

Algorithms (2020 Summer)

コードチャレンジにおける trackの利用方法

杉山 悠司, 鈴木 凌斗, 矢谷 浩司

trackの概要

trackとは

Givery社が提供しているプログラミング学習・試験プラットフォーム



できること

機能・問題/教材

導入事例

料金

資料一覧

メディア

イベント

資料請求

体験版デモ



即戦力若手エンジニア採用



技術力+ビジネス創造力を備えた人材の育成

エンジニアの採用と育成を支援する

プログラミング「学習・試験」プラットフォーム

trackとは

エンジニア採用のための試験として使われていることが多い。

今回，特別にGivery様のご協力を得て，この授業のためにシステムを利用させていただくことになりました！👷

trackとは

左側にコーディング問題、右側にコーディング環境が表示され、ブラウザ上でコーディングを行うことができる。

右下では書いたコードに対するテストが実行され結果を確認できる。

The screenshot shows a coding interface for a problem titled "Rectangle" (1-3). The problem description asks for a program to calculate the area and perimeter of a rectangle given side lengths a and b . The input is a and b separated by a space, and the output is the area and perimeter separated by a space. Constraints include integer inputs and $1 \leq a, b \leq 100$. A sample input of "3 5" is shown. The code editor on the right contains the following Python code:

```
1 import sys
2
3 if __name__ == "__main__":
4     a, b = map(int, input().split())
5     print(a * b, 2 * (a + b))
6
```

Below the code editor, a blue bar indicates "CLEAR" and "Enterキーを押して次に進もう". The test results section shows "Ready!" and two successful test cases (test1 and test2). The summary shows "# tests 2", "# pass 2", and "# fail 0". A "Preview mode" button is visible in the bottom right corner.

trackの特徴

オンラインエディタ上でコードを書くことができるので手元での環境構築が一切いらない。

幅広いプログラミング言語に対応（今回の講義ではPython3に限定していますが）。

その場でテストケースを実行することが出来、自分のコードに対するフィードバックがすぐに得られる。

本講義でのtrackの利用方法

本講義でのtrackの利用方法

講義があるたびに演習問題を計2～3題ずつ配信予定。

- 基本課題
- Extra課題

毎回の講義パート終了後、Track上での問題ページへのリンクを、今日の時点で登録してもらったメールアドレス宛に送信。

track上では、「受験」「試験」などという単語が出てきますが、試験ではないので安心してください。

課題配信メール

noreply@tracks.runからメールが配信されます。スパムフォルダ等に入ってしまう可能性もあるので、気をつけてください。

デモ配信 Inbox ×

東京大学 矢谷研究室 <noreply@tracks.run>

🗨 Japanese ▾ > Cebuano ▾ [Translate message](#)

さま

こんにちは。これはデモの配信メールです。

【試験情報】

試験名 : デモ用

提出期限 : 2020-03-19 00:00 (GMT+09:00) Asia/Tokyo

【受験手順】

以下のURLから受験を開始してください。

[ここに課題提出のためのURLが記載されます。](#)

※モバイルからの受験はできません。次のPCブラウザから受験してください。
(Chrome, Firefox, Safari または Microsoft Edgeの最新版)

課題を行う手順

メールで配信されたリンクをクリックすると課題ページに飛びます。
画面一番下にある，氏名と学生証番号のフィールドに記入し，利用規約などに同意してページの一番下の「**試験を開始する**」をクリックすることで課題が見られるようになります。

受験環境を確認する

受験環境に問題はありません。

[詳細を確認する](#)

エントリーフォーム

氏名

矢谷 浩司

学生証番号

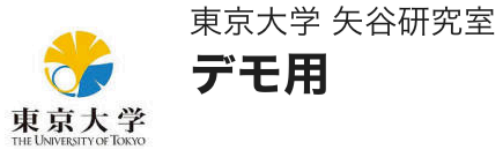
1234567

利用規約と個人情報および回答データの取り扱いに同意する

試験を開始する

課題を行う手順

試験を開始すると下のような課題一覧ページに飛びます。



今回解くべき課題一覧が表示されています
「解答する」を押すと問題ページに移動します



進捗状況
0% **0** / 1

提出期限: **2020/03/19 00:00**
(Asia/Tokyo)

