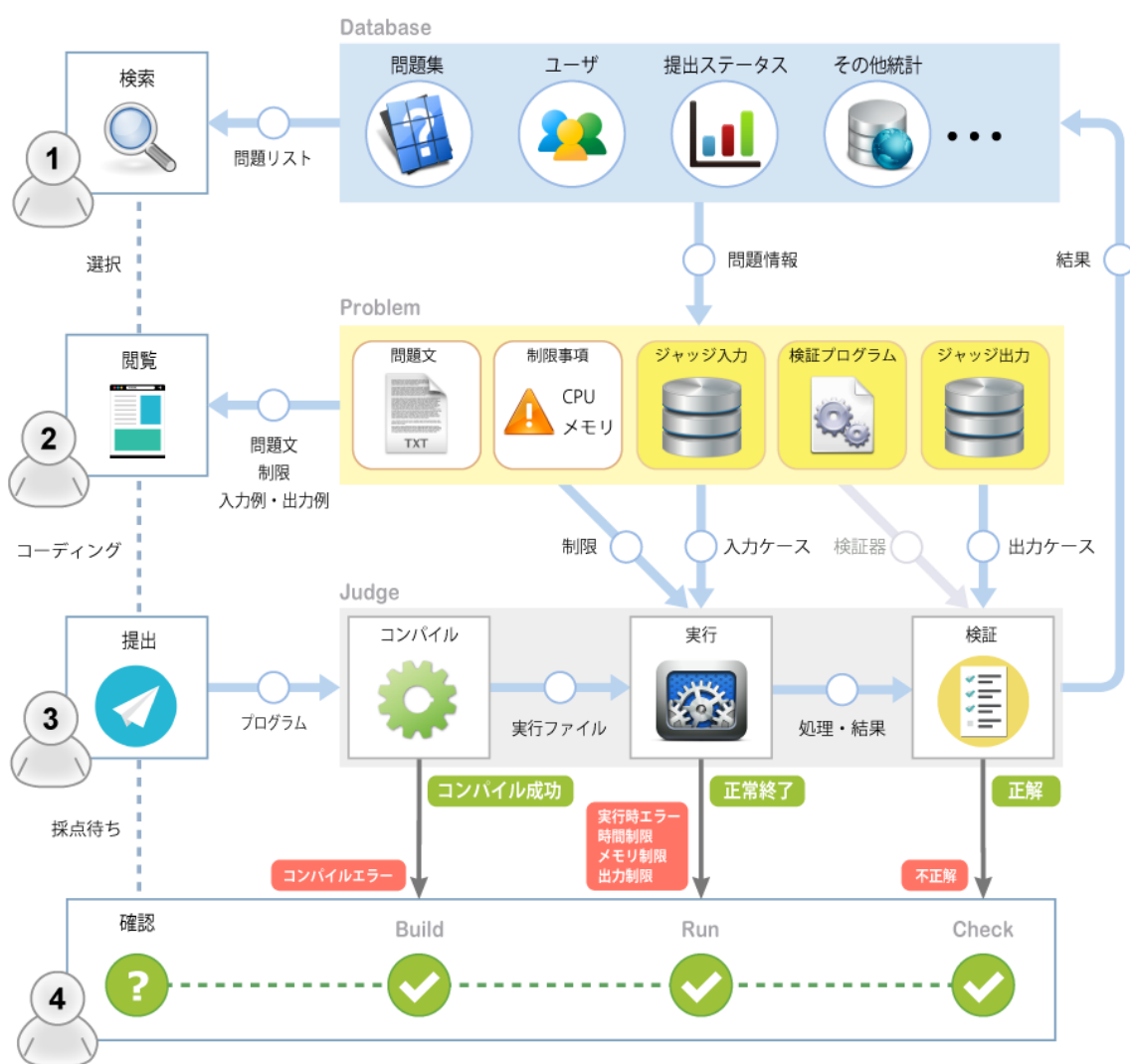
In this document, a short explanation for  $\beta$  version of AOJ is presented. The old version of AOJ is also available from <http://judge.u-aizu.ac.jp> .

# 1. Introduction

AOJ (Aizu Online Judge) is an online judge system which is available for free and automatically grades your program online. It is oriented towards students as well as engineers.

