



プログラミングの基礎及び演習

情報工学科



シラバス

1,2. ガイダンス, 復習

受講上の注意事項および受講に必要な予備知識の修得
事前学修：授業スライド「第1回 ガイダンス, 復習」と教科書p.1～p.44の熟読。(60分)

事後学修：授業スライドの用語とその意味の理解（今後の授業を理解するために必要）。順次処理, 分岐処理, 反復処理のフローとC言語によるプログラミング法, レポートの書き方を説明できること。(60分)

演習課題のプログラム及びレポートの完成と提出。(180分)



プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

- プログラミングの進め方
- 設計, コンパイル
- C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム

授業の目的

■ プログラミングの基礎の修得

- 情報工学科の学びでプログラミングは重要
- 現在広く用いられているC言語が対象

■ シラバスより

プログラミングの基礎及び演習は，ソフトウェアにより実現する機能を分析し，処理手順を記述し，プログラミング言語の基本文法を用い，構造化されたプログラムの作成の理解にある。

達成目標

■ シラバスより

(1) **C言語の基本文法**を理解し，分岐や繰返しからなる基本的な命令をC言語で記述する方法を理解する．更に，**配列**や**構造体**および機能的な処理をまとめる**関数**作成の考えを理解し，配列や構造体を扱う関数を用いたより実用的な形式のプログラムをC言語で記述する方法を理解する．



授業の進め方

■ 講義

- 講義＋練習問題

■ 演習

- 講義内容に関する演習

■ 事前学修・事後学修

- 各回の内容はシラバスを参照



授業の日程

今回授業

9/21	ガイダンス・復習	11/16	多次元配列
9/28	分岐処理	11/30	文字列
10/5	反復処理	12/7	構造体
10/12	関数の基礎	12/14	中間試験(2)
10/19	関数と記憶域	12/21	応用 1
10/26	中間試験(1)	1/18	応用 2
11/2	1次元配列	1/25	期末試験
11/9	関数と配列		

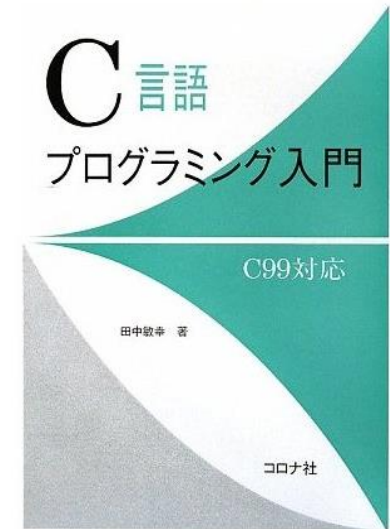
教科書

■ 「C言語プログラミング入門」

□ 田中敏幸 著

□ コロナ社

□ 2,500円



■ 各自で購入すること

■ スライドを用いて講義を行うが、教科書の対応する個所を読み理解すること

試験と成績

■ 成績

- 期末試験 . . . 50%
- 中間試験 . . . 30%
(中間試験第 1 回15%, 中間試験第2回15%)
- 演習課題のソースコードおよびレポート . . . 20%

■ レポート・演習課題

- 原則として, 毎回レポート・演習課題を課す.
- ソースコードは後述の「課題提出システム」にて提出.
- 全て提出する必要がある.



注意事項

■ 講義資料

- Google Classroomに掲示.
- 授業スライド及び教科書の関連部分を熟読すること.

■ 連絡事項

- ポータルサイト及びGoogle Classroomに掲示.
 - 定期的にポータルサイト及びGoogle Classroomをチェックすること.
-



オンラインプログラミング相談室

R5前期プログラミング相談室

プログラミングで困った学生は、プログラミング相談室を訪ねてください。
優しい大学院生が親切丁寧に対応してくれます。

相談時間

- C言語の演習について：平日（月～金），18:00～19:00
 - － 相談できる科目は、プログラミング入門、プログラミングの基礎、データ構造入門のみです。
 - － その他の科目は、科目担当者に直接相談してください。
- Java言語の演習について：火、金のみ，18:00～19:00

相談場所

- 61号館2階演習室(61-202室) （原則は対面ですが、特別な理由がある場合はオンラインも可です。）

相談までの手順

1. 申込フォームに必要事項を記入して送信してください。
2. その後、演習室を訪ねてください。大学院生が待っています。
※ オンラインの場合は、大学院生からビデオ通話(MeetまたはZoom)のURLがメールで送られてきます。
URLをクリックして入室すると大学院生が待ってるので、相談を始めてください。
(画面共有して相談しますので、必ずPCから入室してください。スマホは不可です。)

注意事項

- 受付は18:00-18:45のみ、事前予約はできません。

[申込フォームはこちら](#)

* オンラインの場合、申し込み後はGmailを開いて待機しておいてください！



イブニングクラス

- イブニングクラスは下記の曜日と時間で実施します。
 - 月曜18:00～19:00
 - 木曜18:00～19:00
- 場所： 6 1 号館の 2 0 2 室
- オンラインは下記URLを利用

- zoom情報
 - <https://nihon-u-ac-jp.zoom.us/j/87285260856?pwd=bDVxbW5KWXB6US9mNy9peERtK215Zz09>
 - ミーティングID: 872 8526 0856
 - パスコード: 042210



プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

□ プログラミングの進め方

- 設計, コンパイル
- C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム



プログラミングの流れ

1. 処理の流れを考える
2. 処理の流れを書く
3. テスト設計書を書く

設計

-
4. エディタ上でソースコードを書く
 5. コンパイルする
 6. 5. でエラーが発生したら, 2. か4. に戻る
 7. テストする
 8. 7. の結果が正しければ終了. そうでなければ, 2. か4. に戻る

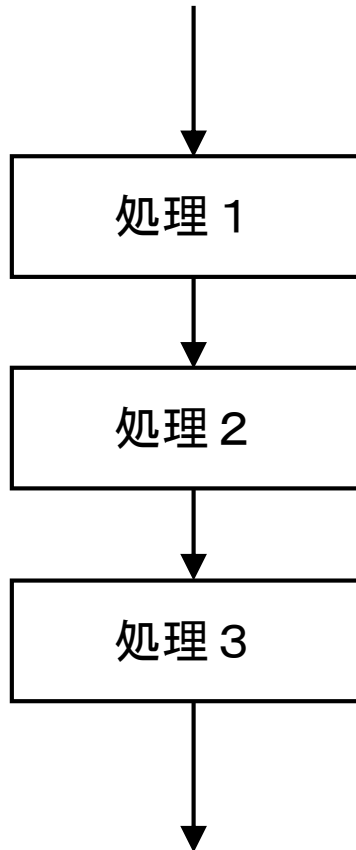
実装

設計

- 「課題に対して、どのような手順で処理をすれば良いか」を考えること
 - 順次処理・分岐処理・反復処理を用いる
 - ・「プログラミング入門及び演習」の資料をよく読んで、復習しておくこと
- 注意
 - フローチャートは自由なフローを記述できるため、**順次、分岐、反復以外の書き方もできる**ので注意すること
 - 設計を省略してはならない