試験を提出する

時差 🕒 無制限

初級 アルゴリズム

Python3

解答する

課題を行う手順 - 問題一覧ページ

Pythonコードを書くエディタ

The screenshot shows a web interface for a programming problem. The page title is "Algorithms 課題2" and the problem name is "時差". The interface includes a problem description on the left, a code editor in the center, and a console on the right. A red arrow points to the "時差" title, and another red arrow points to the "Pythonコードを書くエディタ" text. A third red arrow points to the "問題文および入力データのサンプル" text. A fourth red arrow points to the "ジャッジ結果が表示されるコンソール" text. The code editor contains Python code for a program that prints the line number and value for each line of input. The console shows the output "Hello track!".

時差

メモリ上限(MB): 512, 実行タイムアウト(ms): 2000

track氏は、イギリス出張のスケジュールを作成しました。そのスケジュールには、各予定の時刻が、現地時間で書いてありますが、会社に提出する書類には、日本時間で記入する必要があります。また、時刻のフォーマットも修正する必要があります。

予定の数が多く、手作業で修正するのは時間がかかりますが、track氏はプログラムが書けません。そこで、あなたが代わりに変換プログラムを作成することになりました。以下の仕様を満たすプログラムを作成してください。

- 予定の数 N と、それらの時刻 T_i ($1 \leq i \leq N$) が与えられる。
- T_i は 3 桁または 4 桁の整数で、下 2 桁が予定時刻の分を表し、上 1 桁または 2 桁が時間を表す。 T_i の先頭の数字は 0 でない。
- 日本とイギリスの時差は 9 時間である。つまり、出力すべき時刻は、 T_i が表す時刻の 9 時間後である。
- 変換後の時刻を h_i 時 m_i 分として、下の「出カルール」の項で指定されている通りに出力する。

実装の詳細

```
1 import sys
2
3 def main(lines):
4     # このコードは標準入力と標準出力を用いたサンプルコードです。
5     # このコードは好きなように編集・削除してもらって構いません。
6     # ---
7     # This is a sample code to use stdin and stdout.
8     # Edit and remove this code as you like.
9
10    for i, v in enumerate(lines):
11        print("line[{}]: {}".format(i, v))
12
13 if __name__ == '__main__':
14     lines = []
15     for l in sys.stdin:
16         lines.append(l.rstrip('\r\n'))
17     main(lines)
18
```

Ready!

command>

問題文および入力データのサンプル

ジャッジ結果が表示されるコンソール

課題を行う手順 – 問題一覧ページ

気をつけてほしいこと

与えられた**基本課題**においては、入出力や問題文で指示がある場合を除き、実装すべき処理において標準ライブラリや問題で指定されていない外部の関数やライブラリ等を使用しないでください。 そのような提出物はテストケースに合格していても、採点されませんで注意してください。

Extra課題に関しては、必要に応じて標準ライブラリや外部の関数などを使っても構いません。ただし、必要なものをすべて提出物の中に組み入れるようにしてください。

課題を行う手順－問題一覧ページ

気をつけてほしいこと

問題文をよく読んでから課題に取り組んでください。

課題によってはコードが一部予め与えられているものがあります。

(track上でデフォルトで与えられるコードと違う場合があります。)
この場合、問題文内で指示がありますので、それに従ってください。
従っていない提出物は採点されないことがあります。

