C言語講座
0110回

色々詰め込むよ!!!!!
(°∀°)つ●ウンコクウカ?⊂(°∀°)センキュウー♥

構造体!

- 自分で新しい1型を自分で作れる!!
- データ管理を楽に出来る!!!

- いろいろな種類の互いに関連するデータをまとめて，1つのかたまりにしたもの
- 例…人データの構造体

人データ

変態度（年齢）
属性（身長）
性癖（体重）
髪型（性別）

その前に…

- Typedefを理解しよう

Typedefとは

- 型の名前を自由に変えることが出来る!!
- 例えば…

```
typedef  int  unko;
unko  kato  =0;
```

と「unko」型をintと同じように使える

でだ…構造体

Typedef　　struct{

メンバー変数

いくつでも入れられる

} 構造体の名前 ;

正式は違うけど…

さっきのやつで例

```
typedef struct {
  int age;              int hentai;
  double height;        char zokusei[20];
  double weight;        char seheki[20];
  char sex;             char kamigata[20];
} person;
```

- Person p;
- p.age = 19;
- p.height =174.5;
- p.weight=59;
- p.sex ="male";
- p.hentai =30;
- p.zokusei="のんべー";
- P.seheki
- p.kamigata ="普通";

- Printf("髪型は%s",p.kamigata);

構造体終わり

●問題!!

- トランプの1枚の情報を持つ構造体を作ってみよう!!

⊂(^~^⊃。Д。)⊃　　⊂(。Д。⊂~^)⊃

標準ライブラリ関数

- コンパイラメーカがよく使う機能をオブジェクトライブラリとして提供してくれるもの
- 使うときは#include<ホニャララ>

- 今までよく使っていたstdio.hは入出力用の関数

例

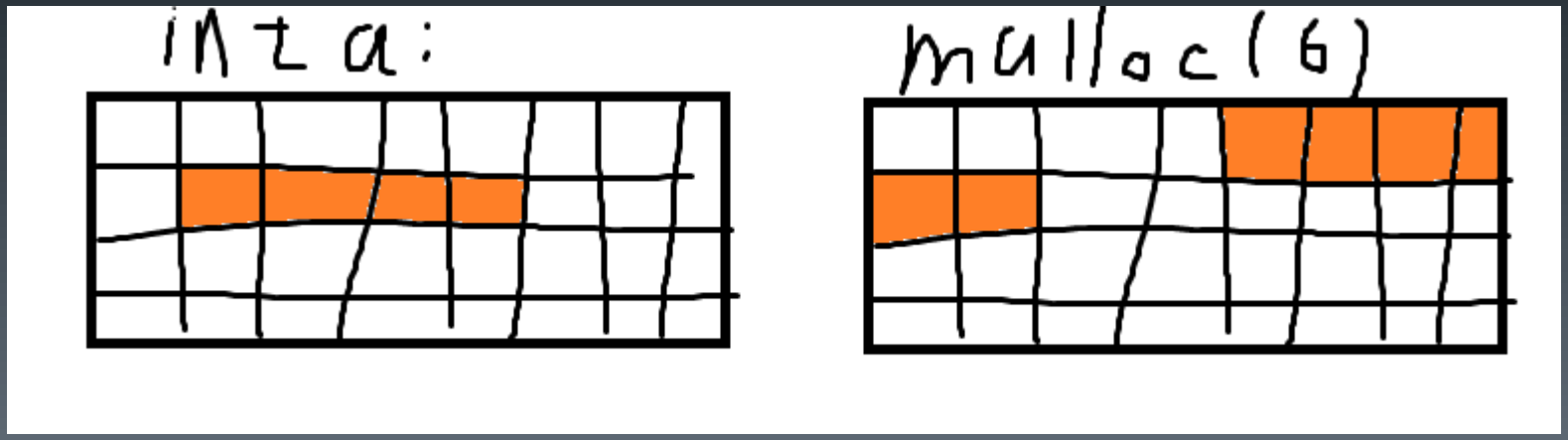
- 入出力〈stdio.h〉
- 汎用〈stdlib.h〉
- 文字処理〈ctype.h〉
- 文字列処理〈string.h〉
- 数学関数〈math.h〉
- 時間〈time.h〉
- http://homepage3.nifty.com/mmgames/c_guide/r_lib.html

