今後の予定

- 今日のレポート
  - 演習問題5，6，7，8を解いて，作ったプログラムを提出
  - ISTUを使って提出してください
  - 締め切り：次回講義時間の終了時まで
- 今日は授業評価をしてもらいます
  - 退出前に，後ろの箱に入れてください
- 次回
  - 説明および新しいレポートなし
  - 出欠は取りません
  - 今回のレポートを未提出の場合，締め切りまでに忘れずに提出してください

演習問題その3:注意事項

問題3-2:
xとyの足し算の結果だけでなく，引き算の結果も表示するように変更せよ.

ダメな実行例

```
x= 6        （キーボードから6を入力）
y = 2       （キーボードから2を入力）
sum= 8    （足し算の結果＝8:これは正しい）
difference=-4      （引き算の結果＝-4 ???）
```

引き算の順番に
注意！

演習問題その4：注意事項

問題4-1：
変数 x と y の値をキーボードから入力して，
x が y より大きかったら「8 is larger than 2」のように表示し，
x が y 以下だったら「1 is not larger than 5」のように
表示するプログラムを作成せよ．

ダメな実行例
x= 2　　（キーボードから2を入力）
y = 6　　（キーボードから6を入力）
2 is not larger than y

x, y ともに数字を出力する！

printf(“aaaa %d bbb %d”, x, y);
の形で printf を使う

プログラムその5：最大値を求める

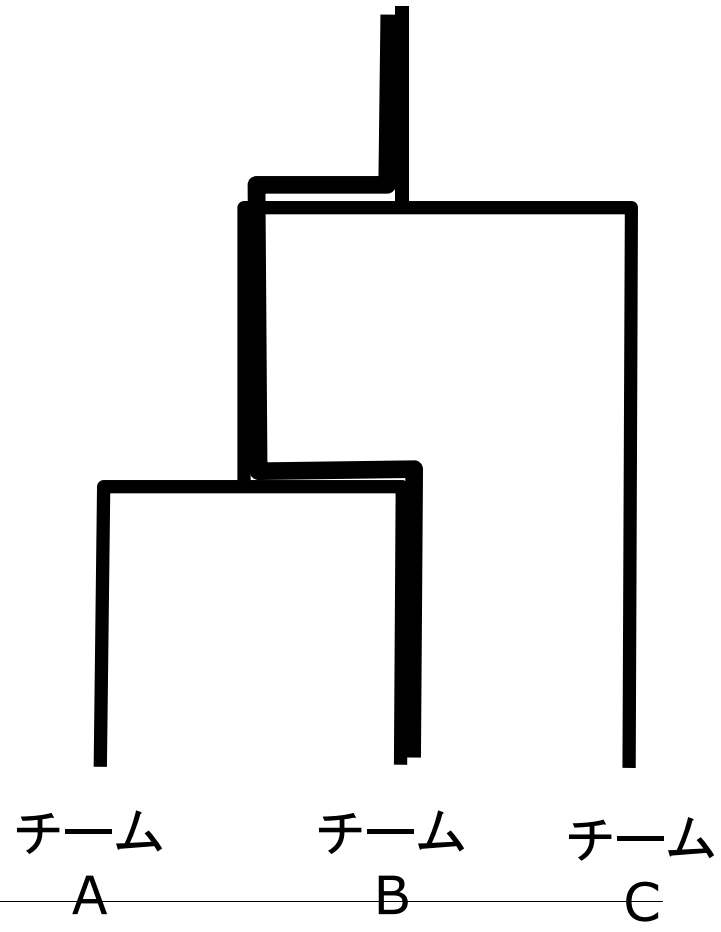
if … else … を使って，3つの値の中の最大値を求める

```
#include <stdio.h>

main()
{
    int a, b, c, max;
    scanf(“%d”, & a);
    scanf(“%d”, & b);
    scanf(“%d”, & c);
    max = a;
    if (b > max) {
        max = b;
    }
    if (c > max) {
        max = c;
    }
    printf(“maximum = %d¥n”, max);
}
```

