◆必要なサイズのメモリを動的確保

void \*malloc(size\_t size);

確保したいメモリサイズ

↑

確保されたメモリのアドレス

（型不定 ）

（格納したいデータ型に ）

（キャストしれ使用 ）

◆確保できたメモリには

配列と同じようにアクセスできる

PLAYER \*pPlayer;

pPlayer = (PLAYER\*)malloc(sizeof(int) \* 100);

pPlayer[5].nLife = 50;

◆不要になったメモリの解放

void free(void \*ptr);

mallocで確保されたメモリのアドレス

確保したメモリは使い終わったら必ず解放すること

解放されたメモリはOSに返され、他のプログラムが再使用できる

忘れるとバグる（メモリリーク）となる。

◆メモリの必要サイズが変わった場合

古いメモリの解放（free）

新しいメモリの確保(malloc)

free(pInt);

pInt = (int\*)malloc(sizeof(int) \* 200);

解放を忘れると

新しいメモリアドレスで上書き→古いメモリは解放できなくなる

◆mallocとfreeの仕様

#include <stdlib.h>

＜課題内容＞

データ入力時（人数入力）

データロード時（人数をファイルから読み込み）に

必要人数に応じて

メモリ管理を行うように変更せよ