

プログラミング基礎

2023年度1Q 火曜日 3, 4 時限(10:45~12:25)
金曜日 1~4 時限(8:50~12:25)

工学院 情報通信系

中山実, 渡辺義浩

伊藤泉, 小杉哲

TA: 小泊大輝, 千脇彰悟

Ex46(2023/05/23)

人と計算機がジャンケンをするプログラムを作成し，40回対戦して，その結果（勝ち，負け，引き分け）を出力する．

計算機は，人が出す手を入力する前に出す手を決めなければならないが，それ以外の制約はなく，なるべく強い（と思われる）戦略を自由に考えること．

次のページのコードを参考に，入出力のやりかたやコードの細かい点などは自分の好みで変えて構わない．

- ・ 5月30日(火)10:45までコードを提出する．
- ・ 自分のプログラムの戦略，5月30日(火)の対戦結果，それを踏まえた反省点や感想などを6月2日(金)までにレポートとして提出する．改良したコードを付け加えてもいい．本講義全体についての感想，意見などがあればそれも記載してください．

Ex46(2023/05/23)

対戦型ジャンケンプログラム例 (一部)

```
#include <stdio.h>
#include <stdlib.h>
#define NUM 40

int main(void) {
    int man[NUM]={}, pc[NUM]={}; //0で初期化
    int i,judge;
    int results[3]={0,0,0}; // [0]引き分け, [1]人の勝ち, [2]PCの勝ち, の回数
    char *janken[3]={"グー", "チョキ", "パー"};
    char *winlose[3]={"引き分け", "人の勝ち", "PCの勝ち"};

    for(i=0; i<NUM; i++){
        printf("%d 1:グー 2:チョキ 3:パー¥n", i+1);

        //pc[i]=?; 1:グー 2:チョキ 3:パー

        while ( (man[i]<1) || (man[i]>3) ) scanf("%d", &man[i]);
        printf("人:%s, PC:%s ¥n ", janken[man[i]-1], janken[pc[i]-1]);
        judge=(pc[i]-man[i]+3)%3;
        printf("%s¥n", winlose[judge]);
        results[judge]++;
        printf("合計%d戦中 PC:%d勝%d負%d分¥n¥n",i+1,results[2],results[1],results[0]);
    }
    return 0;
}
```

Ex46(2023/05/23)対戦型ジャンケン

戦略の例；

```
//常にグーを出す. # この戦略では評価なし.  
pc[i]=0;
```

```
//ランダムにグー(1), チョキ(2), パー(3)を出す. # この戦略も評価なし.  
pc[i]=rand()%3+1;
```

```
//相手が出す手の傾向を読む
```

```
//過去の手のデータを右図のように  
2次元, 3次元配列として表現すると  
戦略を具体化しやすい.
```

i-1 番目の
PCの手

i 番目の人の手

	グー	チョキ	パー
グー	3	10	3
チョキ	2	0	4
パー	6	1	2

注意事項など：

- ・後だし禁止.
- ・対戦の時の忖度禁止.
- ・事前にデータを蓄えるのはOK.
- ・相手と自分の手の履歴を記録するのはOK.
- ・入出カインターフェースを工夫するのはOK.
- ・アルゴリズムの根拠が必要.
- ・毎回の勝ち負けがわかるように出力する.
- ・勝率 4 割越えは素晴らしい.

Ex46(2023/05/23)対戦型ジャンケン

- ・ 最終回（5/30）では，1グループ5～6名（全部で8～9グループ）で，それぞれのプログラムに対して他者数名が挑戦し，合計勝率（勝／回数）が最も高い者を勝者とする [2023年度は時間の関係で3名程度とする]
- ・ 各パソコンでプログラムを実行し，他の人が対戦できるように準備する.
- ・ 各グループで，司会と書記を割り当てるので，司会は全体の進行，書記は最終結果をスタッフに渡す.
- ・ グループの少なくとも2人がソースコードをチェックして不正がないか確認する.
- ・ 全てのグループの対戦が終わったら，各グループの勝者は，自分のアルゴリズムについてみんなに説明する.
- ・ 各グループで勝者以外の1名を選び，その人はグループ内のアルゴリズムについて総括して説明する.

	対戦者1(人)	対戦者2(人)	対戦者3(人)	対戦者4(人)	対戦者5(人)	対戦者6(人)	合計
Prog.1		勝 負 分	勝 負 分	勝 負 分	勝 負 分	勝 負 分	勝 負 分
Prog.2	勝 負 分		勝 負 分	勝 負 分	勝 負 分	勝 負 分	勝 負 分
Prog.3	勝 負 分	勝 負 分		勝 負 分	勝 負 分	勝 負 分	勝 負 分
Prog.4	勝 負 分	勝 負 分	勝 負 分		勝 負 分	勝 負 分	勝 負 分
Prog.5	勝 負 分	勝 負 分	勝 負 分	勝 負 分		勝 負 分	勝 負 分
Prog.6	勝 負 分	勝 負 分	勝 負 分	勝 負 分	勝 負 分		勝 負 分