ソフトウェア概論 A/B

-- 制御構造とプログラム / 文字と文字列 / 型宣言 --

数学科 栗野 俊一 / 渡辺 俊一

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)
- 講義開始前に済ませておく事
  - PC の電源を入れる
  - ネットワークに接続しておく事
  - 今日の資料に目を通しておく事
- 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(2013/05/17)の復習

- 前回(2013/05/17)の内容
  - 作成方法
    - 再帰呼出：関数の定義の中で、自分自身を呼び出す
  - 表現方法
    - 再帰呼出、固有の表現はない (単なる関数呼出と同じ ..)
    - ただ、典型的なパターンがあるので、それを取り敢えず、憶える
  - 操作方法
    - なし (新規に追加した操作はない)

お知らせ

- 本日(2013/05/24)の予定
  - 復習
    - 再帰呼び出しをしてみよう
  - 新規
    - 文字列と文字
    - データ型
    - 亀プログラム再
- 本日(2013/05/24)の目標
  - プログラムの三つの基本構造を把握する
    - 順接 / 条件分岐 / 繰返し
  - 演習
    - 文字の出力
    - 課題の提出

前回 (2013/05/17) の課題

- 前回 (2013/05/17) の課題
  - 課題 3:
    - ファイル名 : 20130510-3-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
    - 注意 : これは前回(2013/05/17)の課題にとしているが課題名は先々週の日付(20130501)のまま出す

本日 (2013/05/24) の課題

□ 本日 (2013/05/24) の課題

○ 課題 1:

- ▹ ファイル名 : 20130517-1-QQQQ.c (QQQQ 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
- ▹ 注意 : これは今回(2013/05/24)の課題にとするが課題名は先週の日付(20130517)のまま出す

○ 課題 1:

- ▹ ファイル名 : 20130524-1-QQQQ.c (QQQQ は学生番号)
- ▹ 内容 : 自分の名前のローマ字を putchar を用いて出力する

○ 課題 2:

- ▹ ファイル名 : 20130524-2-QQQQ.c (QQQQ は学生番号)
- ▹ 内容 :

条件分岐 (復習)

- 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 .. いずれでもないと 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 (復習)

- 文字列を順番にみて行く
  - 「"abc" + 1」は「"bc"」と同じ振舞いをする
    - 何の様な仕組かは *今回は* 説明しない
  - 次々と 1 を加えれば、文字列はどんどん短かくなる
    - 最も短かくなっているかは、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、別の関数を呼び出す事ができた
    - 「自分の中」で「自分自身」を呼び出す事ができる !! : 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 (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 (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」が空っぽの時を忘れずに処理する

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) { if (p(x)) { A } 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) { if (p(x)) { A f(x') } else {} }
    - f は p(x) が成立する限り A を行う
    - x' は x から計算される
- 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 !!

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

亀プログラム (return)

- 複雑な亀プログラムを書いてみる
  - 複雑な図形が、簡単なプログラム描画できる
- 再帰図形
  - 再帰呼び出しを利用すると再帰的な図形が記述できる
    - ペアノ曲線