

プログラミング演習A

(2018年, 3セメ)

月曜日: 佐藤 昭, 伊藤 康一, ヘンリアンディプタラマ
金曜日: 鈴木 潤, 井之上 直也, 越田 俊介, 吉留 崇

この演習で目指すこと

- ・プログラミングの概念や技法を習得する
- ・アルゴリズムを設計・実装する能力を養う
- ・計算機アーキテクチャの理解を深める
- ・UNIX環境の利用方法を習得する
- ・プログラミングを通して、モノを作り上げる感動（楽しさや難しさ）を味わって欲しい



アカウント登録して問題をチェック！



メールアドレスで登録



Facebookアカウントで登録



Twitterアカウントで登録



GitHubアカウントで登録

CodeIQ NEWS

2016年3月25日 [【新着問題】「【Web初級】Web開発の基礎知識はあるよね！」](#)

2016年3月24日 [【新着問題】「社畜ちゃんからの挑戦状？～『いきのこれ！ 社畜ちゃん』コラボ問題～【選択式】」](#)

[一覧へ >](#)

これまでに**99,021**人が挑戦済み！ ただいま**65**問受付中！ => **27**人が現在解答中

リクルートコミュニケーションズ: 過去の出題と回答例（中途採用）

<https://www.rco.recruit.co.jp/career/engineer/entry/pastexam>

— 例題:1 「相手の思考を推理する」過程をプログラムで表現してください。

※こちらは回答例がございません。予めご了承ください。

問題内容

A,B,Cの3人が1~5の5枚のカードを使ってインディアンポーカーをします。

3人は、ランダムに1枚のカードを引いて額にかざします。相手のカードは見えますが、自分のカードは見えません。

この状態で、A->B->Cの順番に、自分が1番大きい(MAX)、自分は2番目に大きい(MID)自分が1番小さい(MIN)、わからない(?)、を答えます。

一人でも答えがわかった場合、そこで終了となります。「わからない」と答えた場合、回答権が次の人に移ります。

Cもわからない場合、再度Aに回答権が移ります。3人ともウソを言ったり、適当に答えてはいけません。

例1) 「A=1 B=4 C=5」だった場合、「A => MIN」で終了します。

例2) 「A=1 B=2 C=4」だった場合、「A => ?, B => MID」で終了します。

Bは「Aがわからないなら、自分は5ではない」と考えるからです。

以上を踏まえて、

引数で「A=1 B=4 C=5」で実行すると「A =>MIN」を出力

引数で「A=1 B=2 C=4」で実行すると「A =>?, B =>MID」

を出力するようなコマンドラインのプログラムを作成してください。

なお、人数やカードの枚数がパラメーター化されていて、さまざまなケースがシミュレーションできるコードが望ましいです。

ACM-ICPC 国際大学対抗プログラミングコンテスト

ACM-ICPC とは? 2017つくば大会 ニュース 大会一覧 世界大会 サイトについて 協賛募集 言語: ●

ACM-ICPC とは?



Tweet



Like 0



G+

ACM-ICPC とは、[ACM \(Association for Computing Machinery\)](#) という計算機学会が主催する、 International Collegiate Programming Contest (国際大学対抗プログラミングコンテスト) という名前のプログラミングコンテストです。同じ大学で3人一組のチームを作り、チームでプログラミングと問題解決の能力を競う大会です。[全世界で毎年3万人以上が参加する大会に成長しています。](#)

ACM-ICPC の[世界大会](#)では、各国で開かれる地区大会を勝ち抜いたチームが、大学を代表して腕を競います。世界大会には、一つの大学から一つのチームしか参加できません。そのため、大学対抗プログラミングコンテストと位置づけられています。

- [3分でわかるACM-ICPC](#)
- [どんな問題を解くの?](#)

最近のアジア地区大会



[2014 Asia Tokyo Regional](#)

関連組織



The screenshot shows the AIZU Online Judge homepage. The top navigation bar includes links for PROBLEM, RANKING, STATUS, CONTEST, COURSE, Logout, and Register/Setting. Below the navigation is a banner for 'HOME - Volume0 - 0000'. On the right side of the banner are links for 'For beginners' and 'My Status'. Below the banner, there's a large 'QQ' logo, followed by text indicating 'Time Limit : 1 sec, Memory Limit : 131072 KB'. A link 'Japanese version is here' is also present. To the right, there are icons for a file, a speech bubble, a magnifying glass, a pie chart, and an upward arrow. At the bottom right of the banner area, it says 'PCK 2003'.

QQ

Write a program which prints multiplication tables in the following format:

```
1x1=1  
1x2=2  
. .  
9x8=72  
9x9=81
```

Input

No input.

Output

```
1x1=1  
1x2=2  
. .  
9x8=72  
9x9=81
```

The screenshot shows the AIZU Online Judge ranking page. The top navigation bar is identical to the homepage. Below it, the banner says 'HOME - Ranking'. On the right, there are links for 'For beginners' and 'My Status', and a link 'Notes on ranking criteria'. The main content is a table titled 'All Users Rank List 1-100 ordered by the number of problems'. A dropdown menu 'Select section : 1-100' is shown. The table has columns for Rank, ID, Name, Affiliation, Solved, and Rating. Each row contains a small profile icon next to the user's name.