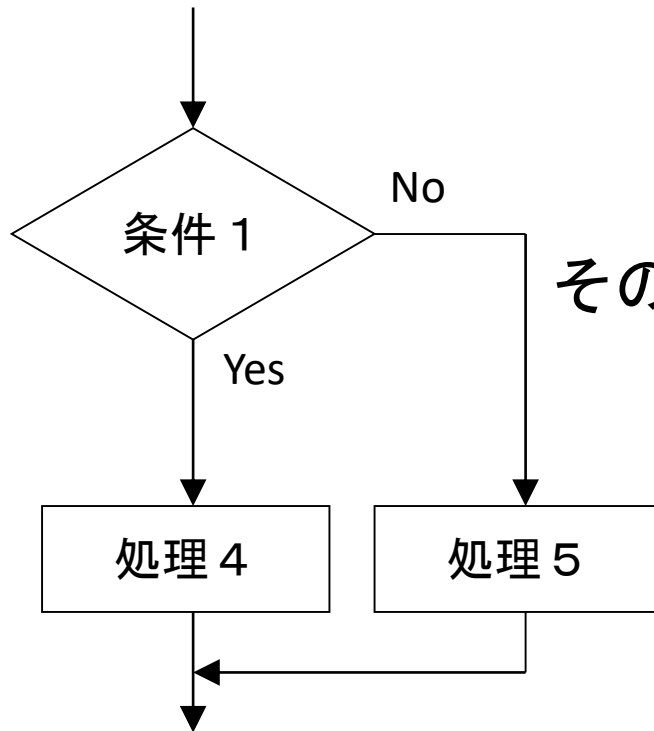
順次処理



上から下へ，順番に実行する

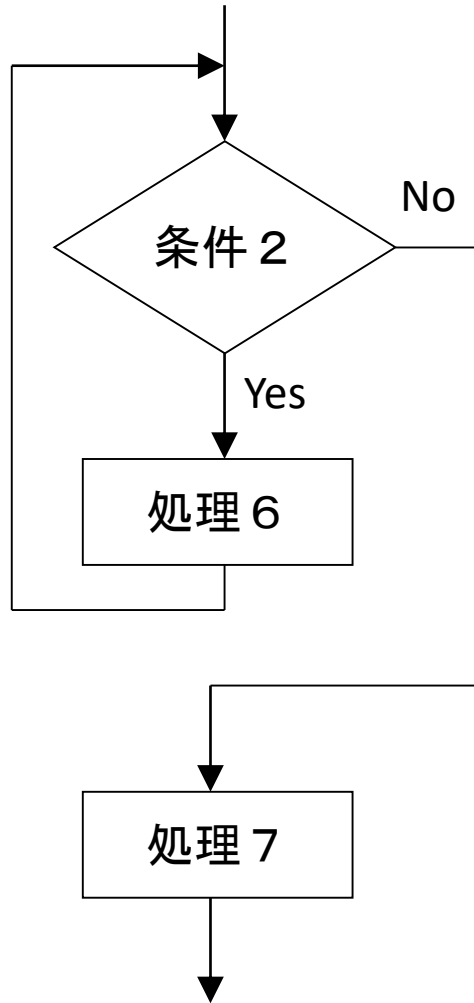
途中で，順序が変わることはない
分岐もしない

分岐処理



ある条件が真かどうかをチェック
↓
その結果に応じて、手順が枝分かれする

反復処理



ある条件が真である間
決められた処理を繰り返す



プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

- プログラミングの進め方

□ 設計, コンパイル

- C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム



C言語のプログラム

■ C言語は「コンパイラ型言語」である

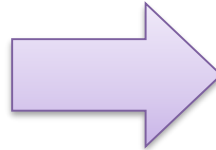
ソースファイル

```
#include <stdio.h>

int main(void)
{
    printf( "Hello¥n" );
}
```

- ・人間にとって読みやすい形式
 - ・コンピュータは直接実行不可能
 - ・拡張子.c が必要
- (例 : test1.c, sample2.c)

コンパイル



実行形式ファイル

```
464c457f 00010101 00000000
00000000 00030002 00000001
08048298 00000034 0000074c
.....
```

- ・人間が読むのは困難
- ・コンピュータは直接実行可能
- ・標準ではa.out

コンパイル : ソースファイルを実行形式ファイルに変換すること
コンパイラ : 変換を行なうためのコマンド (本授業ではgcc)



ソースファイルの作成

■ 設計 ⇒ ソースファイル

□ フローチャートをC言語に変換

- **正しい設計** : ほぼ機械的にできる
- **設計ミス** : 変換がうまくいかない

□ エディタを用いて変換したソースコードを記述

- 正しい文法の記述に基づく

エディタ : コンピュータ上で編集を行なうソフトウェア。
通常は、文章を編集するテキストエディタを指す。どのエディタを用いても
ソースコードは記述可能。

様々なエディタ : Emacs, vi, メモ帳, ワードパッド (Windows)



例：2つの整数の加算 (1)

■ 設計：（順次処理の例）

1. 2つの整数型変数 x, y の作成
2. 変数 x の値を入力
3. 変数 y の値を入力
4. x と y を加算して，結果を画面に表示



例：2つの整数の加算 (2)

```
#include <stdio.h>
```

```
int main(void) {
```

設計: 1. 変数の作成

```
    int x, y;
```

```
    printf( "xの値を入力して下さい: " );
```

設計: 2. xの入力

```
    scanf( "%d", &x );
```

```
    printf( "yの値を入力して下さい: " );
```

設計: 3. yの入力

```
    scanf( "%d", &y );
```

```
    printf( "x + y = %d\n", x + y );
```

```
    return 0;
```

設計: 4. 加算と表示

```
}
```



コンパイルとデバッグ

■ コンパイル

- `gcc -Wall -o 実行ファイル ソースファイル名`
- 例 : `gcc -Wall -o sample1 sample1.c`
- `-o` オプションを省略すると `a.out` という
実行ファイル名になる

■ エラーがない

- 実行ファイルが生成される

■ エラーがある

- シンタックスエラー, 警告, リンクエラー, ...

