

ソフトウェア概論 A/B

-- 条件分岐 / 繰返し(再帰呼出し) --

数学科 栗野 俊一 / 渡辺 俊一

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く
禁じます

伝言

- 出席パスワード : 20200515
- 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>
- やる気のある方へ
 - 今日の資料は、すでに上っています
 - ▶ どんどん、先に進んでかまいません

前回(2020/05/08)のまとめ

ソフトウェア概論 A/B (2020/05/15)

前回(2020/05/08)のまとめ

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

前回(2020/05/08)の復習

□ 前回(2020/05/08)の内容

○ 引数付き関数

- ▶ 関数の内部の「変化する部分」を、「仮変数」で表す。
- ▶ 「変数」の「値(実際に実行する時に定まる値)」は、関数の実引数に指定する
- ▶ 引数付き関数は、関数の振舞いを「後」から決められる。

○ make と makefile

- ▶ make : 色々なコマンドを実行してくれる (ファイルを「作る」コマンド)
- ▶ makefile : make が参照するレシピ(ファイルの作り方が書いてある)

□ 演習

- 引数付き関数を作る
- make を利用してみる

今回(2020/05/15)の予定と課題

ソフトウェア概論 A/B (2020/05/15)

今回(2020/05/15)の予定と課題

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

今回(2020/05/15)の予定

□ 出席パスワード : 20200515

○ 出席は CST Portal で取りますが、成績には(残念ながら?)無関係です

▶ 単位を取りたいならば、課題を提出しましょう

□ 本日(2020/05/15)の予定

○ C 言語に於ける「文字」と「文字列」の一次理解

○ 条件判定を試みよう

○ 再帰呼び出しを試みよう

□ 本日(2020/05/15)の目標

○ 条件判定をするプログラム : 状態によって振舞を変更する

○ 再帰呼び出しをするプログラム : 同じ事を必要なだけ繰り返す

○ 課題の提出

先週 (2020/05/08) の課題

□ 先週 (2020/05/08) の課題

○ 課題 20200508-01:

- ▶ ファイル名 : 20200508-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 引数で指定された名前に、三度、「こんにちは」をする関数の作成
- ▶ ファイル名 : 20200508-01-QQQQ.c (QQQQ は学生番号)

○ 課題 20200508-02:

- ▶ ファイル名 : 20200508-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 童謡の歌詞を出力するプログラムを作成しなさい
- ▶ 可能な限り引数付きの関数で..
- ▶ 曲は何でもよい

□ 提出するファイル形式

- 全てテキストファイル(C 言語プログラムファイル)
- 提出先は CST Portal II

今週 (2020/05/15) の課題

□ 今週 (2020/05/15) の課題

○ 課題 20200515-01:

- ▶ ファイル名 : 20200515-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 引数付き関数で、if文で条件判断をするプログラムを作成しなさい
- ▶ 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 20200515-02:

- ▶ ファイル名 : 20200515-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 底辺の長さが指定した文字列の二倍の長さ-1の横向のピラミッドを作成するプログラムを作成しなさい
- ▶ 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)
- ▶ 再帰呼び出しを利用する

□ 提出するファイル形式

- 全てテキストファイル(C 言語プログラムファイル)
- 提出先は CST Portal II

文字と文字列

ソフトウェア概論 A/B (2020/05/15)

文字と文字列

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

文字と文字列

□ C 言語における「文字列」

- 二つの「" (ダブルクォーテーション)」で挟んだ「『文字』の列」

- ▶ 例: "abc" は、三つの文字「a」、「b」、「c」から成る、長さ 3 の文字列

□ C 言語における「文字」: 「(ASCII) 文字」

- 基本は、キーボードから「1 キー」で入力されて、画面に表示される「モノ」

- ▶ キーボードで「a」を入力すると、画面に「a」と入る (これが『文字』)

- ASCII Code 表 (1 byte で表現可能な「文字」表)

- ▶ C 言語における「文字」は、ASCII Code 表の 1 byte で表現できる「モノ」

- 「日本語文字」は、1 byte で表現できない (ASCII Code 表にない)

- ▶ 「日本語文字列」は扱えるが、今は説明しない (UTF-8 にする事だけ注意)

エスケープシーケンス

□ キーボードに表記されていない文字の表現

- キーボード上で表現されていない「文字」がある (キーボードは 256 もない)

- ▶ エスケープシーケンスを利用して、色々な「文字」を表現する

□ C 言語におけるエスケープ文字

- 「\ (バックスラッシュ)」が利用される (ubuntu : UTF-8)

- ▶ windows では「¥ (エンマーク)」になる (Windows : JIS X 0201)

- ▶ TeX の時も、同じ話をした...