## 1-1. What is Online Judge System?

An online judge provides a lot of problems and automatically judges your source codes online. For each problem, very strict test cases and the judge provides the verdict to check your program is correct and efficient enough for the problem specification. Generally, an online judge system is organized as follows.



## AOJ Tutorial

Through the online judge system, the users can obtain the following capabilities:

- Read the description and understand the specification correctly.
- Design algorithms based on estimation of time/space complexity considering computational resources.
- Implement the algorithm without bugs.
- Obtain a wide range of algorithms and data structures.

Anyway, you can learn programming like a game for fun. The repetitive practice based on the problem set makes you stronger.

Since it includes fundamental tasks for novice programmers, it can be employed as educational materials for schools to teach programming and algorithms.

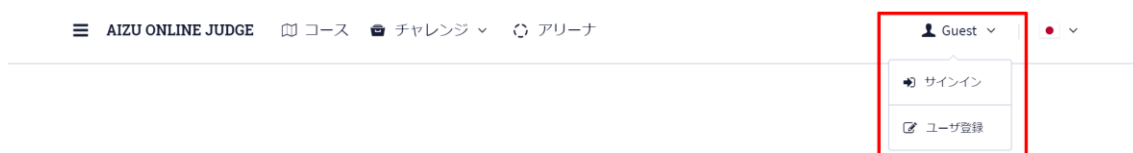
### 1-2. Aizu Online Judge

**Aizu Online Judge (AOJ)** (<https://onlinejudge.u-aizu.ac.jp/>) is an online judge system which is developed and operated by University of Aizu. It includes more than 2000 problems related to introduction to programming as well as to programming competition such as ICPC, JOI, PCK and many voluntary contests.

## 2. Registration

### 2-1. Sign Up

To submit your codes to AOJ, you need to sign up. Let's go "Guest" -> "Sign Up" from the header.



The page for user registration is as follows.

### ユーザ登録をする前に...

利用規約に関する説明と注意書きをお読み下さい : [利用規約について](#)

アカウントID*:	<input type="text"/>	①	半角アルファベット、半角数字、半角記号
パスワード*:	<input type="password"/>		4文字以上 (Do not use special characters like \ and ')
パスワード(確認)*:	<input type="password"/>		
-----			
名前*:	<input type="text"/>	②	半角アルファベット、半角数字、半角記号、空白
アバター:	<input type="file"/>	③	
所属:	<input type="text"/>	④	半角アルファベット、半角数字、半角記号、空白
在住または出身国*:	<input type="text" value="Japan"/>		
生まれた年:	<input type="text"/>		
メールアドレス:	<input type="text"/>		
URL:	<input type="text"/>		あなたのホームページやBLOG

Before you sign up, please read “Terms of Use” which describes notes for submission and treatment the submitted codes. You should input the following items to sign up.

- ① **Account ID** : ID to Sing In. This is the unique ID as a AOJ user.
- ② **Name** : This is a display name for your status, leader board, etc. It is like nickname.
- ③ **Avatar** : (\* this is option) This is an icon to be shown in your status page.
- ④ **Affiliation** : This is the name of our school, organization, etc.
- ⑤ **Display Language** : This is the language to be used for menu (Japanese or English).
- ⑥ **Programming Language** : This is your preferable programming language. When you submit source code, the specified language will be set on by default.
- ⑦ **Policy** : If you select public, all your source codes submitted to AOJ are opened to the public. If you select private, only you can access the submitted codes. We recommend to select “public” for your training.

After you fill all the required items, you can send the information. If you receive a message “Welcome to AOJ!”, the registration is successful.

## 2-2. Sign In

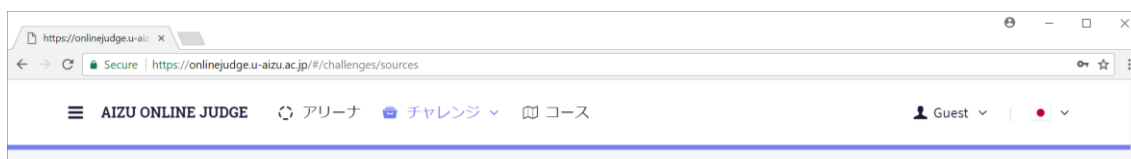
To make full use of AOJ, you need to Sign In (login) to it. Let's go to "Guest" -> "Sign In" to open the dialog and input your user ID and password. After you sign in, you will be able to:

- Check your progress through check marks, scores and progress bars in views for searching and browsing
- Browse your source codes submitted
- Put bookmarks to each problem
- Do other activities for recommendation, tagging, etc.

You can also update your registered information and password after you sign in.

## 3. Menu

The header of AOJ consists of:



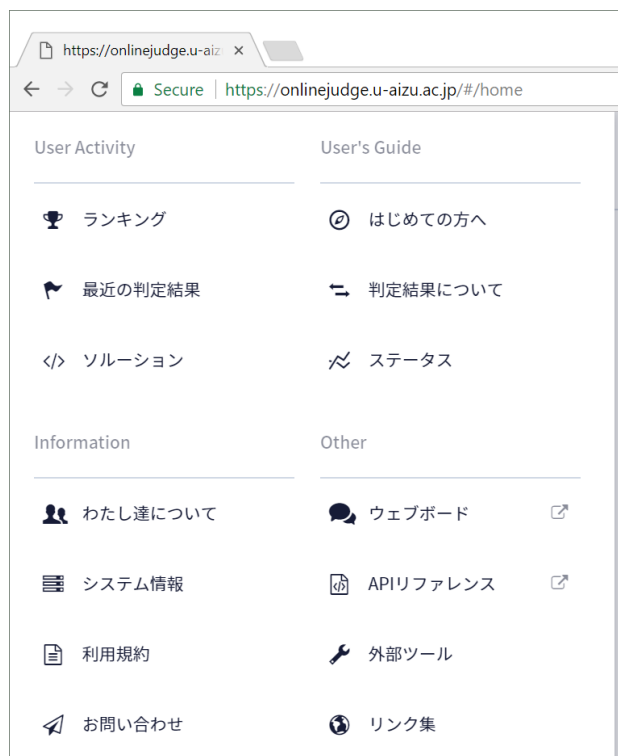
You can choose the display language for menu (Japanese or English).

AOJ consists of three items as follows:

- **Course** to obtain basic skills for programming
- **Challenge** to solve past problems given programming contests
- **Arena** to participate exercises, virtual competitions and real programming contests.

## AOJ Tutorial

You can open the menu from the icon placed at the left corner of the header.



The menu consists of four sections: **User Activity**, **User's Guide**, **Information**, and **Other**, each of which is a link to the corresponding page.

**User Activity** provides statistics and other information updated by daily activities of registered users. The users are ranked by different criteria such as rating and the number of solved problems.

**User's Guild** provides information for new comers. For example, details of verdicts from the judge are explained.

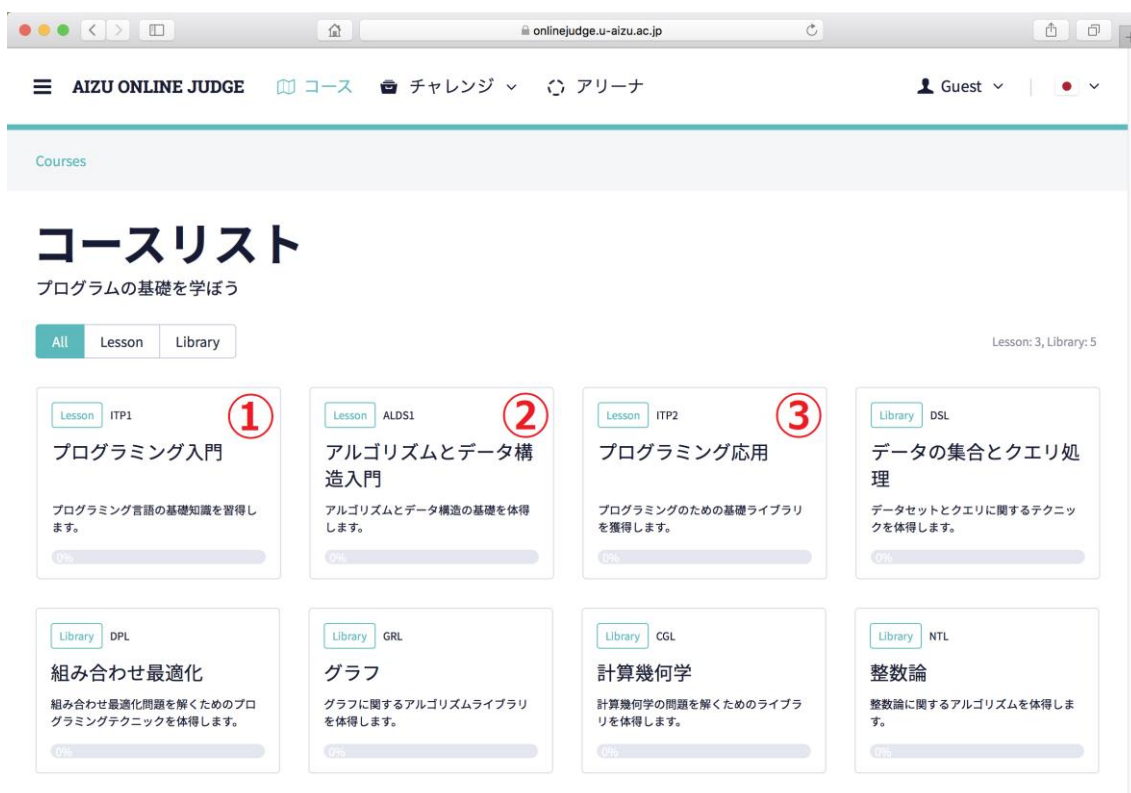
**Information** and **Other** provide details of the system, compilers as well as API for developers as technical information.

## 4. Search for Problems

### 4-1. Course Problem

As the header shows, problems are classified into Course and Challenge which provide different interfaces for searching.

**Course** consists of problems related to introduction to programming as well as algorithms and data structures. They are oriented towards users who want to learn basic syntax and typical knowledge. The Course includes several courses each of which is represented by a card. You can see your progress on the card if you are signing in.

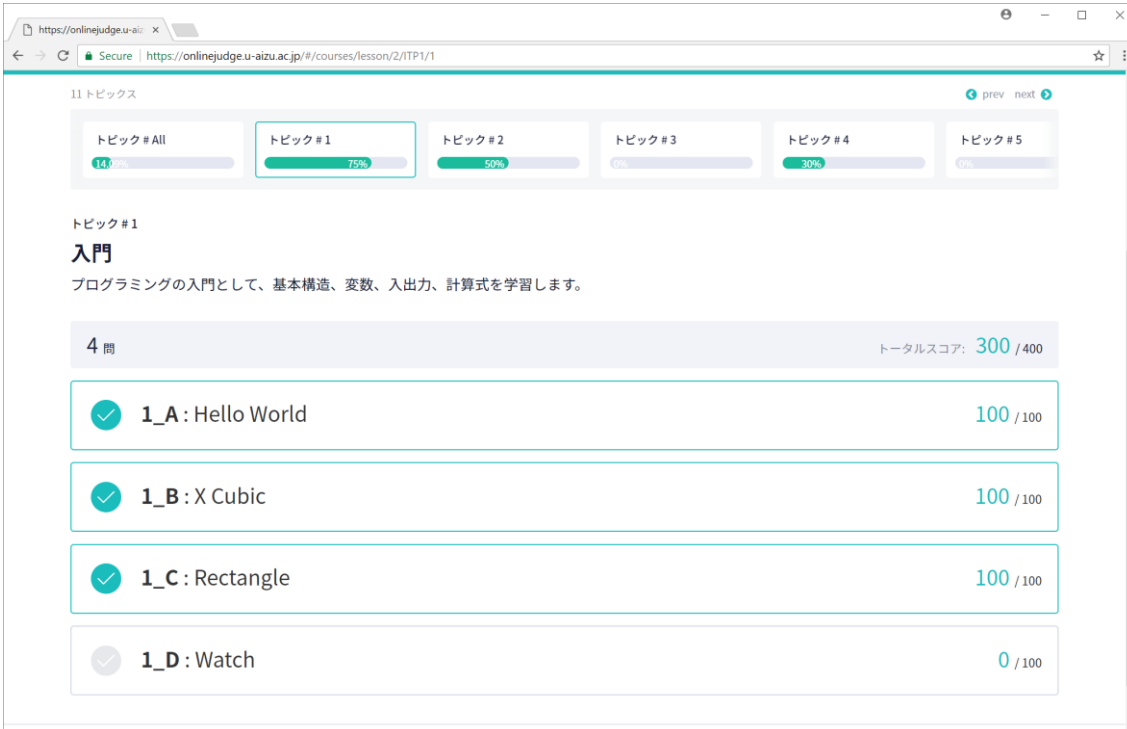


- ① **Introduction to Programming** consists of problems to check basic syntax and oriented toward beginners for programming.
- ② **Algorithms and Data Structures** consists of problems to obtain basic knowledge of algorithms and data structures. This course is oriented toward learners who are familiar with programming languages and want to obtain advanced programming skill.
- ③ **Introduction to Programming II** is oriented to learners who understand the basic algorithms and want to learn how to use standard libraries and general programming techniques.

## AOJ Tutorial

Other courses are to obtain classic and advanced algorithms data structures. They are oriented to advanced programmers.

The course is disclosed by selecting the card.



The screenshot shows a web browser window displaying the AOJ Tutorial course interface. At the top, there are navigation buttons for 'prev' and 'next'. Below this, a progress bar shows the completion status for 11 topics. The current topic, 'トピック #1', is highlighted with a green border and shows 75% completion. Below the progress bar, the title 'トピック #1 入門' is displayed, followed by a description: 'プログラミングの入門として、基本構造、変数、入出力、計算式を学習します。'. A summary bar indicates '4 問' (4 questions) and a 'トータルスコア: 300 / 400' (Total score: 300 / 400). The list of problems is as follows:

Problem ID	Problem Name	Score
1_A	Hello World	100 / 100
1_B	X Cubic	100 / 100
1_C	Rectangle	100 / 100
1_D	Watch	0 / 100

Each course consists of several topics. You can see your progress for each topic. A topic consists of several problems each of which is clickable to be disclosed.



## 5. Problem Solving

### 5-1. Problem Description

You can visit the page of problem description through the cards (links) in finder and search page.

The screenshot shows the AIZU ONLINE JUDGE interface for problem 'PCK-Prelim 0358'. The page is annotated with red boxes and circled numbers 1-1 through 1-5. 1-1 points to the 'Time Limit: 1sec, Memory Limit: 262144 KB, isSolved: [dropdown]' section. 1-2 points to the problem title '買い物' and its description. 1-3 points to the 'Input' section. 1-4 points to the 'Output' section. 1-5 points to the 'Sample Input 1' and 'Sample Output 1' sections.

Time Limit: 1sec, Memory Limit: 262144 KB, isSolved: [dropdown] ①-1

### 買い物 ①-2

あなたは、今日発売された新刊「プログラミング甲子園必勝法」を買うために、友達のA君と本屋に来ています。あなたはどうしてもその本を買いたいので、手持ちの金額で足りないときは、A君からお金を借りようと思っています。それでも足りなければ、今回はあきらめるしかありません。

あなたの手持ちの金額、A君の手持ちの金額、本の値段が与えられたとき、本を買うためにA君から借りなければならない最小の金額を出力するプログラムを作成せよ。

#### Input ①-3

入力は以下の形式で与えられる。

```
m f b
```

1行に、あなたの手持ちの金額  $m$  ( $0 \leq m \leq 10000$ ) とA君の手持ちの金額  $f$  ( $0 \leq f \leq 10000$ ) と本の値段  $b$  ( $100 \leq b \leq 20000$ ) が与えられる。

#### Output ①-4

本を買うためにA君から借りなければならない最小の金額を1行に出力する。ただし、A君の手持ちの金額をすべて借りても本を買えないときは、"NA" と1行に出力する。

#### Sample Input 1 ①-5

```
1000 3000 3000
```

#### Sample Output 1

```
2000
```

#### Sample Input 2

```
5000 3000 4500
```

#### Sample Output 2

```
0
```

The screenshot shows an online judge interface with the following components:

- 2-1**: Language dropdown menu set to C++.
- 2-2**: Code editor area containing C++ source code for a problem involving finding the maximum of two numbers.
- 2-3**: Submit and Clear buttons.
- 3**: Progress bar showing steps: Submitted, Sent to Judge, Build, Run, Resource Limit Check, Result Check, Presentation Check. Below it is a table with columns: Lang, Status, Judge, Time, Memory, Code. The table currently shows "No Data".
- 4**: Test case table with columns: Case #, Status, Time, Memory, In, Out, caseName. The table currently shows "No Data".
- 5**: Footer area with links for Web Board, Solutions, Statistics, and Tagging. Below the links, it says "source : 15th PC Koshien, Preliminary Round" and "created at : 1/1/2018, 3:24:53 AM".

The description page consists of the following item:

- ① **Problem Description** : Problem statements are given
- ② **Coding Form** : A working area for coding (or just paste source code) and submit the code to the judge
- ③ **Status View** : You can check verdict from the judge on this view after you submit the source code
- ④ **Test Case View** : After the judge is finished, you can check the verdict and the judge data for each test case on this view
- ⑤ **Footer** : Links related to statistics and auxiliary information are provided

### 3-2. Contents of Problem Statement

Each problem is in the form “please write a program which output correct data for given input meeting the specification”. More concretely, the statement consists of:

- ①-1 **Header** : includes limitations for the problem.  
Time Limit and Memory Limit are limits of CPU time and memory usage respectively. If CPU time / memory exceeds the limit, your program is rejected by the judge. This means that you need to improve your algorithms.
- ①-2 **Description** : includes problem statements. The specification of the problem is given with a background story.
- ①-3 **Input** : provides the specification of input format. Your program need to read the input data according to the format described here. Note that your program should read the data from the standard input.
- ①-4 **Output** : provides the specification of output format. Your program need to write the output data according to the format described here. Note that your program should write the data to the standard output.
- ①-5 **Sample Input X, Sample Output X** : A sample input is an example of data which is given to your program for the judge. It is to confirm the input format more concretely. The sample output shows a correct answer for the corresponding sample input. So, note that a program which is correct for the sample input/output is not always accepted. Your program is executed with strict data for the judge.

### 3-3. Submission of Program

②Coding Form is to submit your code and consists of the following item:

- ②-1 **Programming Language** : Select a language of your code. The language based on your preference is set on by default. You can use: C, C++, C++11, C++14, JAVA, C#, Python, Python3, Scala, Go, Rust, JavaScript, Ruby, PHP, D, OCaml, Haskell, and Kotlin.
- ②-2 **Editor** : The text area for writing code. You can copy the code from your tool and paste it to the area. The text area provides a simple editor. So you can also directly write code (but not recommended).
- ②-3 **Submit Button** : Send the code to the judge.

### 3-4. Checking Verdicts

After the code is submitted, the status view (③) and the test case view (④) are updated as follows.

JUDGE ID : 2697729 By YUTAKA

COMPILER ERROR

③-2

Submitted Sent to Judge Build Run Resource Limit Check Result Check Presentation Check

Lang	Status	Judge	Time	Memory	Code
C++	CE	0/16	00:00 s	0 KB	16 B

rep/code.cpp: In function 'int main()': rep/code.cpp:1:11: error: expected '}' at end of input int main(){ ^

TEST CASE

Case #	Status	Time	Memory	In	Out	caseName
> # 1	CE	00:00 s	0 KB	15 B	5 B	00_sample_01.in

The top-right corner of the status view (③-1) shows the progress of the judge and the final verdict. The progress is represented by WAITING JUDGE → JUDGING → verdict. In the above example, a code with a syntax error is submitted, so the verdict shows COMPILER ERROR.

## AOJ Tutorial

Check marks on the horizontal line in the status view (③-2) are auxiliary items to confirm the stage the judge stopped. The stage make transition from the left side to the right. If all stages are passed, your program is accepted, otherwise rejected. The following example shows the judge result of the program debugged by removing the syntax error.