課題を行う手順 – 問題一覧ページ

気をつけてほしいこと

問題ページ右上の「解答を提出する」を押して一度提出してしまうと
再編集ができなくなります。

作業途中の時は右下の「テストを実行」を押すと、その時点でのコードが自動的に保存されるので、それを活用してください。



課題を行う手順 – コーディング環境について

本講義ではPython3 (Python 3.6.1)に限定しています。

基本的に標準ライブラリしか含まれていません。状況によって、numpyなど一部のライブラリは追加するかもしれません。

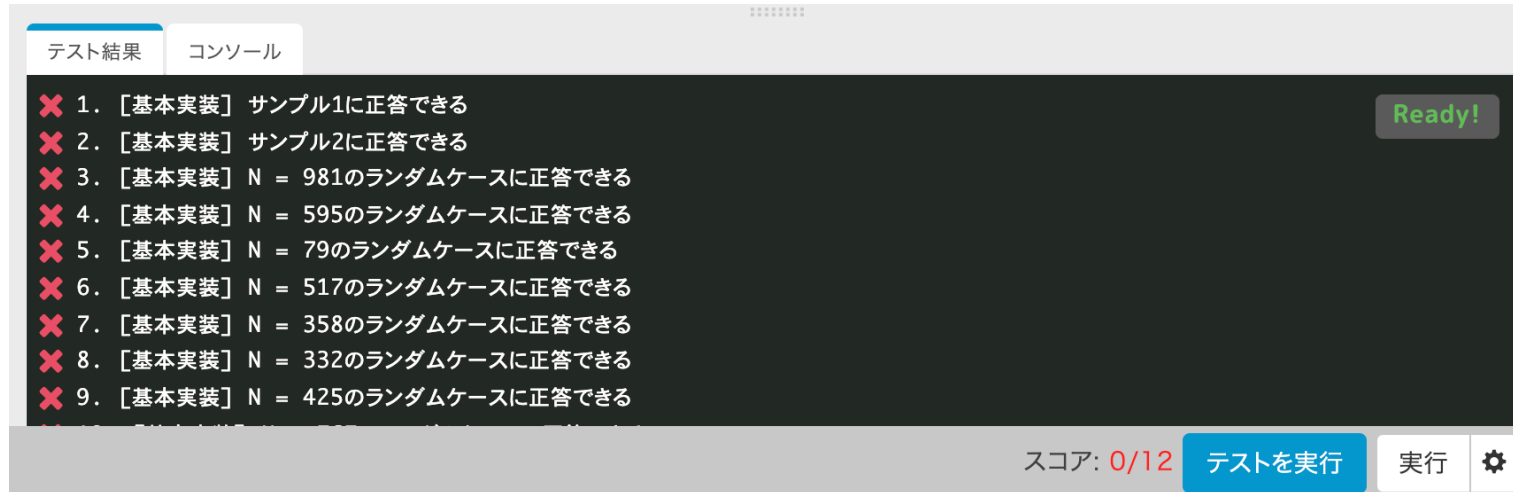
実装・デバッグをできる限りtrack上で行ってください。剽窃等の疑いが生じたときにtrack上でコードの更新履歴を確認することがあります。

課題を行う手順 - ジャッジについて

画面の右下の「**テストを実行**」(または⌘ + Sなど)を押すと用意されたテストケースが実行されます。

右下の「実行」ボタンはこちらでテストケースを自由に設定してコードを実行することができます。

「**テスト結果**」のタブにはそれぞれのテストケースの合否が、「**コンソール**」のタブには具体的な出力・エラーなどが表示されます。



The screenshot shows a coding platform interface with two tabs: "テスト結果" (Test Results) and "コンソール" (Console). The "テスト結果" tab is active, displaying a list of 9 test cases, each marked with a red 'X' and the text "[基本実装] サンプル1に正答できる" through "[基本実装] N = 425のランダムケースに正答できる". A green "Ready!" button is visible in the top right corner of the test results area. At the bottom of the interface, there is a score display "スコア: 0/12", a blue "テストを実行" (Run Tests) button, a white "実行" (Run) button, and a settings gear icon.