Rank	ID	Name	Affiliation	Solved	Rating
1	namonakiaccount	namonakiaccount	The University of Tokyo	1292	2188.62
2	tozangezan	wolf	censored	1265	2309.60
3	s1190048	Tenko Chabashira	Stone College Prep High School	1175	1676.28
4	orpheus	allora	private	1168	2241.11
5	dohatsu	@dohatsutsu	University of Aizu	1164	2025.84
6	ei1333	Yun Iijima	EAGLE JUMP	1157	1962.57
7	bal4u	bal4u	Utsunomiya University	1142	1530.77
8	naoto172	Shishimaru	Nowhere	1135	1525.43
9	asl1024	asl1024	Kyoto University	1122	2160.57
10	kmcyuma	m	ERROR	1112	1763.81

プログラミングを学ぶ利点

- ・モノづくりに必要なスキル・経験が詰まっている
 - ・計画力: ゴールに向けてタスクを分解し構成する
 - ・調査力: 不足している知識を本やウェブで自習する
 - ・解決力: 動かないプログラムの原因を特定し修正する
- ・コンピュータを上手に活用できる人間になれる
 - ・例えば、ビッグデータから知識を発見する
 - ・今後は人間とコンピュータの仕事の棲み分けが進む
- ・プログラミングをきっかけに社会とつながる
 - ・アルバイト、オープンソース開発
 - ・プログラミングコンテスト、ハッカソン
 - ・日本では優秀な人材が不足している

C言語を学ぶ利点

- プログラミング言語における「最大公約数」
 - 後継の言語: C++, C#, Objective-C, D言語, ...
 - 文法が近い言語: Java, JavaScript, PHP, ...
 - C言語を制するものは他の言語も制す
 - 誰でも知っている言語 → 各種試験に出る
- ハードウェアに近い
 - CPUやGPU, メモリを自由に制御できる
 - 家電や医療用機器の制御によく使われる
 - コンピュータの仕組みを勉強するのに最適

演習・面接・試験について

講義計画 (<http://ecei-tohoku.github.io/ppa/schedule.html>)

1. プログラミング演習Aの概要と計算機システム使用法の説明
第1回課題：演習（1）
2. 第1回課題：中間面接－レポート提出
3. 第1回課題：演習（2）
4. 第1回課題：最終面接－レポート提出

5. 第2回課題：演習（1）
6. 第2回課題：中間面接－レポート提出
7. 第2回課題：演習（2）
8. 第2回課題：最終面接－レポート提出

9. 第3回課題：演習（1）
10. 第3回課題：中間面接－レポート提出
11. 第3回課題：演習（2）
12. 第3回課題：最終面接－レポート提出

13. 期末試験

演習（各課題の第1, 3週）

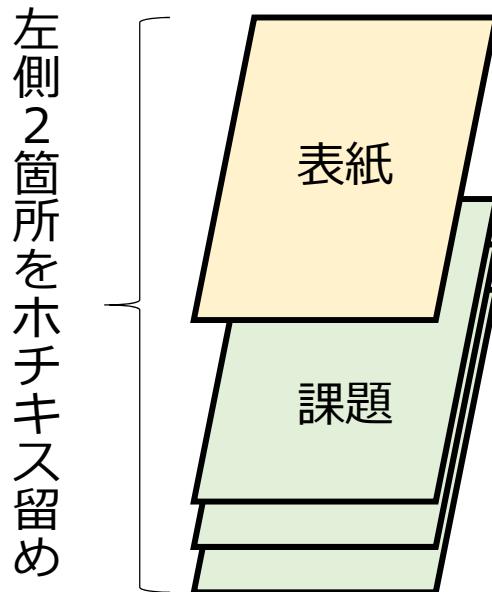
- 場所: この教室（1号館別館 1階大演習室）
- 各課題の第1週で課題の解説
- 課題の内容はウェブサイトに掲載
 - <http://ehei-tohoku.github.io/ppa/>
- 課題の進め方は後ほど解説する
 - <http://ehei-tohoku.github.io/ppa/docs/kadai.html>

レポート（各課題の第2, 4週）

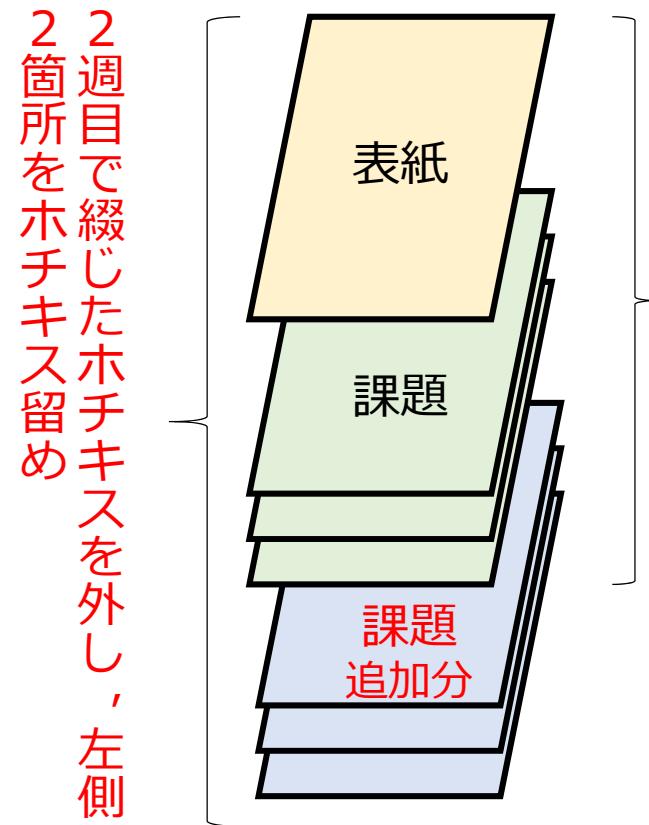
- レポートを提出し面接を受けることで加点
 - レポートを提出するだけでは採点されない
 - 2週目のみ採点できる問題があるので注意
- 提出場所: 1号館 2A, 2B室
- 提出期限: 面接日の13:00 (厳守)
- 提出時の注意 (守っていないレポートは減点)
 - 指定の表紙をレポートに付ける
 - レポートの左側2箇所をホチキス留める
 - 4週目のレポートは2週目のレポートに追加

レポートの構成

2週目のレポート



4週目のレポート



面接（各課題の第2, 4週）

- 面接開始時間の目安はWebに掲載の通り
 - 面接開始時間に関係なく、レポートは13:00が期限
- 面接の流れ（1人あたり20分）
 - 作成したプログラムについて説明する
 - 問題で要求されている事項を説明する
 - 短時間で説明できるように準備して臨むこと
- 面接の空き時間を活用して課題を進めること

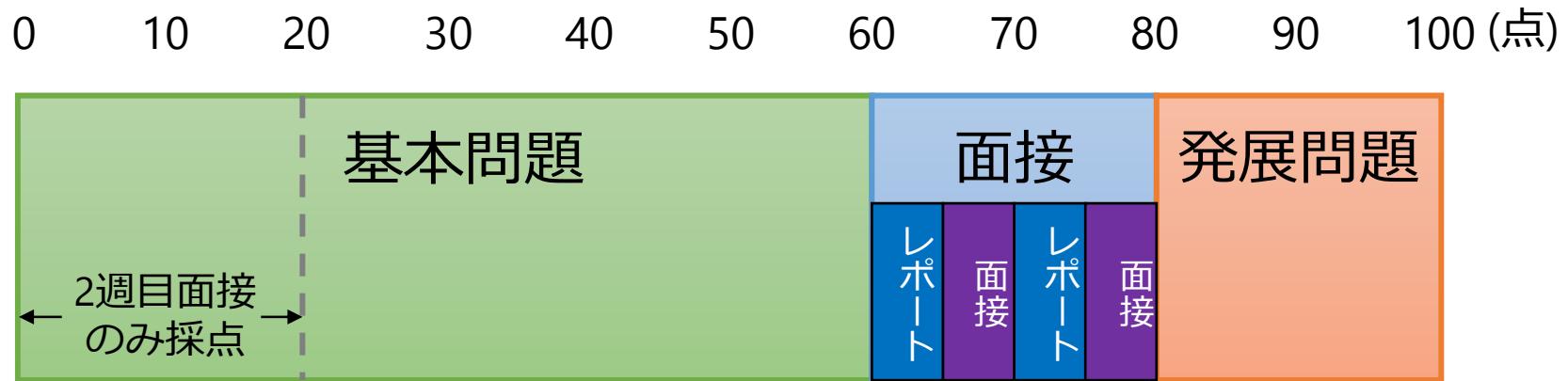
評価

- 3つの課題と期末試験の平均で算出

$$\text{点数} = \frac{1}{4}(\text{課題1} + \text{課題2} + \text{課題3} + \text{期末試験})$$

- 点数が60点以上なら合格
- 各課題の配点は次スライド参照
- 期末試験
 - 場所: 大講義棟
 - 内容: 基本問題, 課題に基づく問題, 力試し問題
 - 持ち込み不可 (簡単な問題を出題)

各課題 (満点100) の配点



- 基本問題 (60) と発展問題 (20)
 - 自動採点のテストに合格したのち、面接で認定
 - 説明問題や論説問題など、例外あり
 - 20点分は2週目の面接時のみ採点
- 面接点 (20): 2週目 (10) と 4週目 (10)
 - レポート点 (5): 整ったレポートが遅延なく提出されたか
 - 面接点 (5): 提出したレポートを適切に説明できるか

レポートのコピペは厳罰！

- コピペは誰の得にもならない
 - モノを作るためのスキルが全く身につかない
- 友達と一緒に課題に取り組むのは構わないが、提出されたプログラムが類似するのはアウト
- 提出されたプログラムのコピペ度合いを自動解析
 - 誰と誰が「仲良し」なのか丸見え
- 自分のペースで課題に取り組むこと
 - 発展問題を解かなくても平均80点になる

講義資料

- 課題・解説資料はウェブ上で公開する
- C言語の自習のための参考資料
 - 林 晴比古:『新・C言語入門 シニア編 (C言語実用マスターシリーズ)』, ソフトバンククリエイティブ, 2004年.
 - 高橋 麻奈:『やさしいC 第4版 (「やさしい」シリーズ)』, SBクリエイティブ, 2017年.
 - 倉 薫:『プログラミング学習シリーズ C言語改訂版 1 はじめてのCプログラミング』, 翔泳社, 2009年.

質問等

- 質問してください
 - 演習時間にどんどん質問してほしい
 - メールでの質問も歓迎（“質問メールの出し方”参照）
- 学生の声を参考に毎年課題を検討・改良
 - ウェブサイトの充実
 - 自動採点システムの利用範囲を学内に拡大
 - 課題や採点システムは独自に開発したもの

計算機環境について

演習室使用上の注意

- 演習室の注意
 - 必ずスリッパに履き替え，退室時に下駄箱に戻す
 - 喫煙・飲食（ジュース等も）厳禁. ゴミは出さない
 - 不要なプリントアウトは回収箱へ
- 夜間（22時～7時）・休日の入室について
 - 学生の要望により夜間・休日の入館が可能に
 - 入館する際は，別館1階東側（1号館側）のドア右側に設置されているカードリーダーに学生証を読み取らせる
 - 夜間・休日に入館ができるのは，1号館別館1階（大演習室があるフロア）のみ. それ以外の部分は立入禁止
 - 女子学生のみ2階の女子トイレまでの立ち入りを許可

計算機システム利用上の注意

- 良識を持って使用すること
 - プライバシーや著作権違反に十分注意
- セキュリティに配慮すること
 - 簡単に推測できるようなパスワードを設定しない
 - パスワードを他人に教えない
 - 同じパスワードを複数のサービスで使わない
 - ログインしたまま端末を放置しない
 - (特に夜間) 演習室の安全を脅かさない
- 本演習のIDは他の講義および3年時も使う

初回の環境設定

IDとパスワードの配布

X-terminalを選択

webブラウザ（FireFox）を起動

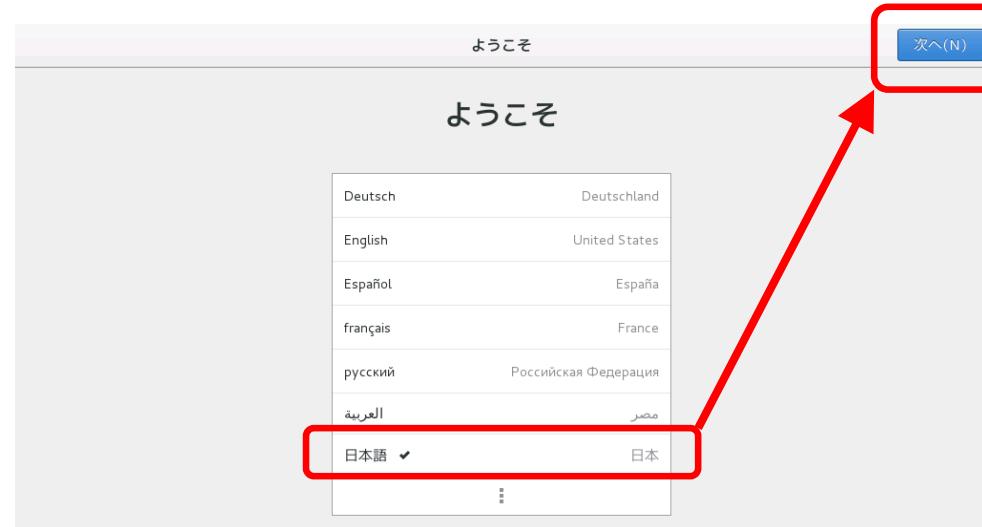
<http://eiei-tohoku.github.io/ppa/>

The screenshot shows a website for 'Programming Practice A'. At the top, there is a navigation bar with links: 'プログラミング演習A', '課題1', 'シラバス', '日程', 'C言語Tips', and 'How To'. The 'How To' link is highlighted with a dropdown menu open, listing several items: '初回の環境設定', '課題の進め方', '質問メールの出し方', 'Windows環境でC言語を書くために', and 'Mac OS X環境でC言語を書くために'. Below the navigation bar, the main content area has a heading 'プログラミング演習A' and a paragraph about the course's purpose.

- 課題1はメニューからアクセスして下さい。
- 自動採点システム（学内ののみアクセス可）
- 初回説明スライド（学内ののみアクセス可）
- 課題の進め方
- 質問メールの出し方
- 講義日程

(5) 初回ログイン 1

- 初回ログイン時のみ、GNOMEの初期設定ウィンドウが表示されます。
- 「ようこそ」画面で、「日本語」がチェックされていることを確認し、「次へ」をクリックします。



