

2024年12月2日



AOJ 3.0 チュートリアル

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

UNIVERSITY OF AIZU

目次

はじめに	2
アカウント登録・ログイン	3
基本操作	4
設定	5
問題を選択する	6
コードを書く	9
テストする	10
AOJ へ提出する	13
レビュー機能	15
解説を閲覧する	19
過去の提出履歴を閲覧する	21
模範解答を閲覧する	22
ブックマーク	24
前の問題・次の問題	25
問題を検索	26
ダッシュボード	27
他ユーザーのダッシュボードを表示	30
フリースペース	31
ショートカットキー	32
お問い合わせ	32

はじめに

このチュートリアルでは、AOJ 3.0 の使い方を解説します。今後のリリースへ向けてユーザーインターフェースや機能が変更される可能性がありますので、あらかじめご了承ください。

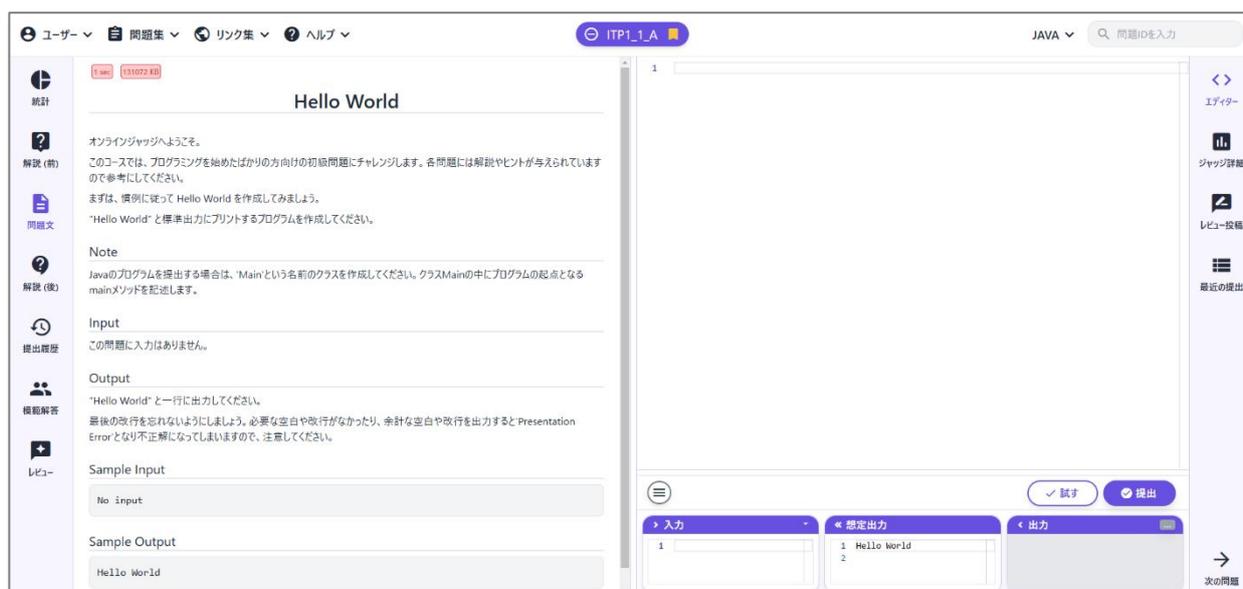
AOJ 3.0 はオンラインコーディングエディタとして実装されています。AOJ 3.0 を通して、以下の作業を画面ひとつで行うことができます。

- 問題の検索
- 問題文の閲覧
- コーディング
- サンプル入出力や指定された入出力を用いた動作確認
- プログラムの提出と判定結果確認
- 学習の進捗を確認
- その他

アカウント登録・ログイン

以下の URL から AOJ 3.0 にアクセスしてみましょう。AOJ 3.0 の全ての機能を使用するには AOJ のアカウントでシステムにログインする必要があります。

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



サイトにアクセスした後、

1. 左上のメニューから「ユーザー」→「ログイン」を選択してください。
2. ポップアップが表示されるので、
 - ① アカウントをお持ちでない方は「アカウント新規作成」から AOJ のアカウントを作成してください。
 - ② アカウントをお持ちの方は、ご自分のユーザー ID とパスワードを入力し、ログインしてください。



基本操作

画面中央にあるバーをクリックしながら左右に移動させることで、パネルの幅を変更できます。またテキストエディタ下部のバーをクリックしながら上下に移動させると、エディタの高さを変更できます。