プログラムのアイディア

- イメージ：スポーツの勝ち抜き戦（もっとも強いチームを決める）
- 最初の暫定王者：チームA
- 暫定王者（チームA）とチームBの対戦
  ➔勝った方が新たな暫定王者
- 新たな暫定王者（AまたはB）とチームCの対戦
  ➔勝った方が新たな暫定王者
- 全部のチームが対戦に参加したので，現時点での暫定王者が，全体のチャンピオン



プログラムの説明

```
#include <stdio.h>

main()
{
    int a, b, c, max;
    scanf(“%d”, & a);
    scanf(“%d”, & b);
    scanf(“%d”, & c);
    max = a;
```

4つの変数 a, b, c, max を準備

数値をキーボードから入力し，
変数 a に入れる

暫定チャンピオン（変数max）を
a にする

プログラムの説明

if … else …の else 以降は（不要ならば）省略可能

```
    if (b > max) {
        max = b;
    }
    if (c > max) {
        max = c;
    }
    printf("maximum = %d¥n", max);
}
```

b と暫定チャンピオンを比較
b の値が暫定チャンピオンより大きい
➔暫定チャンピオンを b にする
（b が暫定王者以下ならば，
暫定王者は変わらず）

c の値が暫定チャンピオンより大きい
➔暫定チャンピオンを c にする

最終的な暫定王者を
最大値として表示

最大値を決めるために必要な比較の回数

- 3チームの中で最強を決めたい
  （ただし，強いチームが必ず勝つ）
  ➔最低2試合は必要（試合＝数値の比較）
  （1試合では試合に参加出来ないチームあり）
- このプログラムでは2試合で最強を決めている
  ➔計算効率がよい（無駄な試合がない）

コンピュータとプログラムと計算時間

一般に,

- プログラムの計算時間は演算回数(四則演算, 比較など)に比例
- 演算回数が少ない方が早く終了

  ➔計算時間の点で良いプログラム
- 複雑な問題の場合, プログラムの善し悪しで計算時間が大きく変わる
  - 例: 5n 回の演算で動くプログラム vs 2^n 回の演算で動くプログラム
  - n = 50 のとき, 1億分の1秒 vs 1.3日
  - n = 100 のとき, 1億分の2秒 vs 500億年

1秒に100億回の演算

- コンピュータの性能が良くても, プログラムが駄目な場合, 複雑な問題はいつまで待っても解けない

➔良いプログラムのアイディアを考える研究の必要性

演習問題その5

問題5:
4つの数字をキーボードから入力し，その中で一番大きい数字を求め，表示するプログラムを作成しなさい.

