

ソフトウェア概論 A/B

-- 模擬試験 --

数学科 栗野 俊一 / 渡辺 俊一 (TA: 石川 溪 / 山脇 直樹 [M1])

2019/07/19 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20190719
- 色々なお知らせについて
 - 栗野の Web Page に注意する事
<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>
- VNC Server Address : 10.9.154.227
 - Password : vnc-2018
- 廊下側の一列は遅刻者専用です(早く来た人は座らない)
- 講義開始前に済ませておく事
 - PC の電源を入れておく
 - ネットワークに接続しておく
 - 今日の資料に目を通しておく
- 講義前の注意
 - 講義前は、栗野は準備で忙しいので TA を捕まえてください
- やる気のある方へ

今後の予定(後ろから)

□ 2019/07/26 (講義最終日)

- 前期試験を行います (試験時間は、90 分 ~)

- ▶ 試験開始 30 分後に出席を取ります

- 時間内に結果を CST Portal に提出してください

- ▶ ギリギリに提出しようとするサーバーが対応できない可能性あり

□ 試験の形式

- ファイルに入った問題をダウンロード

- ▶ ファイル内に個々に問題が入っているので、それを見て解く

- ▶ 問題は、各自異なる(ので答も異なる) / 回答は、ファイルの形で、ポータルに upload する

- 持ち込み : Note-PC を含め、何んでも可 (もう一台の PC / 本 / ノート..)

- 禁止事項

- ▶ 音を出しては駄目 (会話不可 / 携帯電話不可 / チャット可) / 物の貸し借りは駄目

□ 質問

- 問題が「変」と思ったら、手を挙げてください

□ 本日(2019/07/19)の予定

- 模擬試験

前回(2019/07/12)の復習

□ 前回(2019/07/12)の復習

○ 関数の作成方法

- ▶ 値を返す「数学的」関数の定義 / 命令の一部に名前を付ける

○ 制御構造

- ▶ 条件分岐 (if 構文)
- ▶ 繰返し (再帰呼出し)

お知らせ

- 本日(2019/07/19)の予定
 - 前期のまとめ
 - 模擬試験

今週 (2019/07/19) の課題

□ 今週 (2019/07/19) の課題

- 模擬試験の結果を提出する

- ▶ 複数ファイルの提出になるので、注意

先週 (2019/07/12) の課題

□ 先週 (2019/07/12) の課題

○ 課題 20190712-01:

- ▶ ファイル名 : 20190712-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二つの整数の積を返す関数
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

○ 課題 20190712-02:

- ▶ ファイル名 : 20190712-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 自然数の階乗を返す関数
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

○ 課題 20190712-03:

- ▶ ファイル名 : 20190712-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二つの非負の整数の最大公約数を返す(ユークリッドの互除法)
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

試験の開始

□ 試験の開始手順

- 問題ファイル (QQQQ.zip) を Web よりダウンロード
 - ▶ QQQQ は自分の学籍番号 (今回は 9999.zip)
- 問題ファイルを展開 (展開先はどこでも OK)
- question フォルダに回答/問題ファイルがある
 - ▶ 解答ファイル : QQQQ-a.txt
 - ▶ 問題ファイル : q で始まるファイル(色々ある)
- q.txt をサクラエディタで開いて読む
 - ▶ 問題の詳しい内容は、更に別のファイルに書いてある
 - ▶ それぞれの問題を解き、解答する

解答の提出

□ 問題の解答の提出形式は次の二通り

○ q.00 ~ q.11, q.15, q.17 の解答：一問毎にそれぞれ一ファイルを解答として提出

▶ それぞれ、QQQQ-a00.c ~ QQQQ-a11.c, QQQQ-a15.c, QQQQ-a17.c として提出

○ それ以外の解答：QQQQ-a.txt にまとめて書きこむ

▶ QQQQ-a.txt に答を記入し、提出

▶ 基本は PC の表示をコピー・ペーストして欲しい

▶ 英数字や、半角で入力できる記号は、半角で入力する

▶ 逆に半角カナは利用しない

○ 最大 15 個のファイルを CST Portal に提出する

▶ できた分だけ提出すればよい(提出されている分だけ採点対象になる)

▶ QQQQ-a.txt も解けた答だけ記入すればよい(全て記入しなくてもよい)

○ 時間内に提出しないと提出できなくなる !!

□ 試験の出欠：次の何れかで出席扱いになる

○ 試験開始 30 分に出席カードを配布するので、それに記入・提出する

○ CST Portal に、一つでもファイルが提出できている

模擬試験と本番の違い

項目	模擬テスト	本番
開始時間	既に参照可能	試験開始時
提出期間	時間内 (後に翌週まで延長)	時間内 (時間切れ)
問題	全員共通	各人異
前半と後半	共通	問題の
問題ファイル	20180717-9999.zip	20180724-QQQQ.zip
QQQQ-a.txt	9999-a.txt の名前を変更	
会話	可能 (おおいに相談しよう)	不可 (無言で)
質問	可能 (相談する最後の機会)	不可 (問題の不備)
PC 対応	可能 (相談する最後の機会)	不可 (一切)

本日の目標

□ 本日の目標

○ 本番の試験に備える

- ▶ 環境は大丈夫か？ (Note-PC / Soft / Network ..)
- ▶ 課題は解けるか？ (模擬試験の問題 / 過去の課題 ..)

○ PC の「ヘルス証明」

- ▶ トラブルが解っている人：申し出る / 今日の午後に解決できれば..
- ▶ トラブルのない人：課題を提出する / 今週迄は「健康だった」という事

○ 「ヘルス証明」ができている人は、来週トラブルがあっても配慮する

- ▶ 上記の両方ができていない人は、来週、PCトラブルがあっても配慮しない
- ▶ cf. ド=モルガン

○ ポイント

- ▶ 「資料を見ながら、作業ができる」ようにする事

□ 試験日について

○ 体の容態が悪いならば、無理為ず、メールして休む

- ▶ 試験日は、相談して、翌週以降にやりましょう..

前期の復習 1：学習目標の確認

□ 目標の確認

○一言で言うと.. プログラムが書けるようになりたい !!

▶では、「プログラムが書ける」という事はどうゆう事か？

□ 「プログラムが書ける」ために学ばなければならない三つの要素

○[プログラム理解] プログラムとは何か？

▶「書く」対象(目標)が解らなければならないので、それ自身を学ぶ必要がある

○[C 言語] プログラムはどう表現すれば良いか？

▶「書く」という行為は「表現」行為なので、表現手段としての「C 言語」を学ぶ必要がある

○[操作手順] プログラムはどう作り、どう利用すれば良いか？

▶「書く」だけでは意味がない、それを「動かす」方法を学ぶ必要がある

前期の復習 2 : 操作手順

□ プログラムの作成/実行の手順 (コンパイル言語の場合)

○ 起草 [頭] : プログラムの仕様/設計 (アイデア)

- ▶ 「どんな(仕様)プログラムを「どのような手法(設計)」で作成するか
- ▶ 「プログラムを実行したら『どうなる』か？」を「*予め* 考え」ておく

○ 編集 [エディタ] : ソース(Source) プログラムの作成 (*.c)

- ▶ 自分が望むプログラムをプログラミング言語(C 言語)で表現する

○ 翻訳(コンパイル) [コンパイラ] : オブジェクト(Object) プログラムの作成 (*.o)

- ▶ ソースを翻訳して、オブジェクトを作成する
- ▶ 「表現」に問題があれば、ここで確認(エラーになる)でき、最初に戻って検討する

○ リンク [リンカ] : 実行ファイルの作成 (*.exe)

- ▶ オブジェクトとライブラリを結合して実行ファイル(アプリファイル)を作成する
- ▶ 「部品の揃え方」に問題があれば、ここで確認でき、最初に戻って検討する

○ 実行 : 実行結果の確認 (実行結果:画面出力など)

- ▶ プログラムを実行し、その結果を見る事
- ▶ 適切な結果が得られなければ、最初に戻って検討する

□ 確認実習

- 「Hello World」プログラムを作成、実行してみよう

前期の復習 3 : プログラム理解

□ プログラムとは何か？

○ 計算機への(複数の)指示をまとめたもの (狭義)

- ▶ 複数の(単純な)指示を適切に組合せる事により、意味のある(複雑な)結果が得られる
- ▶ 命令の組み合わせ: 文字列を 1 文字ずつ短かくしながら表示する
- ▶ 意味のある結果 : 文字による三角形が表示される

□ プログラムを作成するためにどう考えるか

○ 基本は「分割統治法」を利用

- ▶ (Top-Down 設計) 全体の問題を複数の小さい問題に分割
- ▶ (個々の問題の解決) 個別撃破
- ▶ (Bottom-Up 実装) 解決案を組合せて全体を作る

○ 例

- ▶ 全体の問題(部屋の片づけ/手がつかない)
- ▶ 分割(取り敢えず机の上から、次にベット上, 床は最後に..)

○ 分割は「再帰的に」行い、「直にできる事」まで「分解」する

- ▶ 例 : ベッドは、まず上に乗っている物をどかし、シーツをかえ、下に隠してあるごによごによを...

○ 分解が済んだら、個々に解決し、それを組上げる事により、問題全体を解決

- ▶ 例 : 色々な所から出たゴミは、袋にまとめて、明日の朝出しておく..

前期の復習 3 : プログラム設計

□ プログラムの作成法

○ 問題を分割すれば良い .. だとしても ???

▶ 「何(what)」を「どのような(how)」に、「どこ(when)」で分割すべきか？

▶ 例 : 「ベットの掃除」を「頭側」と「足側」に分ける事が意味があるか？ 右と左では？

□ 分割の「パターン」を身に付ける

○ 時間の前後関係 (X は A をやってから B をする) で分ける

▶ 状況 : ベットを片漬けるとゴミが出るが、ゴミ箱が一杯で捨てられない

▶ 時間分割 : まず、ゴミ箱内のゴミを捨てて、次に、ベッドから出たゴミを入れる

○ 条件に応じて対応 (P の時は A をやって、そうでなければ B をする) を分ける

▶ 状況 : 色々な種類のゴミが出て来た

▶ 条件分割 : 紙屑は燃えるゴミの袋に、アルミ缶やペットボトルは資源ゴミの袋に..

○ 同じ事を繰り返す事 (A を繰り返せば済む) で分ける

▶ 状況 : 大量のゴミが出て来た

▶ 繰り返し分割 : 取り敢えず一つの袋に入るだけゴミを袋詰めして外に出す、ゴミがなくなるまで、これを繰り返す

○ 問題解決のパターンが知られている (cf. アルゴリズム論)

▶ 状況 : 本が本棚に雑多に押し込まれていて、どこに何があるかが解らない

▶ ソーティング : 書名順になるように並べ替えれば、簡単に探せるようになる

前期の復習 4 : C 言語学習 (1)

□ C 言語 : 「A Programming Language C」

- 「C 言語」は「プログラミング言語」

- ▶ プログラムを記述するための言語

- 「C 言語」で、プログラムを「記述(表現)」できる

- ▶ 「言語」なので、「単語」と「構文(文法)」がある

- ▶ 利用可能な「単語」と適切な「構文」を使わないと正しい「プログラム」にならない

- ▶ (所謂)「コンパイルエラー」は、「正しい文でない」という表示

- !!! 「文法的に正しい文」が「意味のある文」とは限らない

- ▶ 例 : 日本語における「文法的に正しい」が「意味のない」文:「机がガソリンを食べた」

□ C 言語の理解

- 利用可能な「単語(基本命令)」と「構文(文法)」を憶えて、「形式」が整られるようにする

- 単語と構文の意味(機能)を理解し、それから得られる「筋」が通るようにする

前期の復習 5 : C 言語学習 (2)

□ プログラムの最小構成を身に付ける

- 最小構成 : main 関数の作成

- ▶ 例 : Hello, World

□ C 言語の基本命令と構文を身に付ける

- 定義文 (単語に意味や情報を与える記述 [単語定義])

- ▶ 関数定義 : 関数の引数と値の型情報をつけ、その関数を呼出した時に実行される命令を定義し、その関数の呼出しができるようにする

- 基本命令 (単独で文になるもの)

- ▶ 関数呼出し (printf, s_print_int, .. 様々な基本命令 / ライブラリ理解)

- 構文規則 (文から新しい文を作る仕組)

- ▶ ブロック構造 (「{」~「}」: 複数の文を一つの文にまとめる仕組)

- ▶ 制御構造 (if 文 / 再帰 / return 文)