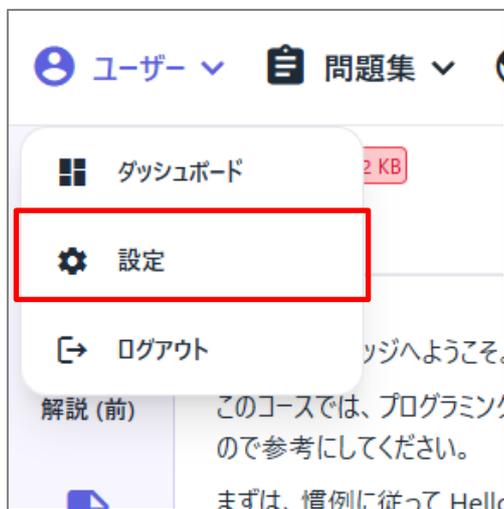


エディタ下部のメニューから拡大・縮小ボタンをクリックすると、エディタのフォントサイズを変更できます。

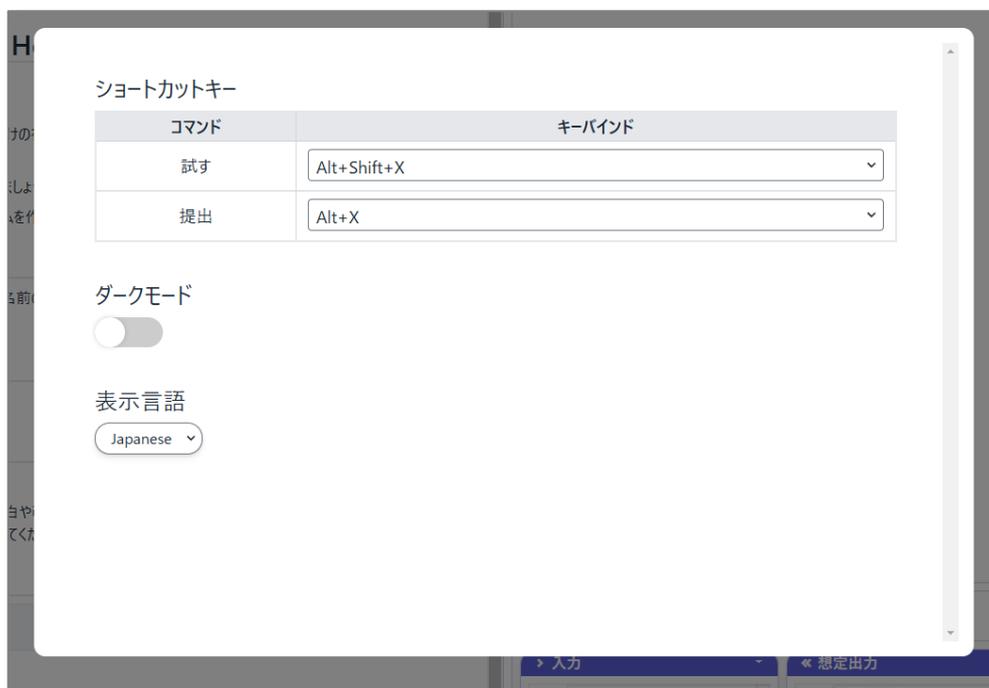


設定

「ユーザー」→「設定」からサイトの設定を変更できます。



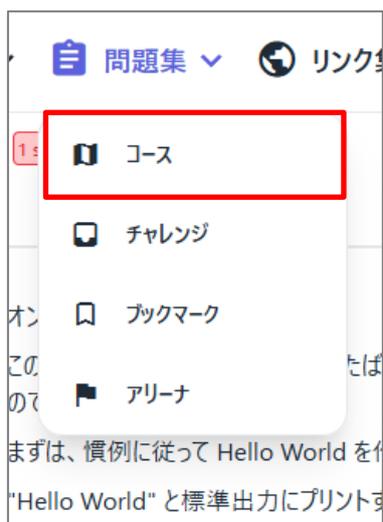
設定画面ではショートカットキーの変更とダークモードへの切り替えが可能です。



問題を選択する

問題を検索して、問題文を開いてみましょう。

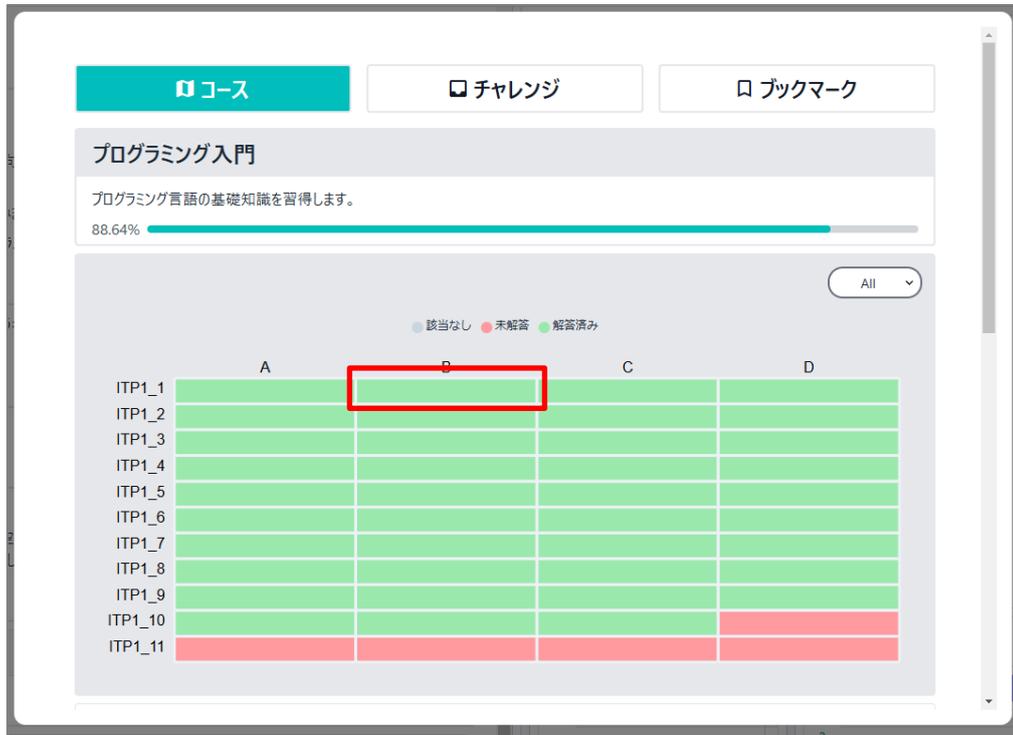
1. 「問題集」→「コース」・「チャレンジ」・「ブックマーク」・「アリーナ」いずれかをクリックすると、問題検索パネルが開きます。