いくつか使ってみよう

mallocとcalloc <stdlib.h>を使用します

- メモリを動的に確保するもの

- どれぐらい使うかわからない時に必要な用だけ確保する



void *malloc(size_t n)

- 確保したメモリのアドレス malloc(確保するメモリサイズ)

- 使用例

```
int *p;
p = (int*)malloc(sizeof(int) * 10);
```

sizeof()演算子は各データ型が何バイトか知ることができる。

Warning!! Warning!! Warning!! Warning!! Warning!! Warning!!

mallocで確保したメモリ領域は
void free(void *p);
を用いて開放してあげましょう。
関数内で定義した変数みたいに
抜けても開放されることは
ありません

Warning!! Warning!! Warning!! Warning!! Warning!! Warning!!

ファイルの分割

- 1つのファイルにすべて書くのは膨大になってくると管理が辛い！
- 複数人で書けない！

- あとで使える

includeについて…

プログラムの一番始めに
書いていたおまじない…

その具体的な意味は…

- .hファイルに書かれている内容を
  include<～.h>と置き換える
  ヽ(°皿°)ヽﾍﾟﾀｰ!!

自作ヘッダーファイルをincludeするときは
include< >ではなくinclude" "です

- ヘッダーファイルに書く一般的な内容は

- 別ファイルに書かれている関数のプロトタイプ宣言
- グローバル変数
- 構造体

あとはリア充に聞いて…

やってみよう!!

- さっき作ったトランプ構造体をヘッダーファイルに記述し、分割してみよう

余力があったらやってみよう

- 1.トランプ型を52個mallocする
- 2.それらをすべて別のカードになるように初期化
  (for*2でできる？)
- 3.randomで2つ選び入れ替える(シャッフル)作業を無数回
- 4.上から5枚表示する!!

ライブラリ

- 標準じゃないやつ

- 1から全部作るのは面倒
- 既存のものを使うのが賢い！
- あるプラットフォーム上で動かすものはそれを使わないと動かない！？

- 黒い画面はもう飽きたーとか
- もっと高度なことしたいって人用

Windows API

- Win32といわれるもの。
- Windowsアプリケーションを CやC++で作りたいならこれ
- ウィンドウの生成やDirectXでの描写、ネットワーク通信、スレッド…などたくさんの機能をもつ
- 結構、難易度は高め？

DXライブラリ

- 前で説明したwindowsAPIのラッパーライブラリ
- DirectXやWindows関連のプログラムを使い易くまとめた形で利用できる
- あくまでラッパーなのでwindowsAPIより自由度は低い
- 難易度は低め？

OpenCV

- 画像処理に特化したライブラリ
- 構造解析やモーション解析、パターン認識(物体検出)、機械学習(顔認識とか)
- 難易度は中くらい!?

OpenAL

- オーディオ系ライブラリ

別にCやC++にこだわることはない!!!

言語は他にも沢山ある

C　C++　C#　Objective-C

Visual Basic.NET　D　JAVA

PHP　Perl　Python　Ruby

JavaScript　Scala

ActionScript…

- それら1つ1つに利点や得意分野があり、それらの言語にもライブラリが存在する。
- Androidアプリケーションがやりたいならjava
- iPhoneアプリケーションがやりたいならObjective-C
- Windowアプリケーションをやりたいならc#などなど…

上達するには？

- 本で基礎知識を身につける。
- 自力でプログラムを組む力を身につける。

- あくまで学ぶことなので自分との戦い！
  けどわからなかったら聞ける
- まずは簡単な物を作ろう。そこから順に難しい！ものを作る。
- アルゴリズム的問題を解いてみる
- 毎日、書こう