- プログラム名は ex5.c としてください

プログラムその6：繰り返し

- 整数1から10の合計を求めたい

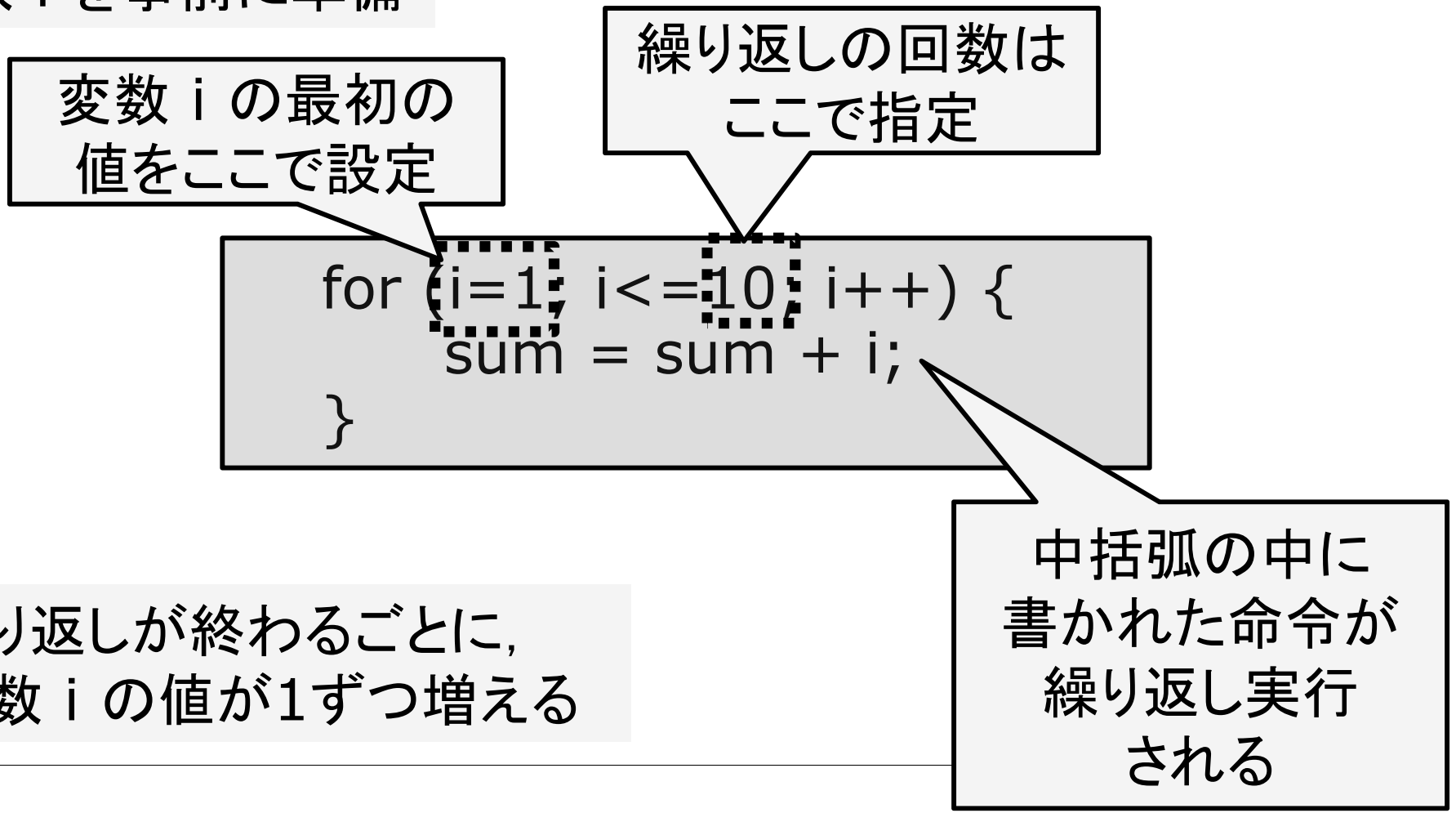
➔繰り返しの命令を使えば簡単

```
#include <stdio.h>

main()
{
    int i, sum;
    sum = 0;
    for (i=1; i<=10; i++) {
        sum = sum + i;
    }
    printf("sum = %d¥n", sum);
}
```