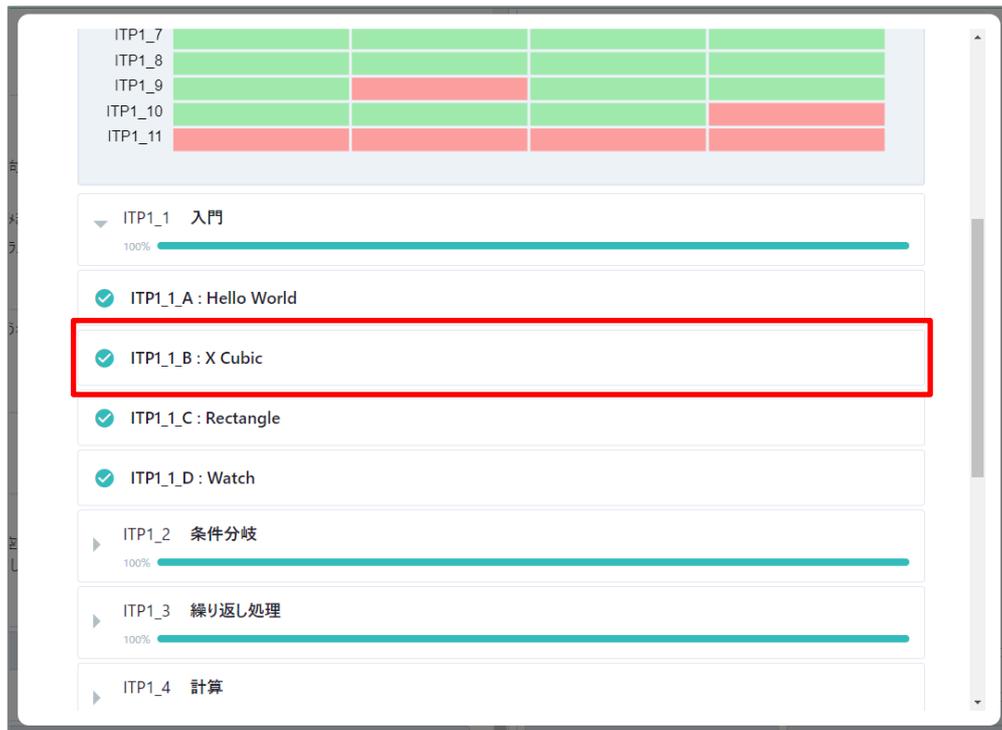
2. 「コース」を選んでみましょう。各コースはカードで表されています。各カードに表示されているプログレスバーは、そのコースの進捗状況を示します。



3. 続いて「プログラミング入門」を開いてみましょう。コースは複数のトピックから構成されています。ヒートマップや各トピックのプログレスバーから進捗を確認できます。



4. ヒートマップ上のセル「ITP1_1_B」または、「ITP1_1 入門」→「X Cubic」を開いてみましょう。



5. 左側のパネルに問題文が表示されます。右側のパネルにはテキストエディタが表示され、下部のコンソールには選択した問題のサンプルケースが自動的に入力されます。

The screenshot shows the AOJ 3.0 problem-solving interface for problem ITP1_1_8. The interface is divided into several panels:

- Left Panel:** Contains navigation and problem details. The title is "x の 3 乗". The description states: "1 つの整数 x を読み込んで、 x の 3 乗を計算し結果を出力するプログラムを作成して下さい。" The input section says "入力として 1 つの整数 x が与えられます。" The output section says "x の 3 乗の計算結果を 1 行に出力して下さい。最後の改行を忘れないようにしましょう。" The constraints section says "Constraints (制約)には、採点に使用される入力データの上界などの情報が書かれています。変数の型を選ぶときや、アルゴリズムを考えるときに必要な情報になります。この問題に関する制約は以下の通りとなります："
 - $1 \leq x \leq 100$Sample Input 1: 2, Sample Output 1: 8, Sample Input 2: 3, Sample Output 2: 27.
- Top Panel:** Shows the problem ID "ITP1_1_8" and the programming language "JAVA".
- Right Panel:** Contains a text editor for writing code and a sidebar with utility icons like "エディター", "ジャッジ詳細", "レビュー投稿", and "最近の提出".
- Bottom Panel:** Contains a console area with three tabs: "入力" (Input), "想定出力" (Expected Output), and "出力" (Output). The "入力" tab shows the sample input "2", the "想定出力" tab shows the sample output "8", and the "出力" tab is currently empty.

コードを書く

テキストエディタでコードを書いてみましょう。



赤い枠で示した部分にプログラムのコードを書きます。または、エディタ下部のメニューからアップロードボタンでエクスプローラーを開き、ローカルファイルシステムからソースコードを読み込むこともできます。加えて、ダウンロードボタンから、エディタ上のソースコードをローカルファイルシステムへ保存することができます。

テストする

コーディングが終わったら、テスト機能でコードが正しく動作するか確認しましょう。テストはサンプル入出力を用いた動作確認を行うための、簡易的なチェック機能です。

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 class Main {
6
7     public static void main(String args[]) throws IOException {
8         BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
9         int x = Integer.valueOf(in.readLine());
10        x = x * x * x;
11        System.out.println(x);
12    }
13 }
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

The screenshot shows the AOJ 3.0 IDE interface. The code editor contains the following Java code:

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 class Main {
6
7     public static void main(String args[]) throws IOException {
8         BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
9         int x = Integer.valueOf(in.readLine());
10        x = x * x * x;
11        System.out.println(x);
12    }
13 }
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

The test interface at the bottom includes:

- 入力 (Input):** A text area containing "1 2".
- 想定出力 (Expected Output):** A text area containing "1 8".
- 出力 (Output):** An empty text area.
- Buttons:** "試す (Try)" and "提出 (Submit)".

A red arrow points to the "試す" button.

1. 「入力」エリアと「想定出力」エリアにそれぞれ、プログラムが読み込む入力値、期待するプログラムの出力値を設定します（あらかじめサンプル入出力が設定されています）。
2. 「試す」をクリックすると、ここで設定された入力値・出力値とともにプログラムが送信されます。結果が返ってくるまでは、時間がかかる場合があります。
3. コンパイルと実行が成功すれば、「出力」エリアに、プログラムの出力が表示されます。コンパイルや実行に失敗すると、エラーメッセージが表示されます。

エディター下部のメニューからフリースペースモードを選択すると「想定出力」エリアが非表示になり、「出力」エリアに、プログラムの出力のみ表示されます。ジャッジステータスは表示されません。回答プログラムの出力のみ確認したい場合は、フリースペースモードをご利用ください。



ジャッジは、以下のいずれかのステータスを返します。

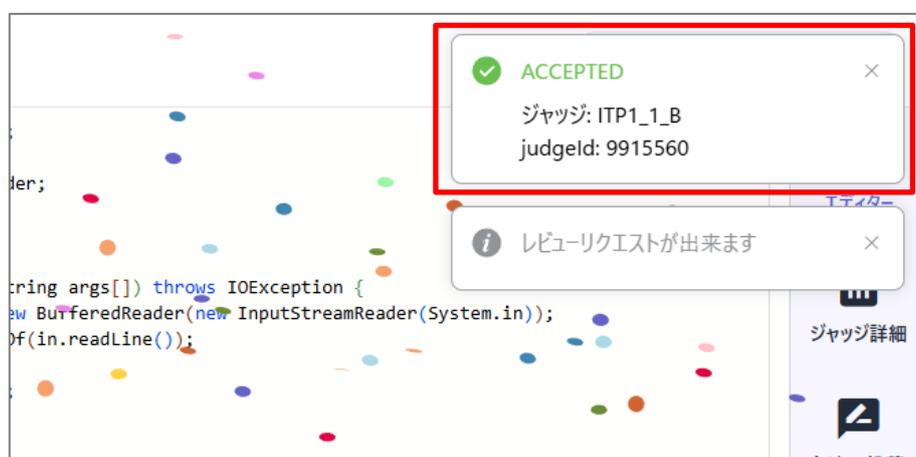
AC 正解 (Accepted)
WA 不正解 (Wrong Answer) ・ ジャッジデータが想定した出力とは異なる出力が行われた
PE 表示形式に誤り (Presentation Error) ・ 余計な空白や改行がある、あるいは必要な空白や改行がない
CE コンパイルエラー (Compile Error) ・ 提出されたプログラムのコンパイルに失敗した
TLE 時間制限超過 (Time Limit Exceeded) ・ 問題で指定された制限時間内にプログラムが終了しなかった
MLE メモリ制限超過 (Memory Limit Exceeded) ・ 問題で指定された以上のメモリを使用した
RE 実行時エラー (Runtime Error) ・ 提出されたプログラムの実行中にエラーが発生した

AOJ へ提出する

テストで正しい結果が得られたら AOJ に提出しましょう。



「提出」ボタンからコードを提出します。提出されたプログラムは、ジャッジデータによって厳格に採点されます。提出結果はサイト上部に通知されます。



ステータスを詳しく確認したい場合は、メニューの「ジャッジの詳細」アイコンをクリックします。テストケースごとの詳しいジャッジの結果を確認することができます。

結果: AC

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

言語	時間	メモリー	コード長
JAVA	0.03 sec	39224 KB	346 B

ケース	ステータス	時間	メモリー	入力	出力	ケース名	貼り付け
# 1	AC	0.03 sec	39172 KB	2 B	2 B	test1	
# 2	AC	0.03 sec	38952 KB	2 B	3 B	test2	
# 3	AC	0.03 sec	39224 KB	3 B	7 B	test3	
# 4	AC	0.03 sec	37116 KB	4 B	8 B	test4	

入力: 3
出力: 27

ダウンロード

画面右端の「貼り付け」ボタンをクリックすると、エディタ下部の「入力」エリアと「想定出力」エリアにそれぞれ選択したテストケースの「入力」・「出力」が貼り付けられます。

試す 提出

入力

1 3
2

想定出力

1 27
2

出力 AC

1 8
2

次の問題

レビュー機能

エディタ下部のメニューや提出履歴画面に表示されているレビューボタンをクリックすると、レビューリクエスト画面が表示されます。

The screenshot displays the AOJ editor interface. At the top left, a vertical menu contains several icons, with the 'レビュー' (Review) button highlighted by a red box. Below the menu are buttons for '試す' (Try) and '提出' (Submit). The main area shows input, expected output, and actual output fields. Below these, a submission history table is visible, with the 'レビュー' button highlighted in red in the bottom row of the table. The code editor shows a Java program that reads an integer and prints its cube.

```
18 レビュー
21 フリースペース
23 アップロード
26 ダウンロード
+ 拡大
- 縮小
```

試す 提出

> 入力 << 想定出力 < 出力 AC

```
1 3
2

1 27
2

1 8
2
```

▼	8676085	RE	JAVA	2023/12/20 18:45:22
▲	8588761	AC	JAVA	2023/12/05 22:42:13

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 class Main {
6
7     public static void main(String args[]) throws IOException {
8         BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
9         int x = Integer.valueOf(in.readLine());
10        x = x * x * x;
11        System.out.println(x);
12    }
13 }
14
15
16
17
18
19
20
```

レビュー コピー

▼	8424051	AC	JAVA	2023/10/24 01:49:50
---	---------	----	------	---------------------

レビュー機能を使用することで、レビュアーからプログラムに含まれるバグや改善点に関するフィードバックを得ることができます。

一覧からリクエストしたい項目を選択すると、内容に応じたレビューを受け取ることができます。

📌 レビュー

より読みやすいプログラムに修正するためのヒントをください。

より効率的なプログラムに修正するためのヒントをください。

より読みやすいプログラムに修正してください。

より効率的なプログラムに修正してください。

方法が異なる別解を教えてください。

自由記述

エントリーされたリクエストに対して、AIがコードレビューを行う場合があります。詳しくは下記の利用規約をご覧ください。

- [ChatGPT](#)
- [Gemini](#)

▶ レビューをリクエスト

レビュー内容を確認する場合は、メニューの「レビュー」アイコンをクリックします。

「オープンエントリー」には、現在レビューを受け付けているレビューエントリーが表示されます。

「クローズエントリー」にはすでにレビュー受付を終了したレビューエントリーが表示されます。

The screenshot shows the AOJ 3.0 Tutorial interface. On the left is a vertical navigation menu with icons and labels: 統計 (Statistics), 解説 (前) (Explanation (Previous)), 問題文 (Problem Statement), 解説 (後) (Explanation (Next)), 提出履歴 (Submission History), 模範解答 (Model Answer), and **レビュー** (Review), which is highlighted with a red box. The main content area is divided into two sections: 'オープンエントリー' (Open Entries) and 'クローズエントリー' (Closed Entries). Each section contains a table with columns for ジャッジID (Judge ID), ステータス (Status), 言語 (Language), and 日時 (Date/Time).

オープンエントリー			
ジャッジID	ステータス	言語	日時
▼ 9077173	WA	JavaScript	2024/09/19 08:51:30

クローズエントリー			
ジャッジID	ステータス	言語	日時
▼ 9405073	AC	Python3	2024/08/29 23:04:23
▼ 9108111	RE	Python	2024/08/29 21:01:13
▼ 9480328	CE	C++20 (Beta)	2024/08/29 20:09:26
▼ 9480346	CE	C++23	2024/08/26 07:21:30
▼ 9480346	CE	C++23	2024/07/19 16:55:43
▼ 9052935	CE	C	2024/04/02 19:56:12
▼ 7243501	CE	OCaml	2024/03/16 02:08:21
▼ 7243501	CE	OCaml	2024/03/16 02:08:21
▼ 8869688	WA	Ruby	2024/02/05 21:12:39
▼ 8856115	CE	C#	2024/02/05 11:59:59

レビューエントリーから投稿されたレビューを選択すると、レビュー内容が表示されます。ユーザーは投稿されたレビューに対して5段階評価を行うことができます。「エントリーをクローズ」をクリックするとレビューエントリーはクローズされ、これ以上レビューを受け取らなくなります。



メニューの「レビュー投稿」アイコンをクリックすると、他のユーザーのレビューエントリーに対して、レビューを投稿することができます。



解説を閲覧する

解答プログラムの書き方が分からない場合や、模範解答を知りたい場合は解説を参考にしましょう。メニューの「解説（前）」と「解説（後）」から、解説を表示できます。

「解説（前）」では、主に問題文を読む前に役立つ、文法や方針の説明を閲覧することができます。

ユーザー ▾ 問題集 ▾ リンク集 ▾ ヘルプ ▾ ITP1

言語解説 JAVA ▾

変数

コンピュータ（プログラム）に入力された値は一時的にメモリに記録する必要があります。メモリの特定領域を確保して、指定した名前アクセスするための仕組みが**変数**です。

変数は、そのデータ型（整数、浮動小数点、文字、など）と名前の組で**宣言**する必要があります。たとえば、1つの整数データを保持することができる x という名前の変数は以下のように宣言します。

```
int x;
```

代入演算

変数の役割は、プログラムで扱う値をメモリへ書き込み、必要な時に参照することです。最も基本となる**代入演算**では、 $=$ の左側の変数（メモリ領域）に、右側の式の計算結果が書き込まれます。変数はその名の通り「変わる数」で、代入演算によって値を何度でも変更することができます。一方、式に含まれる変数は、それが指すメモリ領域の値を返します。たとえば、次のプログラムは、変数 a , b にそれぞれ 5 , 8 を格納し、その後それらの中身を交換するプログラムです。

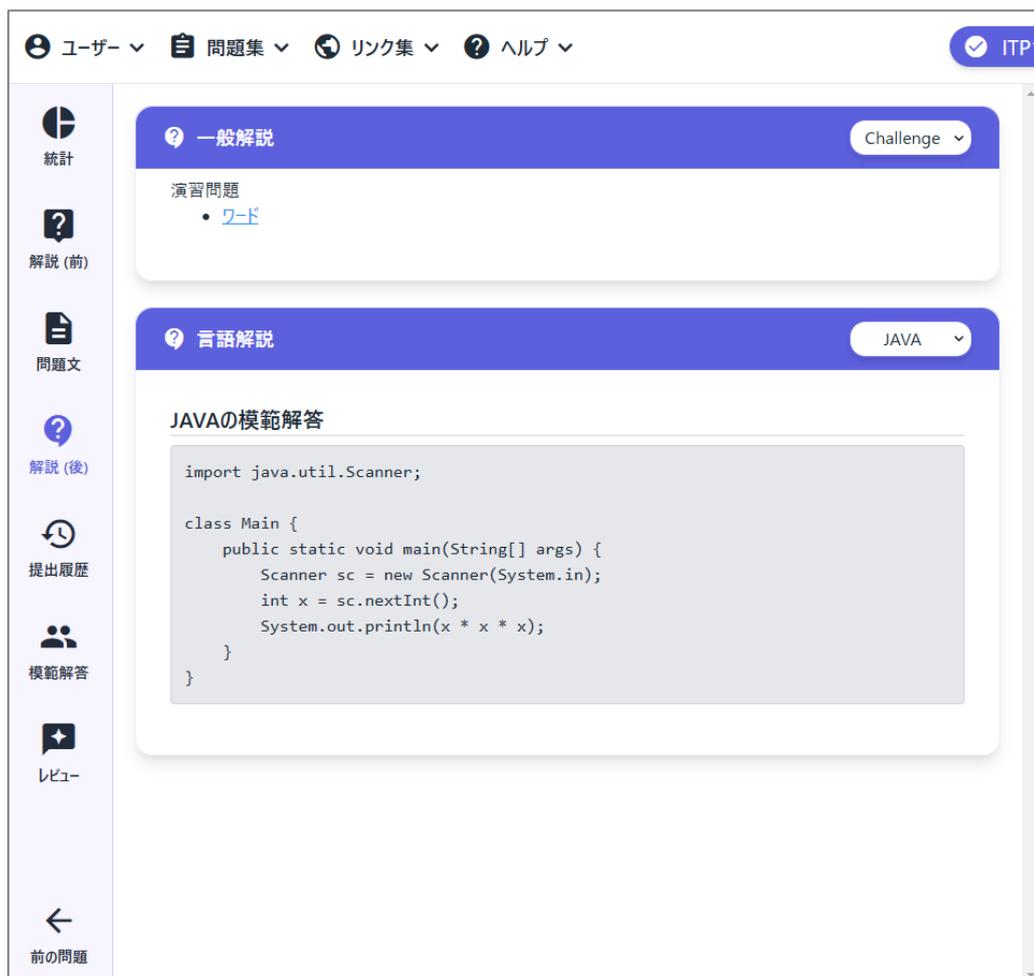
```
int a, b, t; // 3つの整数を宣言
a = 5;      // a に 5 を格納
b = 8;      // b に 8 を格納
t = a;      // t に a の値を格納
a = b;      // a に b の値を格納、この直後 a の値は 8 になる
b = t;      // b に t の値を格納、この直後 b の値は 5 になる
```

算術演算

プログラミングの目的はコンピュータに**計算**をさせることです。計算の最も基本となるものが**算術演算**です。定数、変数、関数などに対して算術演算を行い計算式を記述します。四則演算はそれぞれ、 $+$, $-$, $*$, $/$, で行います。たとえば

```
y = 2 * a - b / c;
```

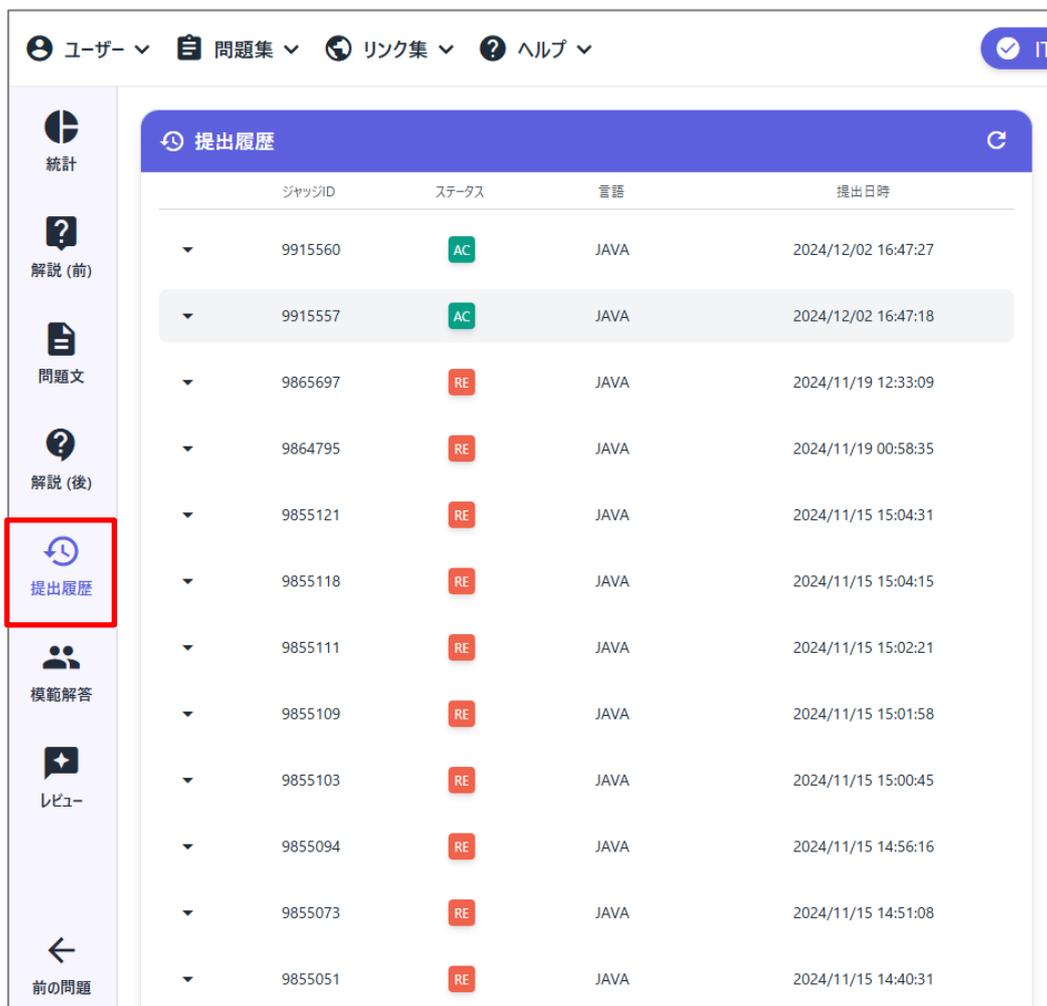
「解説（後）」では、解答後に役立つ、模範解答や詳しい説明を閲覧することができます。



解説には、「言語解説」や「一般解説」などの種類があります。言語解説では、指定したプログラミング言語の文法に関する解説を閲覧することができます。一般解説では、想定するアルゴリズムやヒントを閲覧することができます。（※解説がサポートされていない問題もあります。予めご了承ください）

過去の提出履歴を閲覧する

過去に提出したソースコードを確認できます。メニューの「提出履歴」アイコンをクリックすると、選択中の問題に対する過去の提出履歴を表示することができます。



ジャッジID	ステータス	言語	提出日時
▼ 9915560	AC	JAVA	2024/12/02 16:47:27
▼ 9915557	AC	JAVA	2024/12/02 16:47:18
▼ 9865697	RE	JAVA	2024/11/19 12:33:09
▼ 9864795	RE	JAVA	2024/11/19 00:58:35
▼ 9855121	RE	JAVA	2024/11/15 15:04:31
▼ 9855118	RE	JAVA	2024/11/15 15:04:15
▼ 9855111	RE	JAVA	2024/11/15 15:02:21
▼ 9855109	RE	JAVA	2024/11/15 15:01:58
▼ 9855103	RE	JAVA	2024/11/15 15:00:45
▼ 9855094	RE	JAVA	2024/11/15 14:56:16
▼ 9855073	RE	JAVA	2024/11/15 14:51:08
▼ 9855051	RE	JAVA	2024/11/15 14:40:31

リストから特定の提出履歴をクリックすると、そのコードを閲覧することができます。

模範解答を閲覧する

他のユーザーが解答したコードを閲覧することができます。メニューの「模範解答」アイコンをクリックすると、選択中の問題に対する模範解答を表示することができます。

The screenshot shows the AOJ 3.0 interface. At the top, there are navigation menus for 'ユーザー', '問題集', 'リンク集', and 'ヘルプ'. On the right, there is a 'ITP' button. The main content area is titled '模範解答 (JAVA)'. Below the title is a table of solutions. The left sidebar contains several menu items: '統計', '解説 (前)', '問題文', '解説 (後)', '提出履歴', '模範解答' (highlighted with a red box), and 'レビュー'. At the bottom of the sidebar is a '前の問題' button.

公開	ジャッジID	ユーザー	レーティング	Time	Code Size
▼	6390975	naoto172	4743.31	0.05 sec	246 B
▼	4913905	mohejin	1814.81	0.05 sec	245 B
▼	2064551	ei1333	1479.27	0.03 sec	336 B
▼	4663439	square1001	1369.96	0.08 sec	213 B
▼	9705872	vjudge3	1095.46	0.06 sec	198 B
▼	9911123	vjudge4	1094.14	0.05 sec	207 B
▼	9837245	vjudge2	1080.69	0.05 sec	244 B
▼	9745169	vjudge5	1070.84	0.05 sec	221 B
▼	9705432	vjudge1	1018.81	0.06 sec	199 B
▼	514140	kyuridenamida	820.89	0.09 sec	202 B
▼	2943892	emtsu_ba	641.53	0.06 sec	243 B
▼	2211379	face4	613.73	0.05 sec	377 B
▼	5924926	yutaka	599.13	0.05 sec	201 B

解答は各ユーザーのレーティングをもとに降順に表示されます。提出履歴と同様に、リストから模範解答をクリックすると、そのコードを閲覧することができます。「Diff」ボタンをクリックするとご自身のコードと模範解答のコードを比較することができます。

The screenshot shows the AOJ 3.0 interface for problem ITP1_1_B. A table lists solutions from various users, with the solution by 'yutaka' (ID: 5924926) highlighted in purple. Below the table, two code snippets are compared side-by-side. The left snippet (red background) uses `Scanner` for input, and the right snippet (green background) uses `BufferedReader` and `Integer`. A red box highlights the 'Diff' button, and a blue box highlights the 'Copy' button.

順位	ユーザーID	ユーザー名	レーティング	実行時間	メモリ使用量
1	514140	kyuridenamida	820.89	0.09 sec	202 B
2	2943892	emtsu_ba	641.53	0.06 sec	243 B
3	2211379	face4	613.73	0.05 sec	377 B
4	5924926	yutaka	599.13	0.05 sec	201 B
5	395778	id	594.50	0.08 sec	231 B
6	1727452	DAyama	536.74	0.06 sec	207 B

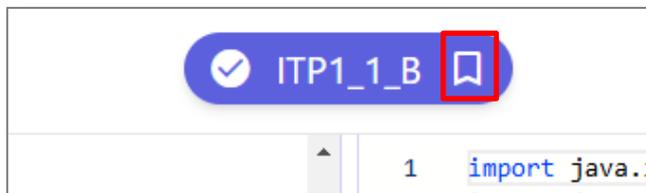
```

1- import java.util.Scanner;
2-
3- class Main{
4-     public static void main(String[] args){
5-         Scanner sc = new Scanner(System.in);
6-
7-         int x = sc.nextInt();
8-
9-         System.out.println(x*x*x);
10-
11-     }
12- }
13-
14-
15-
16-
17-
18-
1+ import java.io.BufferedReader;
2+ import java.io.IOException;
3+ import java.io.InputStreamReader;
4-
5+ class Main {
6-
7+     public static void main(String args[]) throws IOExcep
8+         BufferedReader in = new BufferedReader(new Inputs
9+         int x = Integer.valueOf(in.readLine());
10+         x = x * x * x;
11+         System.out.println(x);
12-     }
13- }
14-
15-
16-
17-
18-

```

ブックマーク

ヘッダーにある「ブックマーク」ボタンをクリックすることで、現在の問題を保存できます。



ブックマークした問題は検索パネルから確認できます。



前の問題・次の問題

サイドメニューにある「前の問題」「次の問題」ボタンをクリックすると、各コースの前の問題・次の問題が表示されます。

The screenshot shows the AOJ 3.0 interface for the problem "xの3乗" (x^3). The page is divided into several sections:

- Header:** Includes user information, problem set, link set, help, and the problem ID "ITP1_1_8".
- Left Sidebar:** Contains navigation buttons: "統計" (Statistics), "解説 (前)" (Previous Explanation), "問題文" (Problem Statement), "解説 (後)" (Next Explanation), "提出履歴" (Submission History), "模範解答" (Model Answer), and "レビュー" (Review). The "前の問題" (Previous Problem) button is highlighted with a red box.
- Main Content:**
 - Title:** xの3乗
 - Description:** 1つの整数 x を読み込んで、 x の3乗を計算し結果を出力するプログラムを作成して下さい。
 - Input:** 入力として1つの整数 x が与えられます。
 - Output:** x の3乗の計算結果を1行に出力して下さい。最後の改行を忘れないようにしましょう。
 - Constraints:** Constraints (制約)には、採点に使用される入力データの上乗などの情報が書かれています。変数の型を基にときや、アルゴリズムを考えるときに必要な情報になります。この問題に関する制約は以下の通りとなります:
 - $1 \leq x \leq 100$
 - Sample Input 1:** 2
 - Sample Output 1:** 8
 - Sample Input 2:** 3
 - Sample Output 2:** 27
- Right Sidebar:** Contains buttons: "エディター" (Editor), "ジャッジ詳細" (Judge Details), "レビュー投稿" (Review Submission), and "最近の提出" (Recent Submissions). The "次の問題" (Next Problem) button is highlighted with a red box.
- Bottom Panel:** Includes "入力" (Input), "想定出力" (Expected Output), and "出力" (Output) sections. The "入力" section shows the sample input values (2 and 3), and the "想定出力" section shows the corresponding output values (8 and 27). There are also "試す" (Try) and "提出" (Submit) buttons.

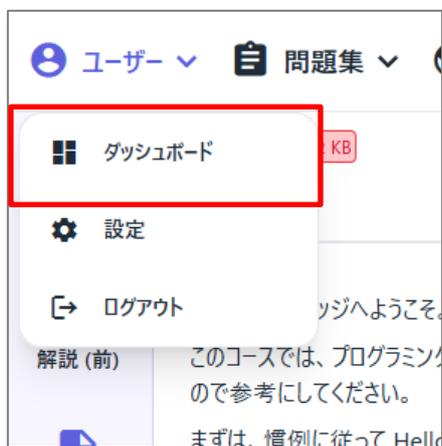
問題を検索

問題検索フォームに問題 ID を入力すると、その問題を開くことができます。問題 ID とはヘッダーに表示されている、各問題に割り当てられている ID を指します。



ダッシュボード

「ユーザー」→「ダッシュボード」をクリックすると、ユーザーの各ステータスがまとめられたダッシュボードを閲覧できます。



ダッシュボードには「ユーザー名」「提出数」「レーティング」などの基本的な情報が表示されます。加えて「直近 1 年の AC ヒートマップ」「毎週のレーティング」「言語別解答数」「問題別レーティング」「AOJ 各コースの進捗ヒートマップ」等、より複雑な情報を閲覧できます。

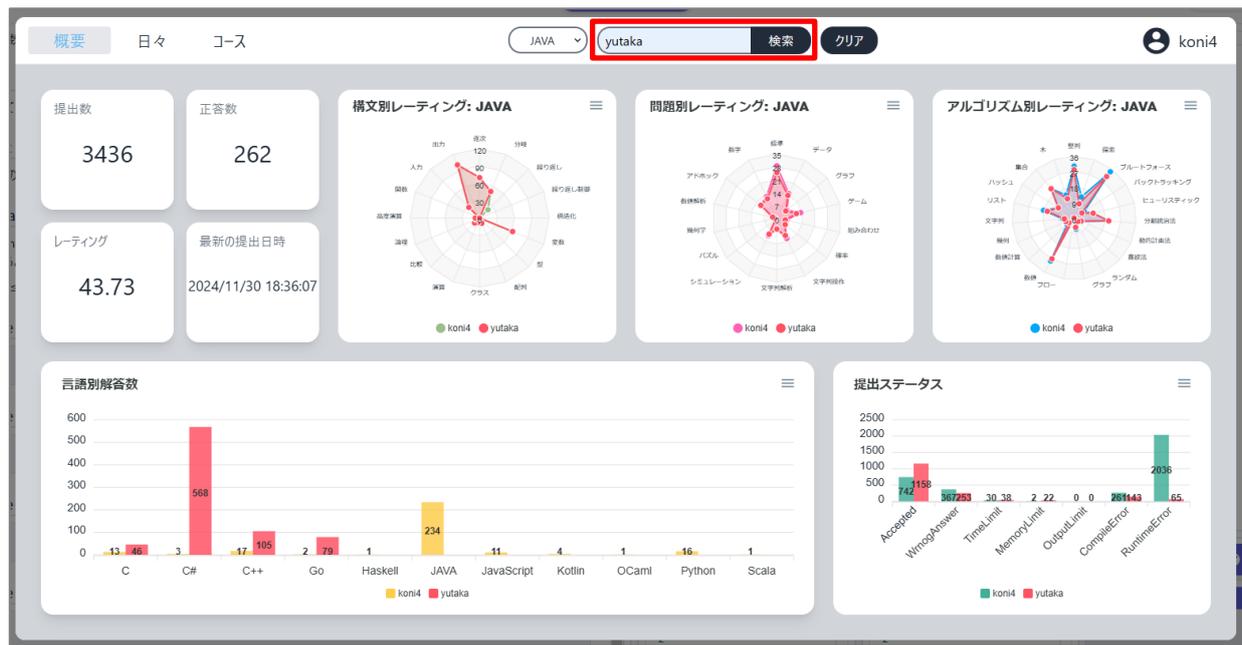




The figure shows the 'コース' (Courses) section for user 'koni4'. It lists various courses: ITP1, ALDS1, ITP2, DSL, DPL, GRL, CGL, NTL, and INFO1. Below this is a completion status table with columns A, B, C, and D, and rows ITP1_1 through ITP1_11. A legend indicates: '該当なし' (None), '未解答' (Not solved), and '解答済み' (Solved).

	A	B	C	D
ITP1_1	解答済み	解答済み	解答済み	解答済み
ITP1_2	解答済み	解答済み	解答済み	解答済み
ITP1_3	解答済み	解答済み	解答済み	解答済み
ITP1_4	解答済み	解答済み	解答済み	解答済み
ITP1_5	解答済み	解答済み	解答済み	解答済み
ITP1_6	解答済み	解答済み	解答済み	解答済み
ITP1_7	解答済み	解答済み	解答済み	解答済み
ITP1_8	解答済み	解答済み	解答済み	解答済み
ITP1_9	解答済み	解答済み	解答済み	解答済み
ITP1_10	解答済み	解答済み	解答済み	未解答
ITP1_11	未解答	未解答	未解答	未解答

検索フォームに他ユーザーの ID を入力すると、そのユーザーと比較したステータスが確認できます。



他ユーザーのダッシュボードを表示

他のユーザーのダッシュボードを表示するためには、WebブラウザのURLから、

「<https://onlinejudge.u-aizu.ac.jp/beta/ice?userId=ユーザーID>」のように指定することでダッシュボードを表示できます。例えばユーザーIDが aizutaro の場合、

「<https://onlinejudge.u-aizu.ac.jp/beta/ice?userId=aizutaro>」と指定します。

指定したURLにアクセスすると、他のユーザーのダッシュボードが表示されます。

加えて、ユーザーID aizutaro のコース ITP1 のヒートマップを表示したい場合、

「<https://onlinejudge.u-aizu.ac.jp/beta/ice?userId=aizutaro&courseName=ITP1>」と指定します。

指定したURLにアクセスすると、ユーザーID aizutaro の ITP1 に関するヒートマップが表示されます。

フリースペース

「リンク集」 → 「フリースペース」をクリックすると、フリースペース画面が表示されます。



この画面では問題に依存しない自由なプログラムのテストが可能です。

※プログラムデータサイズや入力データサイズに制限があります。

```
2 import java.io.IOException;

1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 class Main {
6     public static void main(String args[] throws IOException {
7         BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
8         int x = Integer.valueOf(in.readLine());
9         x = x * x * x;
10        System.out.println(x);
11    }
12 }
```

試す

入力
2

出力
8

ショートカットキー

AOJ 3.0 では以下のショートカットキーをサポートしています。

- ・ Alt+Shift+X: 試す (デフォルト)
- ・ Alt+X: 提出 (デフォルト)

お問い合わせ

本システムは現在開発中です。バグのご報告や、ご提案・ご要望は uoaojs@gmail.com または yutaka@u-aizu.ac.jp までご連絡ください。