JUDGE ID : 2693350 By YUTAKA WRONG ANSWER

Submitted Sent to Judge Build Run Resource Limit Check X Result Check Presentation Check

Lang	Status	Judge	Time	Memory	Code
C++	<span style="color: red;">WA</span>	2/16	00:00 s	3112 KB	205 B

③-3

④

TEST CASE

	Case #	Status	Time	Memory	In	Out	caseName
>	# 1	<span style="color: green;">AC</span>	00:00 s	3112 KB	15 B	5 B	00_sample_01.in
>	# 2	<span style="color: green;">AC</span>	00:00 s	3088 KB	15 B	2 B	00_sample_02.in
∨	# 3	<span style="color: red;">WA</span>	00:00 s	3016 KB	14 B	3 B	00_sample_03.in

input: 500 1000 2000

output: NA

The judge shows that the program is wrong. The stages show that the program passed Build, Run, and Resource Limit Check. If the program is compilable and executed without runtime errors, you can check CPU time and memory usage at the bottom of the status view (③-3).

## AOJ Tutorial

The system judges your program according to the following procedure:

Stage	Progress / Verdicts	Details
Submitted	WAITING JUDGE	The server receives your program, and it is added to the judge queue.
Send to Judge	JUDGING	The program is removed from the queue and sent to the judge. You are waiting for the verdict.
Build	CE: COMPILE ERROR	Compiler the judge uses has failed to compile your program. Please see error messages from the judge. If you need, please see the version of compilers at System Information.
Run	RE: RUNTIME ERROR	Your program has failed during the execution. Possible causes include: out of range in pointer reference (Segmentation Fault), stack overflow (infinite recursive functions), division by zero and many others. Note that, the main function in C language should return 0.
Resource Limit Check	TLE: TIME LIMIT EXCEEDED	The CPU time your program has run has exceeded the time limit specified by the corresponding problem. Note that, the judge runs your program up to around the Time Limit +1 seconds, and the judge may be forcibly terminated around 40 seconds with Runtime Error.
	MLE: MEMORY LIMIT EXCEEDED	The maximum amount of memory that your program has used has exceeded the limit specified for the corresponding problem.
	OLE: OUTPUT LIMIT EXCEEDED	Your program has produced too much output.
Result Check	WA: WRONG ANSWER	Your program has produced the output which is different from the judge data. In the case of special judges, the judge may return Wrong Answer depending on results of validators prepared for the corresponding problem.
Presentation Check	PE: PRESENTATION ERROR	Your program is almost accepted but not perfect. Your program outputs extra spaces or blank lines, and/or does not output required spaces or blank lines.
	AC: ACCEPTED	Your program has been "accepted" since it has not failed into all the above mentioned conditions.

The test case view ④ indicates that your program is judged by using a number of test cases (judge data files). In the test case view, you can check the verdict and resources your program used for each judge data. In AOJ, when the judge rejects your program, further the system will not go any further judge.

The above example shows that the program passed the first two cases, but failed at the third case with WRONG ANSWER status. In the test case view, you can check data which used for the judge by clicking the icon of each test case. In the example, the contents of the third case are disclosed. The judge input is shown in the left side, and the judge output (correct answer) is shown in the right side. These data can be used for debugging the program.

In the online judge system, you can re-submit your program any number of times. So you can try until you are satisfied in terms of both correctness and efficiency.

The following figure shows an example of the status view after submitting a correct program. It shows ACCEPTED status.

The screenshot displays the submission status for a program. At the top, it shows 'JUDGE ID : 2693351 By YUTAKA' and a green 'ACCEPTED' label. Below this is a progress bar with seven steps: Submitted, Sent to Judge, Build, Run, Resource Limit Check, Result Check, and Presentation Check, all marked with checkmarks. A summary table shows the submission details, and a 'TEST CASE' table lists five cases, all with 'AC' status.

Lang	Status	Judge	Time	Memory	Code
C++	AC	16/16	00:00 s	3092 KB	205 B

Case #	Status	Time	Memory	In	Out	caseName
> # 1	AC	00:00 s	2984 KB	15 B	5 B	00_sample_01.in
> # 2	AC	00:00 s	3032 KB	15 B	2 B	00_sample_02.in
> # 3	AC	00:00 s	3000 KB	14 B	3 B	00_sample_03.in
> # 4	AC	00:00 s	3092 KB	12 B	4 B	10_simple_01.in
> # 5	AC	00:00 s	2988 KB	16 B	5 B	10_simple_02.in