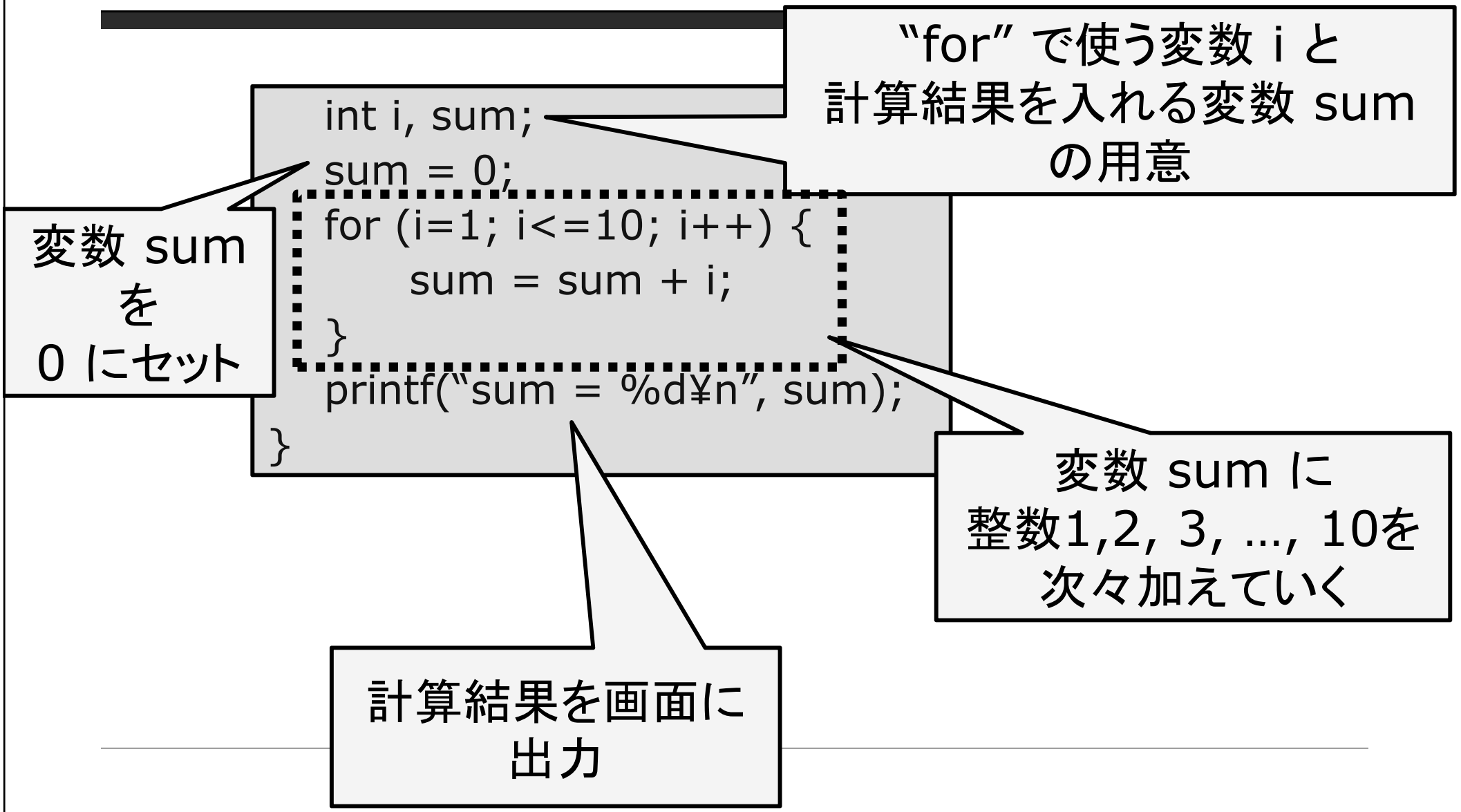
繰り返しの命令 “for” の説明

変数 i を事前に準備



繰り返しが終わるごとに,
変数 i の値が1ずつ増える

プログラムの説明（その１）



プログラムの説明（その2）

```
sum = 0;
for (i=1; i<=10; i++) {
      sum = sum + i;
}
```

1回目の繰り返し
- 変数 i を 1 にセット
- sum に i を足して，sumに再び入れる
（➔sum は0から1になる）

2回目の繰り返し
- 変数 i は自動的に2に変わる
- sum に i を足して，sumに再び入れる
（➔sum は1から3になる）

3回目の繰り返し
- 変数 i は自動的に3に変わる
- sum に i を足して，sumに再び入れる
（➔sum は3から6になる）

これを10回繰り返して終了
➔sum は55になる

演習問題その6

問題6-1:
プログラムを修正して，正の整数 k を入力したら，
1から k までの合計を計算できるようにせよ.
問題6-2:
プログラムをさらに修正して，正の整数 k を入力したら，
1から k までのかけ算1×2×…×k(k の階乗)を
計算できるようにせよ.
※6-1, 6-2 はまとめて ex6.c というプログラム名で作成，提出

演習問題その7

問題7:
10個の数字をキーボードから入力し，その合計を求め，
表示するプログラムを作成しなさい.
※ex7.c というプログラム名で作成，提出

ダメなプログラム例：scanf を10回使うのはダメ
for を必ず使うこと

```
scanf(“%d”, & a);
scanf(“%d”, & b);
…
scanf(“%d”, & i);
scanf(“%d”, & j);
printf(“%d ¥n”, a+b+c+d+e+f+g+h+i);
```