- 「入力」画面で、「日本語（かな漢字）」を選択し、「次へ」をクリックします。

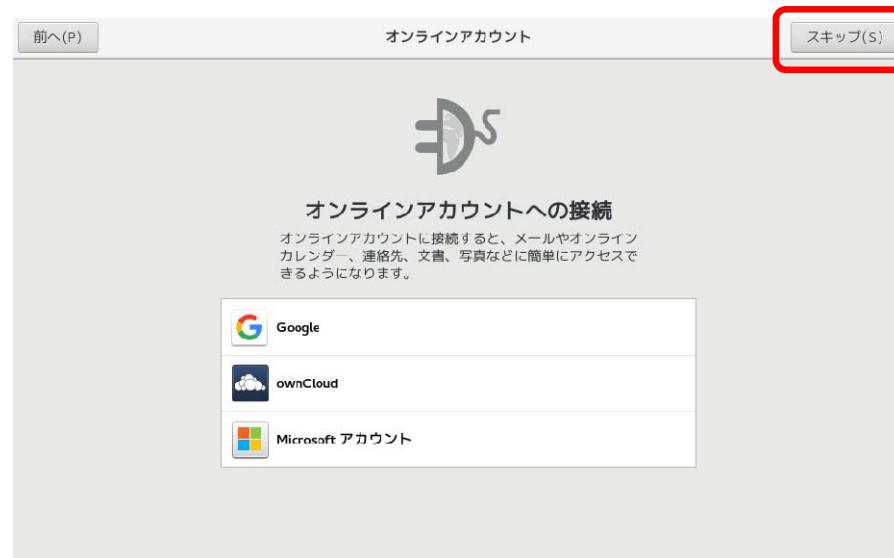


(5) 初回ログイン2

- 「プライバシー」画面で、「維持情報サービス」を任意で選択し、「次へ」をクリックします。



- 「オンラインアカウント」画面で、「スキップ」をクリックします。

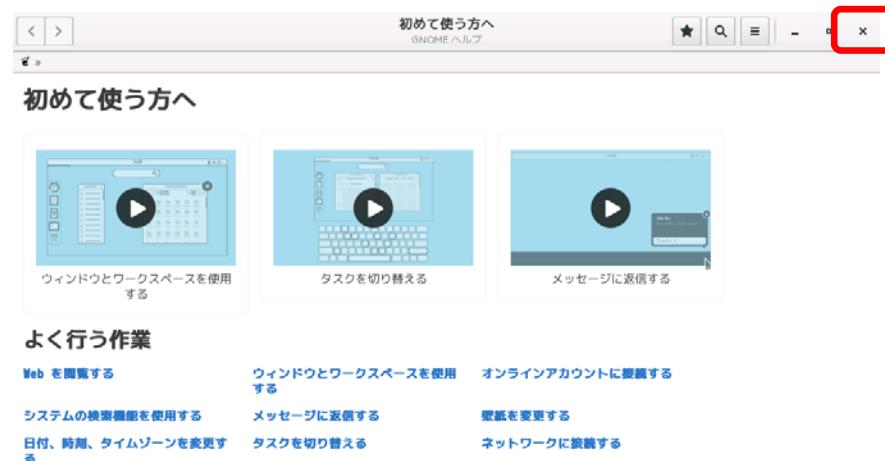


(5) 初回ログイン3

- 「設定が完了しました。」が表示されますので、「Red Hat Enterprise Linux Server を使い始める」をクリックします。



- 「初めて使う方へ」が表示されます。不要であれば「×」をクリックして閉じます。



初回の環境設定

パスワードの変更

初回演習時に教育用計算機システムのログインIDと仮passwordが配布されます。管理サイト（学内限定）にアクセスし、仮passwordを変更して下さい。なお、後で説明する自動採点システムのpasswordとは連携していませんので、個別に変更が必要です。

メールアドレスについて

教育用計算機システムのログインIDに対応したメールアドレスが発行されています。コース分けや研究室配属など、学科から重要なお知らせが届くことがありますので、各自が普段使っているメールアドレスに転送するように設定しておくとよいと思います。



ログイン

利用者ID	<input type="text"/>
パスワード	<input type="password"/>

OK

ログイン情報

最終ログイン日時： 2016/04/01 10:25:26

パスワード更新日時： 2016/04/01 10:25:14

利用者操作

[パスワード変更](#)

[プロフィール変更](#)

[反映状況確認](#)

メニュー

業務名	機能名	概要
利用者操作	パスワード変更	利用者のパスワードを変更します。
	プロフィール変更	利用者情報の変更、または変更申請を行います。
	反映状況確認	変更した情報の反映状況を確認します。

Copyright(C)2012 Fujitsu Limited, All Rights Reserved.

ログイン情報
最終ログイン日時： 2016/04/01 10:25:26
パスワード更新日時： 2016/04/01 10:25:14

利用者操作

パスワード変更

プロフィール変更

反映状況確認

パスワード変更

利用者ID	enshu02
旧パスワード	<input type="text"/>
新パスワード	<input type="text"/>
新パスワード（確認）	<input type="text"/>

パスワード設定ルール

使用禁止文字	" ,
英大文字数	0文字以上
英小文字数	1文字以上
数字数	1文字以上
記号数	0文字以上
パスワード最小文字数	8文字
パスワード最大文字数	20文字

* 利用者IDを含むパスワードはエラーとなります。

変更

メニューに戻る

初回の環境設定

パスワードの変更

初回演習時に教育用計算機システムのログインIDと仮パスワードが配布されます。管理サイト（学内限定）にアクセスし、仮パスワードを変更して下さい。なお、後で説明する自動採点システムのパスワードとは連携していませんので、個別に変更が必要です。

メールアドレスについて

教育用計算機システムのログインIDに対応したメールアドレスが発行されています。コース分けや研究室配属など、学科から重要なお知らせが届くことがありますので、各自が普段使っているメールアドレスに転送するように設定しておくとよいと思います。

メールの転送設定

(sdxxxxx@ec.ecei.tohoku.ac.jp宛のメールを各自が普段使っているメールアドレスに転送出来ます)

- Emacsで.forwardファイルを作成

```
$ emacs .forward &
```

- 次のように入力

```
name@mail.com ← 転送先のアドレス  
sdxxxxx@ec.ecei.tohoku.ac.jp ← ECEIのアドレス
```

- [Ctrl]+[x] [Ctrl]+[s]キーを押して保存

Emacsの立ち上げ方

1. 端末を開く

アプリケーション/システムツール/端末

2. 端末の画面上で下記コマンドを入力

メールの転送設定

(sdxxxxx@ec.ecei.tohoku.ac.jp 宛のメールを各自が普段使っているメールアドレスに転送出来ます)

- Emacsで.forwardファイルを作成

```
$ emacs .forward &
```

スペルミスに要注意！

- 次のように入力

name@mail.com ← 転送先のアドレス

sdxxxxx@ec.ecei.tohoku.ac.jp ← ECEIのアドレス

- [Ctrl]+[x] [Ctrl]+[s]キーを押して保存

プログラミング演習A

東北大学 電気情報物理工学科 第3セメスターの「プログラミング演習A」のウェブページ

- 課題1はメニューからアクセスして下さい。
- 自動採点システム（学内のみアクセス可）
- 初回説明スライド（学内のみアクセス可）
- 課題の進め方
- 質問メールの出し方
- 講義日程

初回の環境設定

課題の進め方

質問メールの出し方

Windows環境でC言語を書くために

Mac OS X環境でC言語を書くために

課題の進め方

はじめに

標準的には、以下の手順で課題を進めることになります。

1. プログラムを入力・編集する。
2. プログラムをコンパイルする。
3. プログラムを実行する。実行結果が正しければ手順6に進む。
4. プログラムをデバッグしてプログラムがうまく動かない原因を特定する。その後、手順1に戻る。
5. プログラムを[自動採点システム](#)に提出する。テストに失敗した場合は、実行結果から失敗原因を探り、手順1もしくは手順3に戻る。
6. レポートを印刷する

これらの手順を、以下の課題を解きながら進めてみましょう。

2つの整数を標準入力から読み取り、その和を標準出力に書き出すプログラムを作成せよ

ソースコードの入力

演習室の環境では、[Emacs](#), [Vim](#), [gedit](#), [nano](#)などのエディタを使い、ソースコードを入力します。どのエディタを使っても構いませんが、後で説明するデバッグではEmacsを使うことを想定していますので、この演習では[Emacs](#)を推奨します。

まず、Emacsで `sum.c` というプログラムを作成するため、ターミナルから以下のコマンドを入力します。ここではプログラムの名前を `sum.c` としましたが、どのような名前をつけても構いません。末尾の & は、emacsをバックグラウンドで実行することで、emacsとターミナルの両方を同時に使えるようにするためのものです。

```
$ emacs sum.c &
```

emacsを使って sum.cを入力してみましょう！

1. プログラムを入力し、保存
2. コンパイル gcc -Wall sum.c
デバッグで有用（後ほど説明）
3. プログラムの実行 ./a.out

自動採点システムへの投稿

プログラムの動作確認・デバッグが完了したら、自動採点システム (<https://130.34.192.88/>) に投稿します。自動採点システムにアクセスすると、ユーザIDとパスワードを入力する画面になりますので、ログインして下さい。初回ログイン時に、右上のメニューから仮パスワードを変更して下さい。

ログイン後、以下のようなホーム画面になります。まだプログラムを提出していませんので、全ての問題が「未提出」と表示されます。

The screenshot shows a Mozilla Firefox browser window titled "プログラミング演習A - Mozilla Firefox". The address bar displays the URL <https://130.34.192.88/home>. The main content area is titled "プログラミング演習A ホーム" and shows a user profile "ppa01". Below this, there is a table with six rows, each representing a programming problem:

問題	状態	ソースコードの提出
p10: 和 (体験用)	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>
p11: 和差算	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>
p12: コラッツの問題	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>
p13: トリム平均	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>
p14: 単純移動平均	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>
p15: Luhnアルゴリズム	未提出	<input type="button" value="参照..."/> ファイルが選択されていません。 <input type="button" value="提出"/>

A yellow rectangular box highlights the "ホーム画面" button located at the bottom right of the "ソースコードの提出" column for the p13 row.

プログラミング演習A

ユーザID

パスワード

サインイン

初期パスワードは、 IDと同じ

URL: <https://130.34.192.88/home> Go [go: up one dir, main page]

パスワードの変更
ログアウト

URL Form

問題	状態	ソースコードの提出
p10: 和 (体験用)	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p11: 和差算	✓ 成功 (10 / 10) at 2016-03-28 19:29:04 [レポート]	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p12: コラッツの問題	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p13: トリム平均	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p14: 単純移動平均	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p15: Luhnアルゴリズム	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p16: カブレカーチェンジ	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p17: 三目並べ	未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>

パスワードを変更
自分のプログラムを入れる

URL: <https://130.34.192.88/home> Go [go: up one dir, main page]

パスワードの変更
ログアウト

✓ URL Form

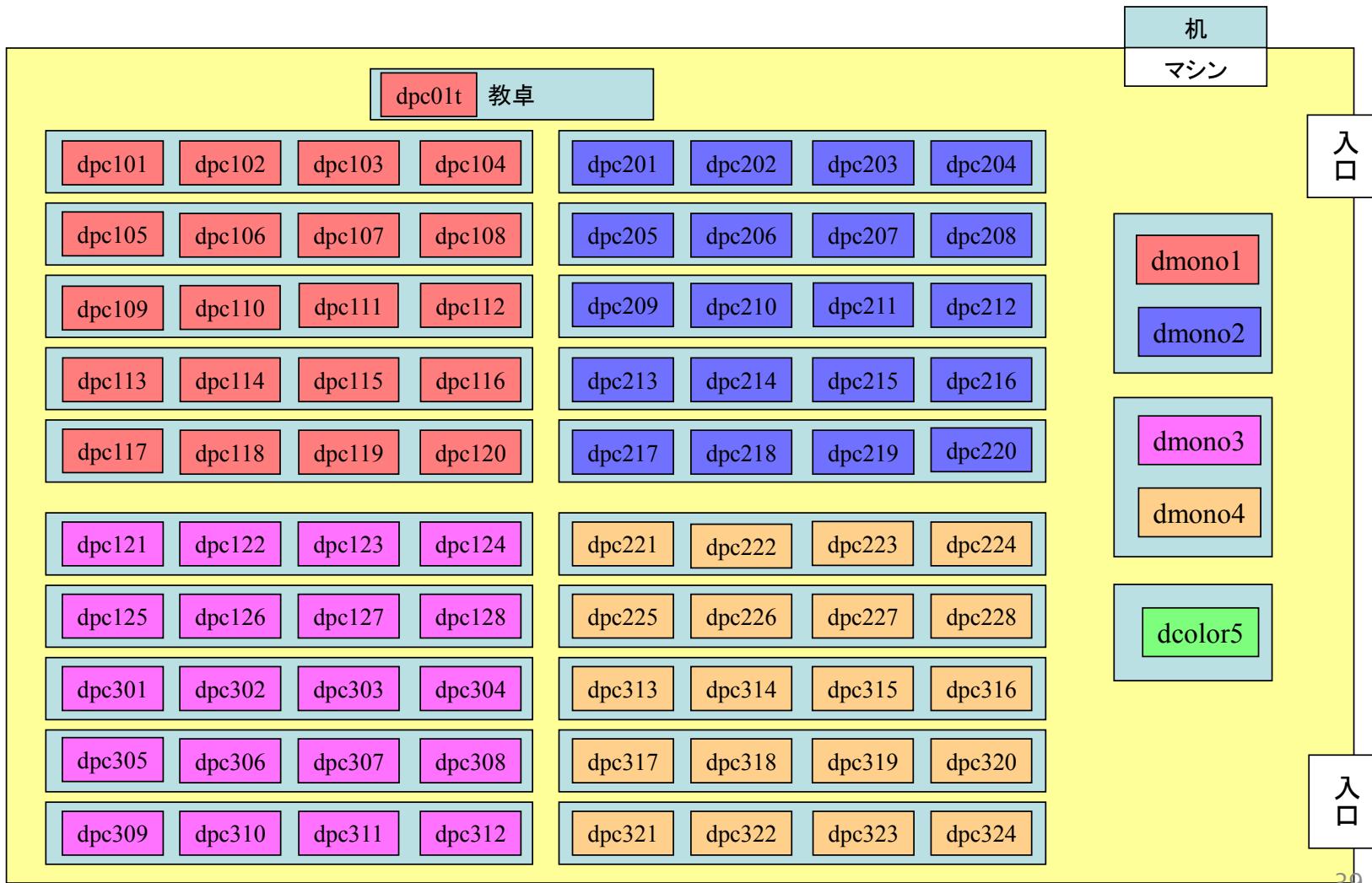
問題	状態	ソースコードの提出
p10: 和 (体験用)	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p11: 和差算	✓ 成功 (10 / 10) at 2016-03-28 19:29:04 [レポート]	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p12: コラッツの問題	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p13: トリム平均	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p14: 単純移動平均	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p15: Luhnアルゴリズム	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p16: カブレカーチェンジ	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>
p17: 三目並べ	— 未提出	<input type="button" value="ファイルを選択"/> ファイルが未選択です <input type="button" value="提出"/>

[レポート]をクリックし、印刷（今はしない！）

プリンタは5台あり、どのプリンタに接続されているかは本体の左を見る

プリンターの割当

- 大演習室は、島ごとにデフォルトプリンターが異なります。
- X-Terminal環境、Windows環境ともに、出力先は以下の様になっています。



デバッグ

プログラミング演習A

課題1 ▾

シラバス

日程

C言語Tips

How To ▾

プログラムのデバッグ

上の例では、プログラムが正しく動作したので、デバッグは不要です。ここでは、以下のような間違ったプログラムを入力してしまった場合について説明します。このプログラムでは、`scanf` 関数の引数の変数に `&` を付けるのを忘れてています（最初のうちはよく間違えるミスです）。

```
#include <stdio.h>

int main()
{
    int a, b;
    scanf("%d %d", a, b);
    printf("%d\n", a+b);
    return 0;
}
```

驚くことに、`-Wall` オプションを付けずにコンパイルすると、何のエラーも出さずにコンパイルが成功てしまいます。

```
$ gcc sum.c
```

実行すると、次のようにSegmentation Fault（セグメントエラー）が発生します。

```
$ ./a.out
2 3
セグメントエラー
```

プログラムのどこでセグメントエラーが発生したのか調べるには、デバッガを使います。デバッガの使い方は、こちらの[資料](#)や[動画](#)にまとめられていますので、各自で確認してください。

```
$ gcc sum.c
```

実行すると、次のようにSegmentation Fault（セグメントエラー）が発生します。

```
$ ./a.out  
2 3  
セグメントエラー
```

プログラムのどこでセグメントエラーが発生したのか調べるには、デバッガを使います。デバッガの使い方は、こちらの [資料や動画](#)にまとめられていますので、各自で確認してください。

なお、先ほどのプログラムは、[-Wall](#)付きでコンパイルすると警告が表示されます。

```
$ gcc -Wall sum.c  
sum.c: In function 'main':  
sum.c:6: 警告: format '%d' expects type 'int *', but argument 2 has type 'int'  
sum.c:6: 警告: format '%d' expects type 'int *', but argument 3 has type 'int'  
sum.c:6: 警告: 'a' is used uninitialized in this function  
sum.c:6: 警告: 'b' is used uninitialized in this function
```

注意事項（その他）

プログラムは仕様を満たすこと！

プログラミング演習A [課題1](#) ▾ [シラバス](#) [日程](#) [C言語Tips](#) [How To](#) ▾

1-1. 和差算【10点、2週目のみ採点】

以下のような問題を解くことを考える。

あるクラスの男子学生は女子学生よりも2人多く、クラスには全部で40人の学生がいる。このクラスの男子学生の人数、女子学生の人数をそれぞれ求めよ。

簡単な連立方程式を解けばわかる通り、男子学生の人数は $(2 + 40)/2 = 21$ 、女子学生の人数は $(40 - 2)/2 = 19$ と、与えられた数の「和」と「差」に用いて求めることができる。こうした問題は和差算と呼ばれており、小学校で習った人も多いだろう。ここでは和差算を解くプログラムを作成しよう。