コンパイルエラー

- 英語で表示される
 - 自力で読もう
- 何行目でエラーが発覚したか表示される
 - ⇒ その行, または直前に注目して修正する
- 多数のエラー
 - ⇒ 一番先頭のエラーだけを修正⇒再コンパイル
 - 他のエラーは無視してよい
 - 1つずつ地道に直していく



シンタックスエラー

```
#include <stdio.h>
```

```
int main(void) {  
    printf("Hello ! ");  
    printf("Welcome to UNIX and C, ¥n")
```

```
    return 0;
```

```
}
```

セミicolon (;) を
忘れると？

ファイル名 : 行番号 : 桁:error : エラーの内容

```
% gcc -Wall -o hello hello.c  
hello.c:5:37: error: expected ';' after expression  
    printf("Welcome to UNIX and C, ¥n")
```

^

;

```
1 error generated.
```

先頭のエラーを確認し、どういう意味か理解しソースファイルを修正

警告

- warning（警告）と表示
- 警告はエラーではない
 - コンパイルは中断されず、
実行ファイルも作られる。
ただし、間違いがある可能性がある



リンクエラー

```
#include <stdio.h>
```

```
int main(void) {  
    print("Hello ! ");  
    printf("Welcome to UNIX and C. ¥n");
```

命令の名前を
間違えると？

```
    return 0;  
}
```

未定義のシンボル
(そのような名前の命令はないという事)

```
hello.c:4:2: warning: implicit declaration of function 'print' is invalid in C99  
    [-Wimplicit-function-declaration]  
    print("Hello ! ");  
    ^  
1 warning generated.  
Undefined symbols for architecture x86_64:  
  "_print", referenced from:  
      _main in hello-824ba0.o  
ld: symbol(s) not found for architecture x86_64  
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```



プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

- プログラミングの進め方
- 設計, コンパイル

□ C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム



C言語の基礎： 基本的な形式

```
/* コメント */  
#include <stdio.h>  
int main(void) {  
    ここに処理を書く  
  
    return 0;  
}
```

mainから実行が開始される



文字列の出力

- 標準出力（画面）へ文字を出力したい場合
 - `printf("もじをかく");`
- `""`で囲んだ文字をそのまま出力
 - いくつかの制御文字がある
 - `¥t` はタブ
 - `¥n` は改行文字を出力
 - `¥¥` は`¥`を出力
 - 変換指定子で変数の値を出力可能
 - `%d` は整数を出力（`""`のあとに、整数型の変数を指定）
 - `printf("合計は%dです", total);`
 - `%f` は実数を出力（`""`のあとに、実数型の変数を指定）
 - `%%` は`%`を出力



変数

■ 変数： データを入れておく箱

□ 整数型変数

- int型
- 整数だけを格納できる

□ 実数型変数

- double型
- 小数点のある実数も格納できる



変数の宣言

- 変数は，最初に宣言しないと使えない

- 例：

- 整数型のxという変数を使いたい
- プログラムの最初で， `int x;` と書く

- 例2：

- 複数の変数を宣言するとき
- `int x, y;`
- `double a, b, c;`

カンマで区切る
最後にはセミコロン



変数への値の代入

- 変数には値を代入できる

- 例：

- $x = 10;$

- ・ 変数 x に10という値を代入する

- $10 = x;$

- ・ エラー
 - ・ 必ず，左辺に変数，右辺に代入する値を書く



変数の演算 (1)

■ 変数の値を使って，計算ができる

□ 数学と同様に，乗除算は加減算より優先

□ 例：

- $x = 3 + 5;$

3と5を足した結果をxに代入

- $y = 1 + x * 2;$

xを2倍してから1を足して，yに代入

- $a = b / (c + 3);$

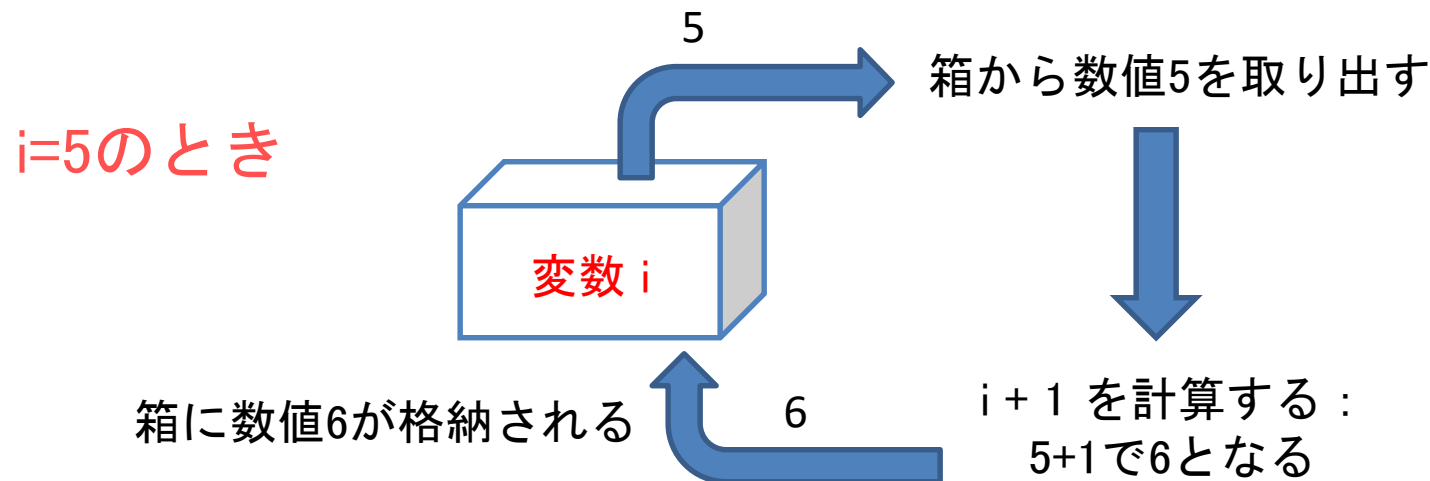
bをc+3で割って，aに代入

変数の演算 (2)

■ 変数の値を使った特殊な計算

□ 例 : $i = i + 1;$

- 変数*i*に格納されている値に1を加えて、結果を再度変数*i*に格納する
- 数学的にはあり得ない式だが、プログラムではよく使う





変数の出力

■ 変数に格納されている値を画面に表示可能

- x: int型, y: double型

- `printf("x = %d", x);`

- `printf("y = %f", y);`

- `printf("x = %d, y = %f", x, y);`

- int型は%d, double型は%fで表示できる



変数への入力

- **scanf**関数を利用して，プログラム実行中にキーボードから入力可能
 - 一つ目の引数は入力する型，二つ目以降が入力された値を代入する変数
 - **変数名の前に&記号**をつける必要あり
 - 変換指定子は，
int型は %d，double型は %lf（エル・エフ）
 - 例
 - x:int型， scanf("%d", &x);
 - y:double型， scanf("%lf", &y);

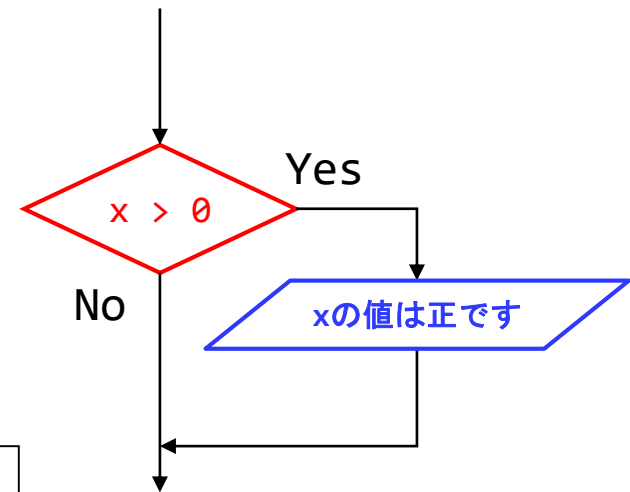


分岐処理：if文

- 変数 x の値が正であれば, " x の値は正です"と改行表示する