□ エスケープシーケンス

- エスケープ文字(「\」) + 「文字列」で、一つの「文字」を表現

□ エスケープシーケンスの例

- ▶ \n 「改行コード」

- ▶ \\ 「\ (バックスラッシュ)」 そのもの

- ▶ \" 「"」を表す

「文字列」と計算

□「文字列」の計算

- 「文字列」に、正の整数値を加える事ができる

 - ▶ 結果は、「先頭の文字」が失われて、1文字短くなった「文字列」になる

 - ▶ 例: "abc" + 1 -> "bc"

- 空文字列

 - ▶ 長さ0の文字列 ("")で表現可能)

□「文字列」の比較 (条件式)

- 表現「!strcmp (文字列1, 文字列2)」で、「文字列比較」が可能

 - ▶ 「文字列1」と「文字列2」が「同じ」ならば「真」になる

 - ▶ そうでなければ、「偽」になる

- 空文字列""と比較する事で、文字列の「長さが0かどうか」が解る

条件分岐

ソフトウェア概論 A/B (2020/05/15)

条件分岐

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

条件分岐

□ 引数の内容によって振舞いを「大幅」に変更したい

○ if 文と strcmp 関数を利用して対応できる

▶ strcmp 関数 : 二つの文字列を比較する

○ if (!strcmp (A, B)) { X } else { Y }

▶ A と B が同じなら X を、そうでなければ Y を行う

○ 「else if」を使うと更に複数の命令が選べる

▶ if (C1) { P1 } else if (C2) { P2 } .. else { Pn }

▶ C1 の時 P1、そうでなく C2 の時は P2 .. いずれでもない Pn

○ おまじない (strcmp を利用する場合に使う)

▶ #include <string.h>

○ strncmp (A, B, N);

▶ A と B の先頭の N 文字だけを比較する

▶ !strncmp ("abc", "abz", 3); : 等しくない

▶ !strncmp ("abc", "abz", 2); : 等しい

再帰呼び出し

ソフトウェア概論 A/B (2020/05/15)

再帰呼び出し

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

再帰呼び出し

□ 文字列を順番に見て行く

- 「"abc" + 1」は「"bc"」と同じ振舞いをする

 - ▶ どういう仕組みでこのようになるかは、今回は説明しない

- 次々と 1 を加えれば、どんどん短くなる

 - ▶ 最も短くなった状態に成っているかどうかは、空文字(" ")と比較すれば判定できる

□ 再帰呼び出し

- 普通の関数は、別の関数を呼出す事ができた

 - ▶ 「自分の中」で「自分自身」を呼出す事ができる !! : 再帰呼び出し

□ 再帰呼び出しと数学的帰納法

- 再帰呼び出しは、数学的帰納法の考え方で問題を解く場合に利用できる

 - ▶ 再帰呼び出しが上手く行く事は、数学的帰納法で証明できる (数学との関係)

- 再帰呼び出しをする場合は次の二点が重要 (数学的帰納法と同じ)

 - ▶ 最も小さい場合 (ここでは、文字列が "" の場合) には終了する

 - ▶ そうでない時は、再帰呼び出しするが、その時には文字列を短くする

再帰呼び出しの考え方

□ 目標

- 「全部」をやりたい

- ▶ でも一挙にはできない

□ 対策

- そこで問題を二つに分ける

- ▶ 扱いやすい一部分：これは、そのまま対処してしまう

- ▶ 残り全部：(残り)「全部」なので、再帰呼び出しする

□ 注意点

- 「全部」が空っぽの時に忘れずに処理する

万能性

ソフトウェア概論 A/B (2020/05/15)

万能性

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

三つの基本制御構造と万能性

□ 三つの基本制御構造

○ f を関数, A, B を命令, $p(x)$ を条件とする時、次の三つの基本構造がある

○ [順接] $f() \{ A B \}$

▷ f は A をしてから B をする

○ [分岐] $f(x) \{ \text{if} (p(x)) \{ A \} \text{else} \{ B \} \}$

▷ f は $p(x)$ が成立すれば A 、そうでなければ B をする

○ [繰返] $f(x) \{ \text{if} (p(x)) \{ \} \text{else} \{ A f(x') \} \}$

▷ f は $p(x)$ [x は最小] が成立しない限り A を行う

▷ x' は x から計算される [x' は x より小くなる]

□ 万能性

○ 任意のプログラムこの三つの基本制御構造で構成可能

▷ 「三つの基本制御構造」を憶えれば、後は「組み合わせ」を考えるだけ !!

s_curl の遊び方

ソフトウェア概論 A/B (2020/05/15)

s_curl の遊び方

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

s_curl

- s_curl : web page の内容を取得する
- 準備
 - zip ファイルを 20200515 にダウンロード
 - zip ファイルを展開
 - ubuntu を起動
 - cd c/20200515/s_curl
 - ./install.sh
 - sudo apt install make
- 遊び方
 - make
 - make test

おしまい

ソフトウェア概論 A/B (2020/05/15)

おしまい

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます