

11年度C言語講座

2011/06/02 第4回

講師: フェイス、ork、蟹男

関数

- 命令を一括りにしたもの
 - ……と考えるのが手っ取り早い。
- 関数は『引数』と『返り値』を持つ
 - 引数: 関数に予め与えておく数値
基本的に、**関数内部から関数外部の変数はみえない**ので、こちらから必要なデータを与えてあげないといけない。
 - 返り値: 関数から帰って来る値
同じように、**関数を抜けると関数内部で宣言した変数は消えてしまう**ので、結果を関数外部にフィードバックするのに必要である。
 - 返り値の型は、関数の宣言時に指定する。

ポインタ

- C言語の勉強でぶち当たる壁
 - ……らしいです。
理解しにくい概念なので、「なんとなくこんな感じ」と自己解釈できたらグッド。
- 変数の1つで、内部に『アドレス』を保持する
 - というのが定義。
 - アドレスについて後述。

アドレスと変数

- アドレスとは
 - コンピュータのメモリに割り振られた『address』
記憶領域の境を識別する番号。
- 変数との関係
 - 宣言された変数は、必ずメモリ上のどこかに配置される
→ 変数とアドレスは1対1対応する。
 - 変数を『家』としたとき、アドレスはそのまま『番地』。
“&変数名”でその変数のアドレスを表す。
- コード上では名前で識別されている変数も、内部的にはアドレスで識別されている。

何故swap関数は動かなかったか

- 変数名の同じ、違う変数に対して操作を実行したから
 - 『一丁目の鈴木さん』と『三丁目の鈴木さん』は別人という話。
- 解決策
 - ポインタ(アドレス)を用いて、関数内部に外部から別の変数を認識させる。
 - 上の話で行くと、三丁目の人に一丁目の鈴木さんの住所を教える要領。

ポインタの用法

- “変数型 *変数名” という宣言で、ある変数型に対するポインタが生成される。
 - 例) `int a; int *pa; pa = &a;`
 - int型のポインタはint型変数のアドレスしか保持できない。
- “*変数名” でそのポインタの保持するアドレスにある変数の中身を参照できる。
 - 例) `*pa = 8;` //aに8が代入される。

(余談) 配列とポインタ

- 配列の要素は、メモリ上に隙間なく並んでいる。
- 配列名は『配列の要素の先頭のポインタ』である。
 - 配列を関数に渡して内部で値を変更したい時は、配列名をポインタに渡せばいい(サンプル参照)。
- ポインタは可減算できる
 - $pa \pm 1$ でそれぞれ1つ後、1つ前のアドレスを示すようになる。
 - 1つズれるの『1つ』というのは変数型に依存する。
 - 配列名に用いれば任意の番号の配列要素を取り出せる。

課題1

- 講義内容を参考にして、先のswap関数を完成させよ。
 - どこにポインタを用いればいいのかを考える。
- 二つの数字を入力し、演算方法を指定すると二つの数字に対する演算結果を出力するプログラムを作成せよ。

但し、演算部分を関数化すること。

- ヒント: 3つの引数を持ち、
演算結果を返す関数を作成する。