```
if ( 条件 )  
    条件が真の時の処理;
```

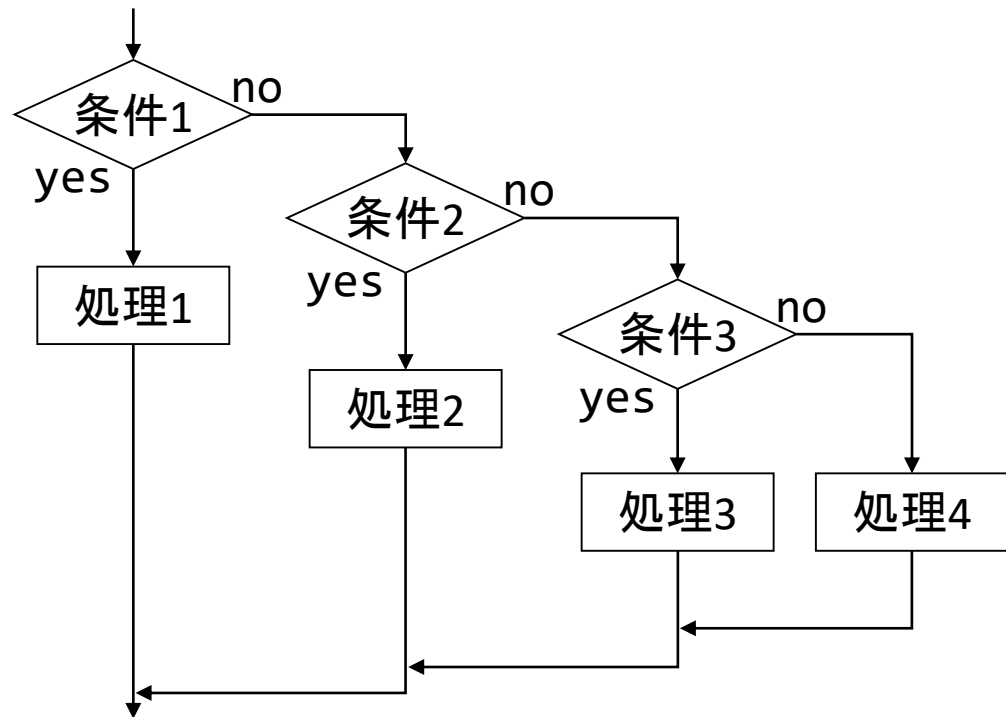
```
if (  $x > 0$  )  
    printf("xの値は正です¥n");
```



else-if 形式

- ネスト構造のうち，ある条件が偽であった時に，さらに条件分岐をする形式：else-if 形式
 - else-ifは何個でも続けることができる

```
if (条件1) {  
    処理1;  
} else if (条件2) {  
    処理2;  
} else if (条件3) {  
    処理3;  
} else {  
    処理4;  
}
```



繰り返し処理1: for文

- 条件が真である間一定の手順を繰り返す

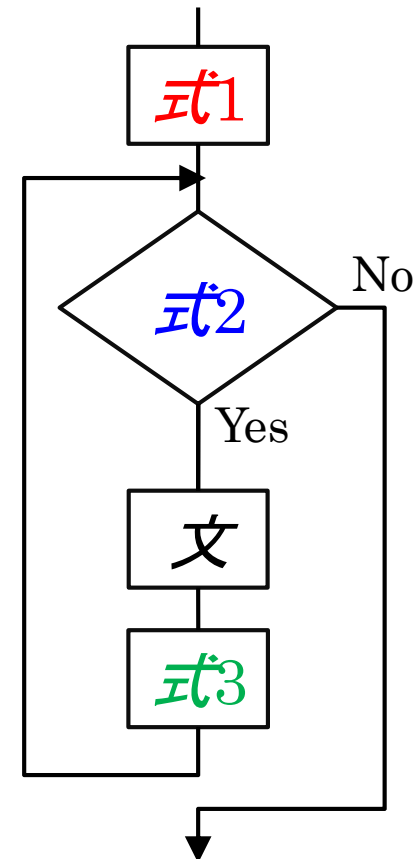
- 書式

```
for ( 式1; 式2; 式3 ) {  
    文;  
}
```

□ 式1: 初期値

□ 式2: 条件（前判定型）

□ 式3: 再設定（増分）



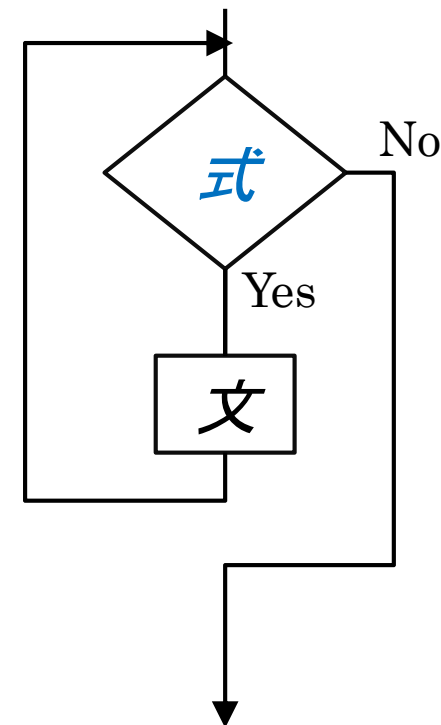
繰り返し処理2: while文

- 条件が真である間一定の手順を繰り返す

- 書式

