

ソフトウェア概論 A/B

-- 条件判断と繰り返し --

数学科 栗野 俊一

2012/05/18 ソフトウェア概

伝言

私語は慎むように !!

□ 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>

□ 廊下側の一列は遅刻者専用です(早く来た人は座らない)

□ 講義開始前にすませておくこと

- PC の電源を入れる / ネットワークに接続しておくこと

▶ 今日の資料に目を通しておくこと

□ 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

前回の復習

□ 内容

○ 分割コンパイル

- ▶ プログラムを複数のファイルに分割して作成し個々にコンパイル
- ▶ 後で、一つの実行ファイルにまとめる (リンクする)

○ make と makefile

- ▶ **make** : 分割コンパイルの作業を軽減するためのツール
- ▶ **makefile** : これに作業手順を記述すると **make** がそれを見て自動的に作業

□ 演習

○ 引数付関数の作成

- ▶ 引数付関数を作ってみる

○ PC で Turtle Graphics (亀プログラム) をしてみよう

お知らせ

□ 本日の予定

- 引数付き関数を作ってみよう
- 条件判定を試してみよう
- 再帰呼び出しを試してみよう

□ 本日の目標

○ 文字列と文字

- ▶ putchar / strcmp

○ プログラムの制御構文

- ▶ if 文
- ▶ 再帰呼び出し

○ 演習

- ▶ 条件判定をするプログラム
- ▶ 再帰呼び出しをするプログラム
- ▶ 課題の提出

前回の課題 (2012/05/11)

□ 前回 (2012/05/11) の課題

○ 課題 1:

- ▶ ファイル名 : 20120511-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 漢字の「回」という文字(にみえる..) 絵を Turtle Graphics で書なさい
- ▶ 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)

□ 今回に引き継ぐ課題 (2012/05/11) の課題

○ 課題 2:

- ▶ ファイル名 : 20120511-2-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 3:

- ▶ ファイル名 : 20120511-3-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
- ▶ 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

本日の課題 (2012/05/18)

□ 本日 (2012/05/18) の課題 (CST Portal のみ)

○ 課題 1:

- ▶ ファイル名 : 20120511-1-XXXX.c (XXXX は学生番号)
- ▶ 内容 : 渦巻模様 (10 辺) の絵を Turtle Graphics で書なさい
- ▶ 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

引数付き関数の作り方 (再録)

□ 引数とは

○ 関数に与える事により、関数にその引数に対応した挙動をさせるもの

- ▶ 引数付き関数の定義：引数の値によって挙動が変わる
- ▶ 引数付き関数の利用：指定したい挙動をさせるための値を指定する
- ▶ cf. 三角関数：引数の角度によって異なる値を返す

□ 引数付き関数の作り方

○ 似ている二つ関数を一つの引数付き関数にまとめる

- ▶ 関数の本体の部分を、同じ部分と違う部分に分ける
- ▶ 違う部分は「変数」に置き換えて、一つの関数定義にまとめる
- ▶ 関数の仮引数の所に、「変数」を追加する
- ▶ 呼出す側は、実引数に、「違っていた部分の内容」を指定する

文字列と文字

□ C 言語での「文字」の扱い

- 文字 : 文字をシングルクォート('')ではさんだもの

 - ▷ cf. 'A', 'a', '1'

 - ▷ 「一文字」と対応している「表現」

- 文字の出力 : **putchar** 関数を利用する

 - ▷ cf. putchar ('A');

- 文字列 : 文字列をダブルクォート("")ではさんだもの

 - ▷ cf. "ABC", "123", "" (空文字列)

 - ▷ 「文字の並び(0個以上)」と対応している「表現」

 - ▷ <<注意>> : 全角(日本語)は、一文字で、二文字分になる

- 文字列の出力 : **printf** 関数を利用する

 - ▷ cf. printf ("abc");

文字列の構造

- 文字列は、文字の並び + 文字の終りからなる
 - "ABC" == { 'A', 'B', 'C', '\0' }
 - ▷ 長さ 3 (n) の文字列は、4 (n+1) つの部分からなる
 - ▷ '\0' を EOS (End Of String) と呼ぶ
 - k 番目の文字の取り出し方 : [k] をつける (k は 0 から始まる事に注意)
 - ▷ cf. "ABC"[0] == 'A', "ABC"[3] == '\0', "ABC"[9] == ? (未定義)
 - 先頭の文字を取り出すには * を付けてもよい
 - ▷ cf. *"ABC" == 'A'
 - 先頭の文字を取り除いた残りを取り出すには 1 を加えればよい
 - ▷ cf. "ABC" + 1 == "BC"
- 文字列の判定 : strcmp を使うと、「二つの文字列が同じなら偽」になる
 - cf. strcmp ("ABC", "ABC") ==> 偽, strcmp ("ABC", "XYZ") ==> 真
 - 文字列が「空文字列(長さが 0 / 文字を含まない文字列)」は先頭が EOS かどうかで判定できる
 - ▷ (*"" == '\0') ==> 真

条件分岐

□ 条件分岐 (if 文)

- 構文「if (条件式) { X } else { Y }」

- ▶ 条件式が真 (0 以外) なら「X」を行う
- ▶ そうでない(偽すなわち 0)なら「Y」を行う
- ▶ X と Y のどちらか一方しか行わない事がポイント

□ 引数の内容によって振舞いを「大幅」に変更したい

- if 文と 論理式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

- ▶ #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

□ 文字列を順番にみてゆく

- 「"abc" + 1」は「"bc"」と同じ振舞いをする

 - ▶ どういう仕組みかは今回は説明しない

- 次々と 1 を加えれば、どんどん短かくなる

 - ▶ 最も短くなったかは、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

 - ▶ 自分自身も呼出す事ができる !! : 再帰呼び出し

□ 再帰呼び出しと帰納法

- 再帰呼び出しは、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、帰納法で証明できる

- 再帰呼び出しをする場合は次の二点が重要 (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