課題を行う手順 - テストケースについて

各問題について15個程度テストケースが用意されており，通ったテストケースの数に比例して採点が行われます。

テストケースは全て皆さんも確認できるようになっています。

テスト結果 コンソール

- ✓ 1. [基本実装] サンプル1に正答できる
- ✓ 2. [基本実装] サンプル2に正答できる
- ✓ 3. [基本実装] N = 981のランダムケースに正答できる
- ✓ 4. [基本実装] N = 595のランダムケースに正答できる
- ✓ 5. [基本実装] N = 79のランダムケースに正答できる
- ✓ 6. [基本実装] N = 517のランダムケースに正答できる
- ✓ 7. [基本実装] N = 358のランダムケースに正答できる
- ✓ 8. [基本実装] N = 332のランダムケースに正答できる
- ✓ 9. [基本実装] N = 425のランダムケースに正答できる

Ready!

スコア: 12/12 テストを実行 実行 ⚙️

課題を行う手順 – 入出力について

入力は標準入力によって与えられます。input関数などを用いると良いと思います。入力形式は各問題文を参照してください。

出力はprint関数などを用いて標準出力に出力してください。

課題を行う手順 – メモリ制限・実行時間

ほぼ全ての問題で

メモリ上限: **512MB**

実行タイムアウト: **5000ms**

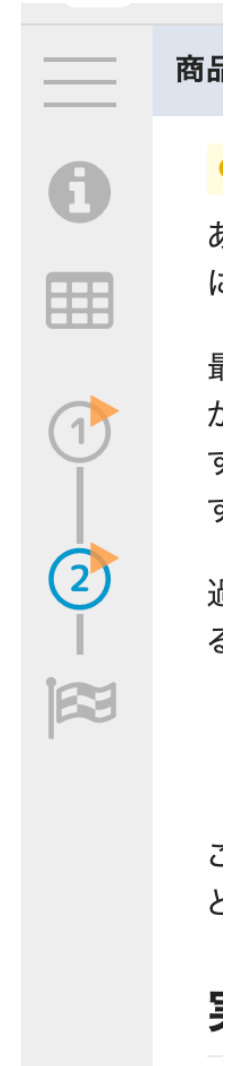
に設定する予定です。

あまりに実行時間がシビアな問題は一部を除いてありませんが、Pythonには処理が遅い書き方などがあるので注意してください。

課題を行う手順 – 問題の切り替え

左のメニューから課題を切り替えることができます。

別の問題を解く際は一度テストを実行することで自動セーブさせておくと安心です。



課題を行う手順 – 解答の提出

それぞれの問題でもうこれ以上変更を加えなくて良い状態になったら、右上の「**解答を提出する**」をクリックしてください。

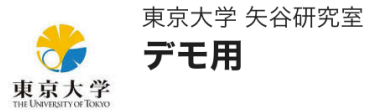
一度提出を完了するとコードの変更はできません。 再提出のリクエストは認めませんので、注意してください。

遷移後、「**ソースコードの説明をする**」画面が出てきます。今回は特に活用する予定はないので書かなくても結構です（何か伝えたいことがある際は書いてもよいです）。

課題を行う手順 – 課題全体の提出

初めの課題一覧ページに戻ると「試験を提出する」ボタンがあるので、全問解き終えたらこのボタンを押してください。

これを行わずに提出期限が来た場合は、最後に保存された状態のものが自動的に提出されます。



進捗状況
0% 0 / 1

提出期限: 2020/03/19 00:00
(Asia/Tokyo)