```
while (式) {  
    文;  
}
```

- 式: 条件（前判定型）





プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

- プログラミングの進め方
- 設計, コンパイル
- C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム

演習の取り組み方

1. 設計する（手書き可）
2. 設計が完了した後，プログラミング開始
3. コンパイルして，テストする
4. 完成したら提出システムにアップロード



提出期間を設けるため，演習時間内に終わらせる必要はありません．早く出すことよりも，理解を深めるように取り組みましょう．



注意事項 (1)

- C言語のソースファイルは拡張子.c
 - 付け忘れたとき
 - Emacsの場合： [C-x C-w] ソースファイル名
 - ここで拡張子.cをつける
- ソースファイル名は"授業回数-課題番号".c
 - 第1回（本日）の演習課題1-1 → 1-1.c
 - 第1回（本日）の演習課題1-2 → 1-2.c
 - 第2回の演習課題2-1 → 2-1.c
 - :

注意事項 (2)

- インデント（字下げ）：
ソースファイル左側の空白
 - Emacsの場合，[Tab]で入力すると便利
 - [Tab]で揃わない場合
 - ・ 「それ以前の入力に間違いがある」
 - ・ 入力を中断して，間違いを探す。
無理に揃えてはいけない（バグの原因になる）

テスト

- プログラムが意図した通りに動作しているか否かを調べる必要がある
 - すべての入力を試すことは現実的に不可能
 - 効率よくテストケースを作成する必要がある
- 実際に行ったテストについて表形式で記述
 - テスト番号
 - テスト項目
 - 入力
 - 期待される出力・動作
 - テスト結果（日付）



テスト仕様書

- 入力値範囲の条件が1～64の仕様で，範囲外の場合”範囲外”と表示し，範囲内の場合に入力値を表示するプログラムの場合

テスト番号	テスト項目	入力	期待される出力	テスト結果
1	下限の境界(off)	0	範囲外	9/24
2	下限の境界(on)	1	1	9/24
3	上限の境界(on)	64	64	9/24
4	上限の境界(off)	65	範囲外	9/24

ここまでは設計終了後に作成

プログラム作成後にテストし，
期待通りなら実施日を記入



演習課題の提出について

■ プログラム

- 原則として課題提出システムにアップロード

■ 筆記課題

- レポートに記載する項目は都度示す
 - ・ フローチャート, テスト仕様書, 変数設計, 考察など

■ 注意

- 課題を**高い完成度で作成**すること
- 他人の課題を写した場合, 原則は再提出とする



筆記課題（レポート）執筆時の注意点

- セクション番号，タイトルを付ける
- フォントの大きさ，種類を適切に
- フローチャートは見やすく，
分岐にはYes/Noを必ず書く
- ページ番号をフッターに付ける



他にも様々な点があります。

レポートの書き方の本を読むことを推奨します。



プログラミングの基礎 第1回

■ ガイダンス

- 授業の目的, 進め方, 日程, 成績評価

■ 復習

- プログラミングの進め方
- 設計, コンパイル
- C言語の基礎

■ 演習

- 演習の取り組み方
- 演習課題提出システム



演習課題提出システム

- ユーザ登録
- ログイン
- 提出と結果確認



ユーザ登録

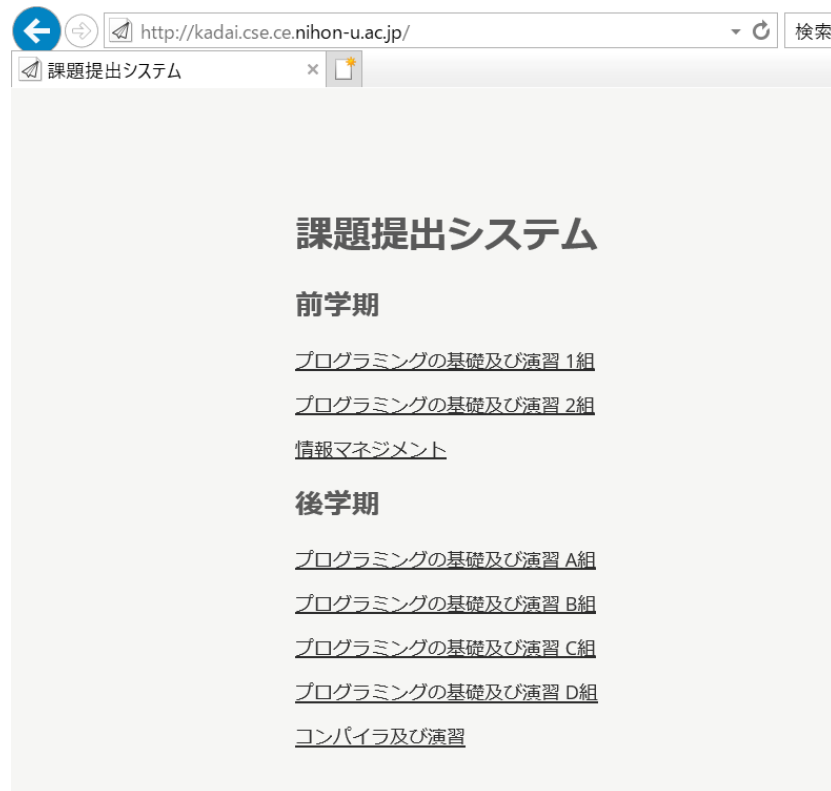
初回利用時，次の手順でユーザ登録を行なう

1. 演習課題提出サイトにアクセス
2. 各自のクラスの選択
3. 登録の開始
4. 登録に必要な項目の入力



ユーザ登録: 手順1 - 演習課題提出サイト

- ブラウザから，次のURLにアクセスする。
<https://kadai2.cse.ce.nihon-u.ac.jp/>





ユーザ登録: 手順2 - クラスの選択

- 自のクラスのリンクをクリックする.





ユーザ登録: 手順3 - 登録の開始

■ "Register"をクリックする

Login - Sharif Judge

prog.ce.nihon-u.ac.jp/kiso-c/login

プログラミングの基礎及び演習 C
組

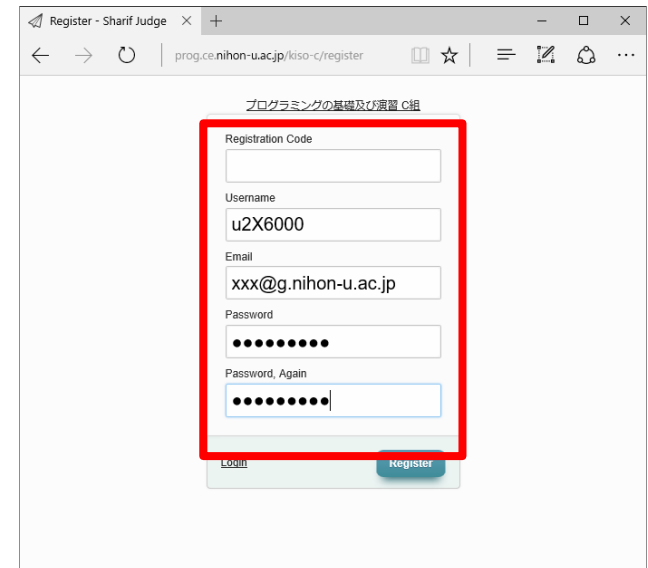
Username

Password

[Register](#) [Reset Password](#) [Login](#)

ユーザ登録: 手順4 - 項目の入力

- 各項目を入力し, "Register"をクリックする
 - "Registration Code": 次ページに提示
 - "Username": u学籍番号 (演習環境と同じ)
 - "Email": 日大の公式アドレス (NUメール)
 - "Password": パスワード
 - "Password, Again"



Register - Sharif Judge

prog.ce.nihon-u.ac.jp/kiso-c/register

プログラミングの基礎及び演習 C組

Registration Code

Username

u2X6000

Email

xxx@g.nihon-u.ac.jp

Password