プログラムその8:ランキング決定

if … else … を使って，3つの値のランキングを出力

```
#include <stdio.h>

main()
{
    int a, b, c, La, Lb, Lc;
    scanf("%d", & a);
    scanf("%d", & b);
    scanf("%d", & c);
    La = 0; Lb = 0; Lc = 0;
   if (a > b) {
         Lb = Lb + 1;
    }  else {
         La = La + 1;
    }
    if (a > c) {
         Lc = Lc + 1;
    }  else {
         La = La + 1;
    }
    if (b > c) {
         Lc = Lc + 1;
    }  else {
         Lb = Lb + 1;
    }
    printf("rank of A = %d¥n", La+1);
    printf("rank of B = %d¥n", Lb+1);
    printf("rank of C = %d¥n", Lc+1);
}
```

授業のWebページよりダウンロード可能です

プログラムのアイディア

- 3チームでの総当たり戦（A対B，A対C,B対C）
  - ただし同じ強さのチームは無しとする➔引き分け無し
- 各チームの負け数を数える
  - ➔0敗，1敗，2敗のチームが一つずつ現れる
- 0敗のチーム➔1番目に強いチーム
- 1敗のチーム➔2番目に強いチーム
- 2敗のチーム➔3番目に強いチーム

	A	B	C	負け数	ランク
A		×	○	1	2
B	○		○	0	1
C	×	×		2	3

プログラムの説明

```
#include <stdio.h>

main()
{
    int a, b, c, La, Lb, Lc;
    scanf("%d", & a);
    scanf("%d", & b);
    scanf("%d", & c);
    La = 0; Lb = 0; Lc = 0;
   if (a > b) {
         Lb = Lb + 1;
    }  else {
         La = La + 1;
    }
```

La, Lb, Lc はそれぞれ，
A, B, Cチームの
負け数を数えるための変数

最初は負け数を0にセット

AとBの対戦
Bが負けたら，負け数Lbを1増やす
Aが負けたら，負け数Laを1増やす

プログラムの説明

```
    if (a > c) {
        Lc = Lc + 1;
    }  else {
        La = La + 1;
    }
    if (b > c) {
        Lc = Lc + 1;
    }  else {
        Lb = Lb + 1;
    }
    printf("rank of A = %d¥n", La+1);
    printf("rank of B = %d¥n", Lb+1);
    printf("rank of C = %d¥n", Lc+1);
}
```

AとCの対戦
Cが負けたら，負け数Lcを1増やす
Aが負けたら，負け数Laを1増やす

BとCの対戦
Cが負けたら，負け数Lcを1増やす
Bが負けたら，負け数Lbを1増やす

負け数+1が，そのチームのランクになる

ランキングを決めるために必要な比較の回数

- 3チームの中でランキングを決めたい
  （ただし，強いチームが必ず勝つ）
  - 2試合では不十分（同じチームが2勝した場合，残り2チームの優劣が付けられない．例：A>B, A>C）
  - 3試合あれば総当たり戦になるので，大丈夫
    ➔最低3試合は必要
- このプログラムでは3試合で最強を決めている
  ➔計算効率がよい（無駄な試合がない）