問題

標準入力から2個の整数 $a, b(a < b)$ を読み込み、標準出力に $(a + b)/2$ および $(b - a)/2$ の計算結果を書き出すプログラムを作成せよ。

ただし、入力される2つの整数 a, b は半角スペースで区切られており、 $-10000 < a < b < +10000$ を満たすと仮定してよい。プログラムは以下の仕様を満たすこと。

1. $a + b$ と $b - a$ の一方もしくは両方が奇数だった場合は、**ERROR**と標準出力に出力せよ。
2. 計算結果は $(a + b)/2, (b - a)/2$ の順にスペース区切りで標準出力に出力せよ。
3. 各表示の末尾には改行文字 **\n** を付けること。

注意事項（その他）

ヒントの活用

プログラミング演習A 課題1 ▾ シラバス 日程 C言語Tips How To ▾

ヒント

変数の宣言と代入、型

- C言語では、変数を使い始めるときに、その型（整数型、浮動小数点型、などなど）も併せて宣言する必要がある。
- 林：3.1 データ型の分類、3.2 変数宣言の方法、3.3 データ型の一覧、8.9 代入演算子
- 倉：1巻 2-2 変数、2-4 型
- 高橋：3.1 変数、…、3.5 変数の利用

条件分岐（`if` 文、`else` 節）と比較演算子

- 条件分岐は `if` 文や `else` 節を用いる。
- 林：9.2 `if` 文、8.4 関係演算子、8.5 等価演算子、8.6 論理演算子
- 倉：1巻 3-1 条件分岐1- `if` 文
- 高橋：Lesson 5 場合に応じた処理

標準入力からデータを読み込むための関数、たとえば `scanf` 関数、`fgets` 関数

- 標準入力からデータを読み込むには、`scanf` 関数や `fgets` 関数を用いる。
- 空白や改行で区切られた複数の入力を読み込む場合は `scanf`、空白を含むような文字列をひとつ読み込む場合は `fgets` を用いると便利である。
- `scanf` 関数を利用する際は、入力先の変数名に「`&`」をつけることを忘れないこと。
 - （より深く知りたい人へ）`int` 型の変数 `x` を宣言しているとき、「`&x`」と書くことで、`x`へのポインタ、つまり `x` という変数のために用意されたメモリ領域のアドレスを表現できる。`scanf` 関数を用いて変数 `x` に入力値を流し込む際は、`x`（`x`という名前の変数に格納されている値）ではなく、`&x`（`x`のために用意されたメモリのアドレス、要するに、読み込んだ値をどこに仕舞っておけば良いのか）に興味があるため、`x`ではなく `&x` を用いる。
 - （より深く知りたい人へ）C言語はメモリなど計算機のハードウェアにも多少気を遣って書く必要のある言語であり、はじめはとつまづらいかもしれないが、その代わり、多くのハードウェアで非常に高速に動作する。一方で、LL（Lightweight Language）と呼ばれる

注意事項（その他）

コピペはやめてください！

インターネット上の情報を信用しない！
書籍の方が信頼できる

レポートには表紙を付けてください

The screenshot shows a navigation bar with tabs: プログラミング演習A, 課題1 ▾, シラバス, 日程, C言語Tips, and How To ▾. The main content area has two columns. The left column contains sections for 'ヒント' (Hint), '変数の宣言と代入' (Declaration and Assignment of Variables), and '条件分岐 (if文, else文) の実装' (Implementation of if-else statements). The right column contains sections for '課題1表紙' (Assignment 1 Cover), '問題1-1 【2週目のみ採点】' (Problem 1-1 [Scoring only in week 2]), '問題1-2 【2週目のみ採点】' (Problem 1-2 [Scoring only in week 2]), '問題1-3' (Problem 1-3), '問題1-4' (Problem 1-4), '問題1-5' (Problem 1-5), '問題1-6 【発展問題】' (Problem 1-6 [Development Problem]), '問題1-7 【発展問題】' (Problem 1-7 [Development Problem]), and '浮動小数点型, などなど) も併せて宣言する必要がある.' (Floating-point type, etc., must also be declared).

ヒント

変数の宣言と代入

- C言語では、変数を使い
- 林：3.1 データ型の分類
- 倉：1巻 2-2 変数, 2-4 を
- 高橋：3.1 変数, ..., 3.3

条件分岐（if文, else文）の実装

- 条件分岐は if 文や else 節を用いる。

注意事項（その他）

2016 年度プログラミング演習 A レポート

課題 1 「C 言語の基礎」

受講クラス（○で囲むこと）： A B C D

学籍番号： _____ 氏名： _____

（以下は指導員が記入）

課題 1 得点（2 週目 + 4 週目）： _____

<2 週目採点欄>

<4 週目採点欄>

注意事項（その他）

余裕を持ってテストを通しておく！

プログラミング演習 A ホーム 進捗状況▼ 管理▼ enshu02▼

問題	状態	ソースコードの提出
p10: 和 (体験用)	— 未提出	<input type="button" value="ファイルを選択"/> ファイル未選択 <input type="button" value="提出"/>
p11: 和差算	✓ 成功 (10 / 10) at 2016-03-28 19:29:04 [レポート]	<input type="button" value="ファイルを選択"/> ファイル未選択 <input type="button" value="提出"/>
p12: コラッツの問題	— 未提出	<input type="button" value="ファイルを選択"/> ファイル未選択 <input type="button" value="提出"/>

レポートに書き込みをしてもOK.
面接がスムーズに進むように準備.

注意事項（その他）

面接は順番に行う

講義室で早めの待機を！