試験を提出する

時差

初級 アルゴリズム

Python3

無制限

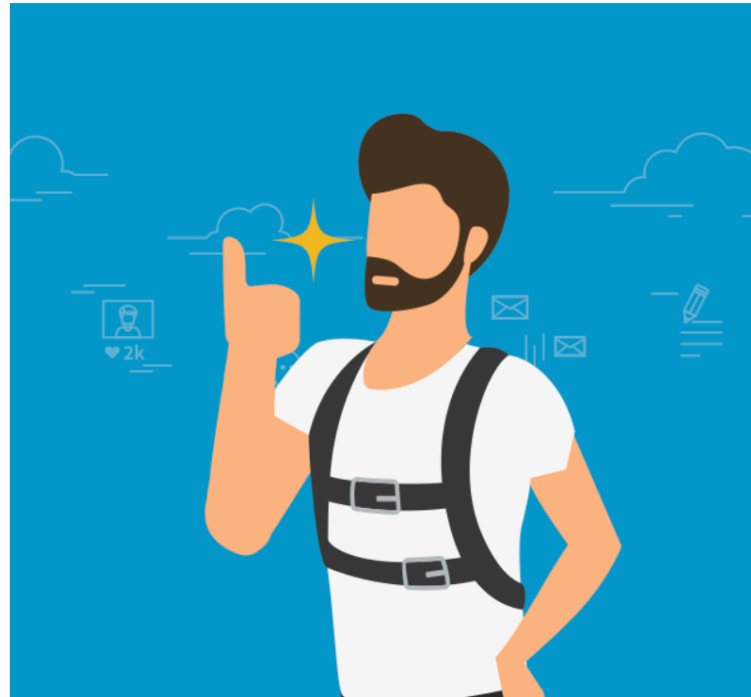
解答する

課題を行う手順 – 課題全体の提出

このイラストが出てくれば課題の提出は完了です。お疲れ様でした！

お疲れ様でした。これで試験は終了です。

もし解答中に深刻な問題が発生した場合は [こちら](#) までお知らせください。



課題提出にあたっての注意点

学生さん同士で議論することは推奨します。ただし、コードを直接共有する、などを行わないようにしてください。

参考書、Webサイト等を適宜参考にしながら、課題に取り組んでもらっても構いません。

TAさんはtrack上のトラブル等には対応しますが、個々のコードのデバッグには手助けできませんので、ご承知おきください。

課題提出にあたっての注意点

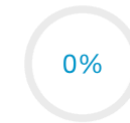
個々の課題（基本課題，Extra課題）自体には制限時間は設定されていませんが，**全体としての提出期限が翌日0:00に設定されています**ので，勘違いしないように注意してください。



東京大学 矢谷研究室

デモ用

東京大学
THE UNIVERSITY OF TOKYO



進捗状況

0 / 1

提出期限: **2020/03/19 00:00**

(Asia/Tokyo)

試験を提出する

時差

🕒 無制限

初級 アルゴリズム

Python3

解答する

課題提出にあたっての注意点

可能な限りtrack上で作業をしてください。

頻繁にテストケースを実行し、デバッグを行ってください。これにより提出間際でのミスを防ぐことができる他、コードが自動保存されるので、万が一提出作業を行えなくても、最終保存されたコードを元に採点が行われます。

採点は最終提出物に対してのみ行われます。したがって、途中でどれだけミスをしていてもペナルティはありません。

課題提出にあたっての注意点

課題によってはコードの一部（コーディング画面の初期コードとは別のコード）があらかじめ与えられている場合があります。 その場合、問題文に明記してありますので、指示に従ってください。

こちらが事前にインストールや用意しているもの以外のライブラリ、外部の関数等の利用はできません。そのような提出物はテストケースに合格していても、採点されませんので注意してください。

課題提出にあたっての注意点

提出物に対しては後日類似度チェックを行います。以下のような提出物は該当するものすべてに対して採点を取り消します。

- 提出物間でほぼ同一のコード
- Webや参考書に記載されているものとほぼ同一であることが判明したコード
- Track上での課題取り組み時間（終了時刻 - 開始時刻）が極端に短い
- 与えられたテストケースに通るようにだけ設計されたコード
- その他、明らかに不正行為の証拠があるもの

悪質な場合にはより大きなペナルティになることがあります。十分気をつけてください。