演習問題その8

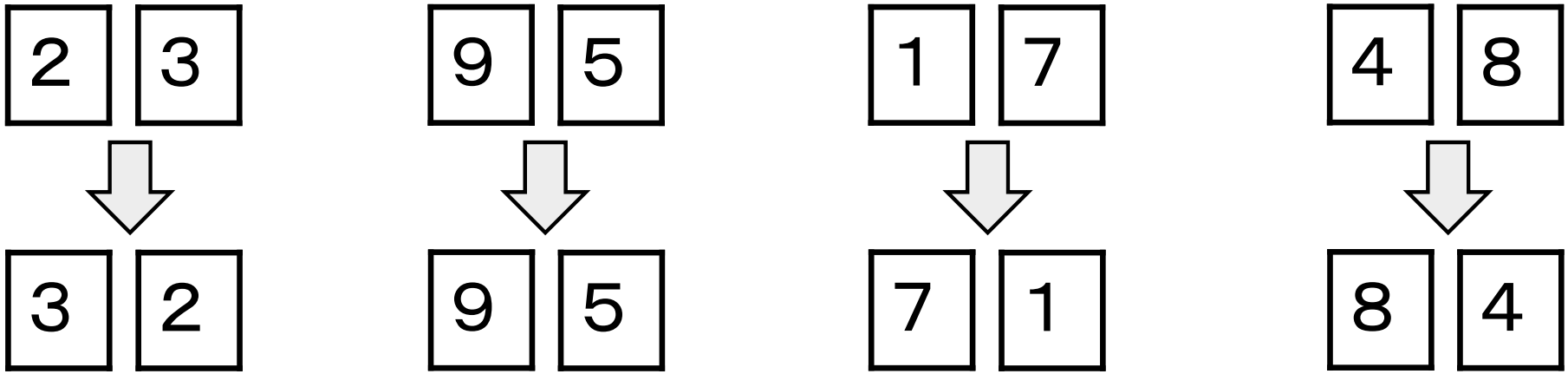
問題8:
4つの数字をキーボードから入力し，4つの値のランキングを求め，表示するプログラムを作成しなさい.
（3チームの場合と同様に，総当たり戦のアイディアを使う）

- プログラム名は ex8.c としてください

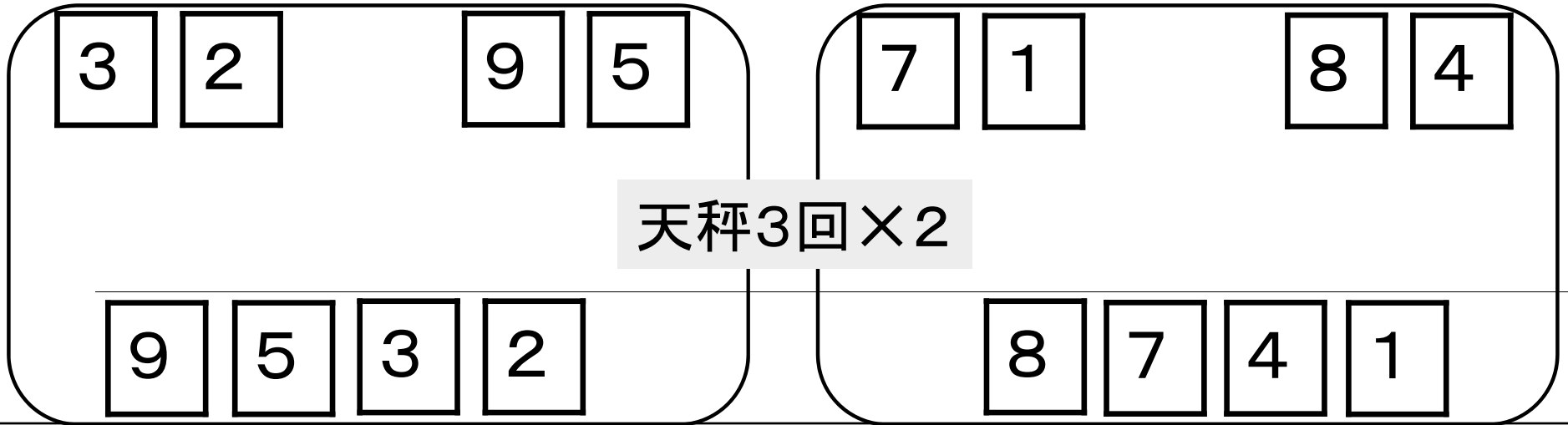
発展問題:4チームのランキングを決めるとき，
- 総当たり戦（合計6試合）は必ず必要か？
- 5試合では決定できないか？

宝石の問題の解答例

（1）2個ずつのグループに分け，大きい順に並べる（天秤1回×4）



（2）2個ずつのグループでペアを作り，大きい順に並べる



宝石の問題の解答例

（3）4個ずつのグループをひとつにまとめ，大きい順に並べる

（天秤7回）