Password, Again

Login Register



Registration Code

2355304



演習課題提出システム

- ユーザ登録
- ログイン
- 提出と結果確認



ログイン

■ 各クラス画面

1. Username, Passwordを入力
2. "Login"をクリック

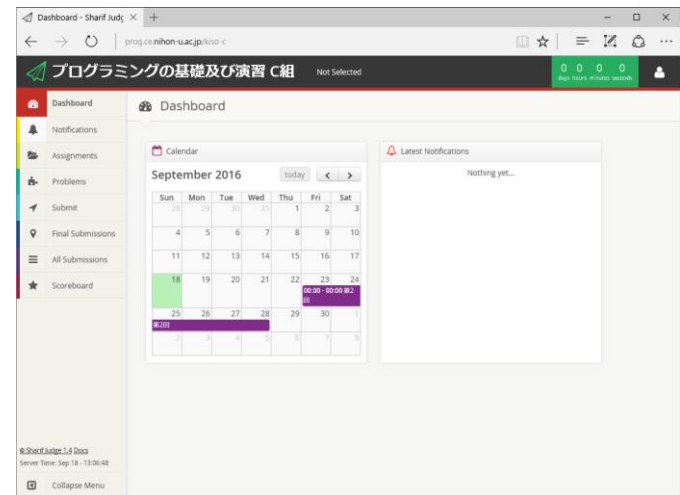
プログラミングの基礎及び演習 C組

Username
u2X6000

Password
.....

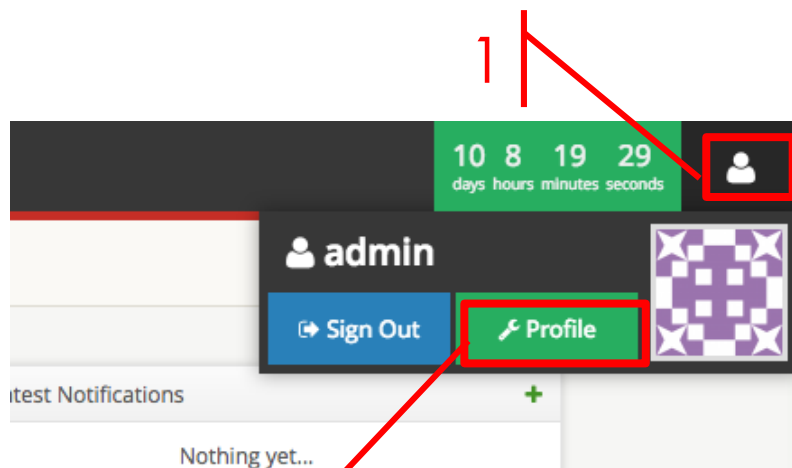
Remember / Reset Password Login

ログイン



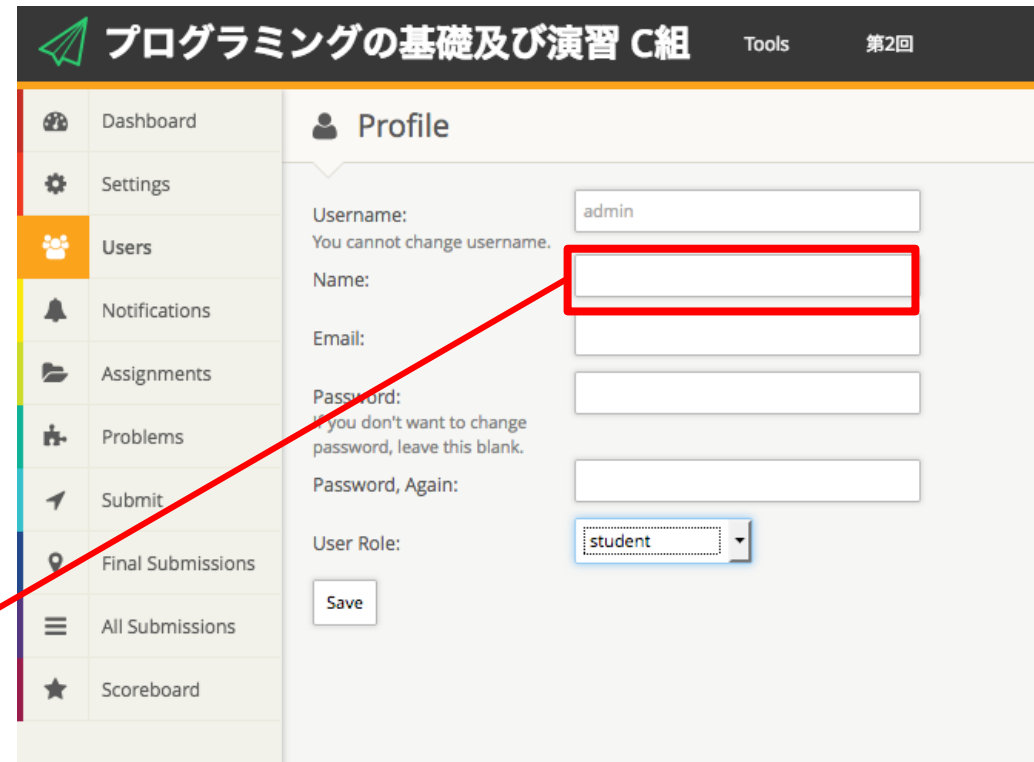
プロフィールの変更

1. 右上の人型アイコンをクリック
2. "Profile"をクリック
3. 氏名を記述



2

氏名を記入
“日大 太郎”



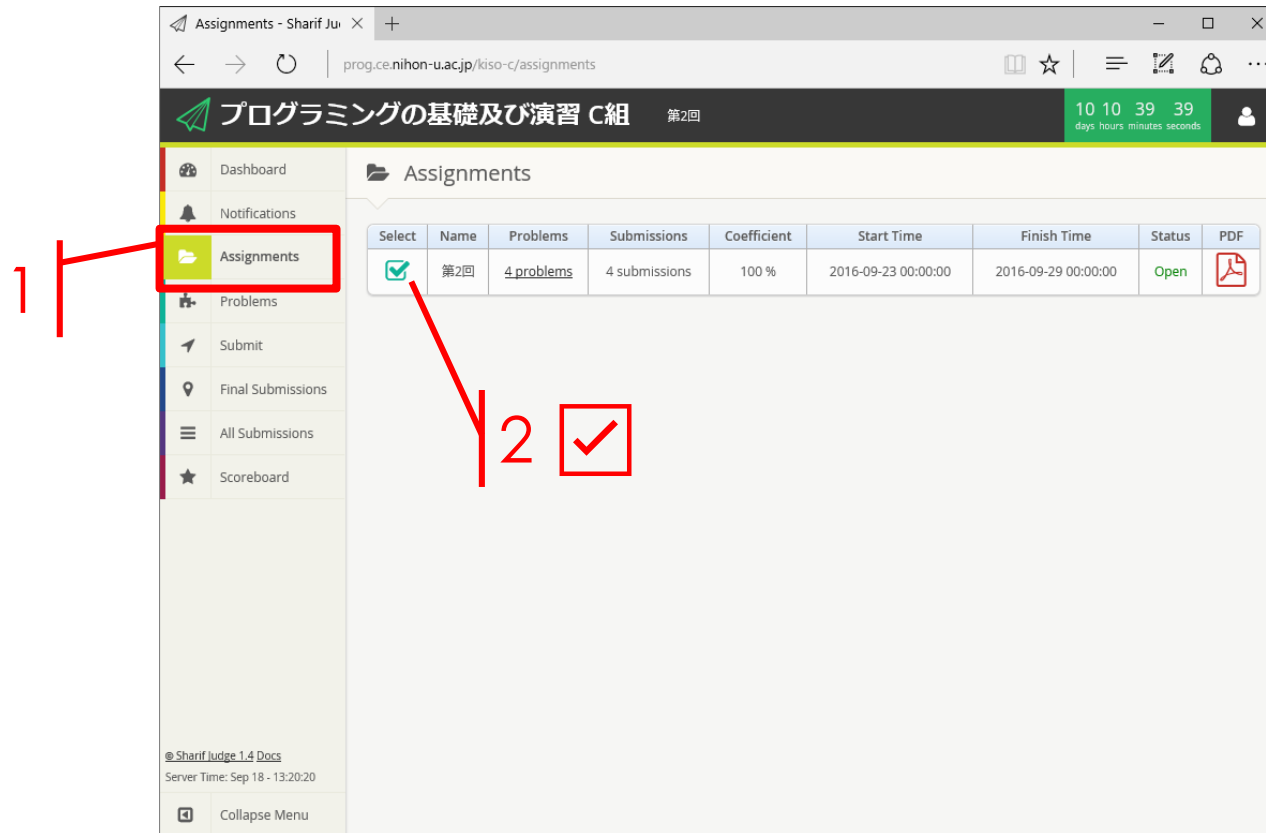


演習課題提出システム


- ユーザ登録
- ログイン
- 提出と結果確認

提出とテスト – 実施回の選択

1. "Assignments"をクリック
2. 実施回の"Select"に☑を入れる



The screenshot shows the 'Assignments' page of the Sharif Judge system. The sidebar on the left contains a menu with items: Dashboard, Notifications, Assignments (highlighted with a red box and labeled '1'), Problems, Submit, Final Submissions, All Submissions, and Scoreboard. The main content area displays a table of assignments. The table has columns: Select, Name, Problems, Submissions, Coefficient, Start Time, Finish Time, Status, and PDF. The first row of the table is for '第2回' (2nd Round) and is highlighted. In the 'Select' column of this row, there is a checked checkbox, which is highlighted with a red box and labeled '2'.

Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF
<input checked="" type="checkbox"/>	第2回	4 problems	4 submissions	100 %	2016-09-23 00:00:00	2016-09-29 00:00:00	Open	

© Sharif Judge 1.4 Docs
Server Time: Sep 18 - 13:20:20
Collapse Menu



提出とテスト – 課題の選択と提出

1. "Problems"をクリック
2. 課題を選択
3. 提出

1. 言語を選択
2. ファイルを指定
3. "Submit"

Problem 1 - Sharif Judge

prog.ce.nihon-u.ac.jp/kiso-c/problems

プログラミングの基礎及び演習 C組 第2回 10 10 22 4 days hours minutes seconds

Dashboard

Notifications

Assignments

Problems

Submit

Final Submissions

All Submissions

Scoreboard

演習課題2

以下に示すように、正の整数を読み込んで、その値が奇数であるか偶数であるかを表示するプログラムを作成せよ。
(ヒント: 剰余算を行う算術演算子%を使用してみよう)

ただし、コロン(:)は全角を用いて、以下の例と同じ出力をさせること。

実行例:

正の整数を入力してください: 5
5は奇数です

正の整数を入力してください: 6
6は偶数です

第2回

#	Problem	Score	Upload Only
1	演習課題 2	100	No
2	演習課題 3	100	No
3	演習課題 4	100	No
4	演習課題 5	100	No

Submit

C

参照...

Submit



提出とテスト – テストと結果

- "All Submissions" (提出後に自動で移動)
- 結果の確認
 - "Status"が"Pending"のときは, 再読み込み

10 10 15 21
days hours minutes seconds

Dashboard All Submissions Excel

All Submissions of 第2回

⚠ You cannot change your final submissions after assignment finishes.

Final	Problem	Submit Time	Score			Language	Status	Code
			Score	Delay %	Final Score			
✓	1	2016-09-18 13:44:23	100	No Delay 100%	100	C	100	Code



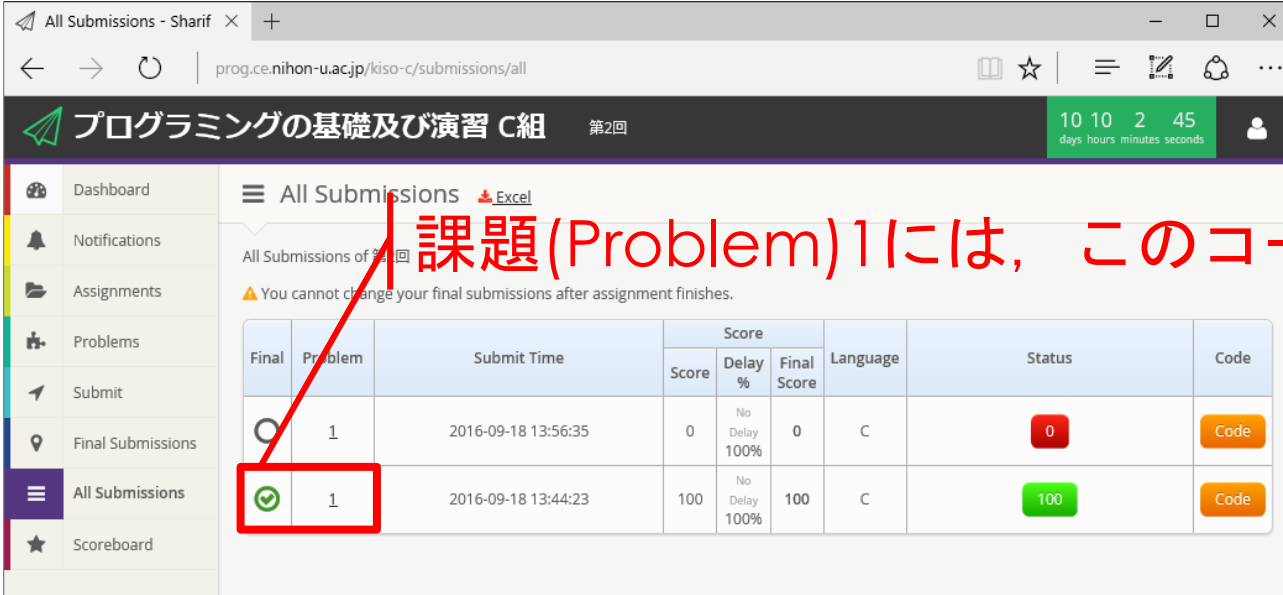
1つ以上のテストが実施されています。

課題提出は期間内は何度でも可能. 100点を目指しましょう.

提出とテスト – 提出結果の選択

複数のコードを提出した場合、
最終提出のコードを選択可能

1. "All Submissions"
2. 提出したいコードの"Final"に☒



プログラミングの基礎及び演習 C組 第2回 10 10 2 45 days hours minutes seconds

All Submissions Excel

All Submissions of 課題 (Problem) 1

⚠ You cannot change your final submissions after assignment finishes.

Final	Problem	Submit Time	Score			Language	Status	Code
			Score	Delay %	Final Score			
<input type="radio"/>	1	2016-09-18 13:56:35	0	No Delay 100%	0	C	0	Code
<input checked="" type="radio"/>	1	2016-09-18 13:44:23	100	No Delay 100%	100	C	100	Code

課題 (Problem) 1 には、このコードを提出



提出とテスト – 提出結果の確認

1. "Final Submissions"

Final Submissions - Sha x +

prog.ce.nihon-u.ac.jp/kiso-c/submissions/final

プログラミングの基礎及び演習 C組 第2回 10 9 51 18 days hours minutes seconds

Dashboard

Notifications

Assignments

Problems

Submit

Final Submissions

All Submissions

Scoreboard

Final Submissions of 第2回

Problem	Submit Time	Score			Language	Status	Code
		Score	Delay %	Final Score			
1	2016-09-18 13:44:23	100	No Delay 100%	100	C	100	Code
2	2016-09-18 14:07:39	100	No Delay 100%	100	C	100	Code
3	2016-09-18 14:07:56	100	No Delay 100%	100	C	100	Code
4	2016-09-18 14:08:09	100	No Delay 100%	100	C	100	Code

@ Sharif Ludge 1.4 Docs
Server Time: Sep 18 - 14:08:41
<http://prog.ce.nihon-u.ac.jp/kiso-c/problems>

All Submissions

All Submissions - Sha x +

prog.ce.nihon-u.ac.jp/kiso-c/submissions/all

プログラミングの基礎及び演習 C組 第2回 10 9 51 35 days hours minutes seconds

Dashboard

Notifications

Assignments

Problems

Submit

Final Submissions

All Submissions

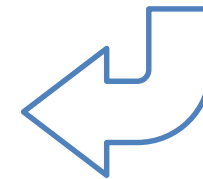
Scoreboard

All Submissions of 第2回

You cannot change your final submissions after assignment finishes.

Final	Problem	Submit Time	Score	Delay %	Final Score	Language	Status	Code
✓	6	2016-09-18 14:08:09	100	No Delay 100%	100	C	100	Code
✓	3	2016-09-18 14:07:56	100	No Delay 100%	100	C	100	Code
✓	2	2016-09-18 14:07:39	100	No Delay 100%	100	C	100	Code
○	1	2016-09-18 13:44:23	0	No Delay 100%	0	C	0	Code
✓	1	2016-09-18 13:44:23	100	No Delay 100%	100	C	100	Code

© Sharif Ludge 1.4 Docs
Server Time: Sep 18 - 14:08:41
Collapse Menu





補足：演習課題提出システム

■ 100点でない場合の差異

提出したファイル名

テスト番号と結果 (ACCEPT / WRONG)

提出ファイルの出力
'<' で始まる行

期待される出力
'>' で始まる行

```
compiler02_step3_wrong.c | Submit ID: 424 | Username: | Problem: 3
Test 1
WRONG
---
< a (n) b (n) c (y) d (y) e (y) f (y) g (y)
---
> a(Y) b(Y) c(N) d(N) e(N) f(N) g(N)
Test 2
WRONG
3c3
< x (y) y (y) z (y)
---
> x(N) y(N) z(N)
```

【演習課題】

- 演習課題提出システムの「第1回」（全2問）を実施する
- 演習環境(gw.cse.ce.nihon-u.ac.jp)にリモートログインし，emacsでプログラムコーディング，gccでコンパイル，テスト動作確認，プログラムを完成させる
- WinSCP等でソースファイルをPCへ転送
 - 注意：PCでファイルにソース追記する場合はTeraPadで追記し，「文字/改行コード指定保存」で文字コード「UTF-8N」で保存
- ソースファイルを演習課題提出システムへ授業翌週水曜日の午後5時までに提出
- 各問の得点（100点満点）を成績に算入

【レポート】

- 演習課題提出システムの
演習課題1-1について、下記内容を記載し、
印刷し、授業時に配布した表紙をつけた、
”レポート”を提出せよ。
 - プログラム設計（フローチャート）をastahにて作成し「ツール」，「画像出力」にてPNG形式ファイルを作成し”レポートファイル”に貼り付け.
 - テスト仕様書を”レポートファイル”に貼り付け.
 - ソースコード，コンパイル結果，実行結果をscriptコマンドでファイルにし印刷し”レポートファイル”に貼り付け.
- 授業翌週水曜日の午後5時までに
55号館304室前のレポート提出小箱へ提出