

東北大学全学教育科目

情報基礎 A

Linux の利用

C言語によるプログラミングその3

担当: 大学院情報科学研究科

塩浦 昭義

プログラムその4:if ... else ...

条件判定, 場合分けをしたいときには if ... else ... を使う

```
#include <stdio.h>

main()
{
    int x;
    printf("input x =");
    scanf("%d", & x);
    if (x >= 0) {
        printf("x = %d is non-negative\n", x);
    } else {
        printf("x = %d is negative\n", x);
    }
}
```

このプログラムの動き:

- 端末に「input x=」と表示される
- キーボードから数値を入力
- 数値が非負ならば「x = *** is non-negative」と表示される
- 数値が負ならば「x = *** is negative」と表示される

if ... else ... の使い方

ここに条件式を入れる

条件式の形

- 「 $x \geq y$ 」(x は y 以上)
- 「 $z < 0$ 」(z は 0 未満)
- 「 $x == y$ 」(x は y に等しい)
- 「 $x != 0$ 」(x は 0 に等しくない)

```
if (x >= 0) {  
    printf("x = %d is non-negative¥n", x);  
} else {  
    printf("x = %d is negative¥n", x);  
}
```

条件式が成り立たない場合に実行する命令を、2番目の中括弧の中に書く複数でも可

条件式が成り立つ場合に実行する命令を、最初の中括弧の中に書く複数でも可

演習問題その4

問題4:

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「 x is larger than y 」と表示し、
 x が y 以下だったら「 x is not larger than y 」と表示する
プログラムを作成せよ。

- プログラム名は `ex4.c` としてください
-

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;
```

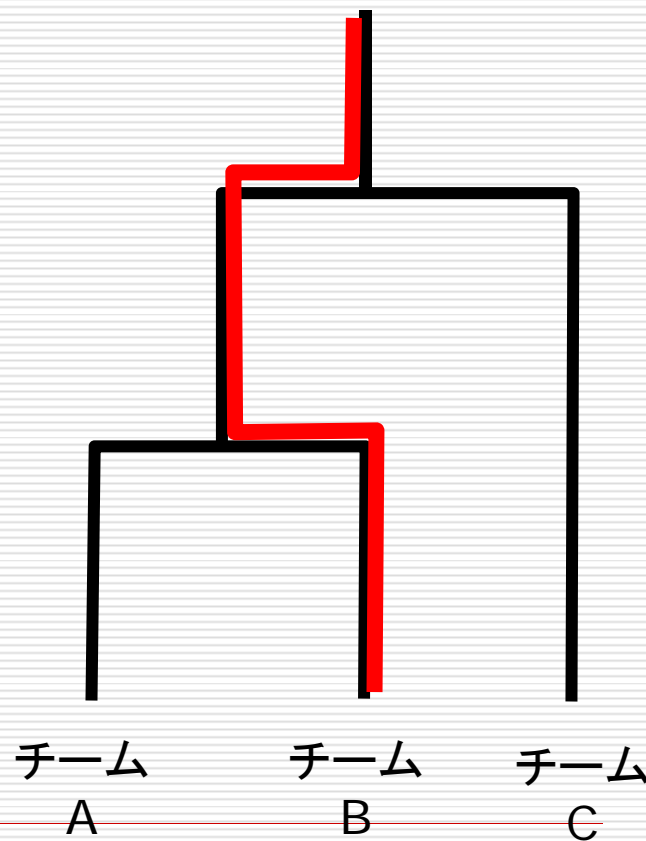
```
        }
```

```
        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: 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6

問題6:

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アを使う)

- プログラム名は `ex6.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?

演習問題その7



次の3つの問題に対し、回数およびその方法を答えなさい。

(プログラムを考える必要なし)

答えは OpenOffice Writer を使って書くこと(紙のレポート希望の場合は応相談)

一個の天秤と数個のメダルがあります(メダルの重さは全て異なる)。

天秤を使うと、重さの大小をしらべることができます。

天秤の皿には複数のメダルを置くことが可能です

問7-1:メダルが8個の場合に、一番重いメダルを探したい。

天秤を何回使うとできるだろうか？

問7-2:メダルが4個の場合に、重い順番に並べたい。

天秤を5回のみ使ってできるだろうか？

問7-3:メダルが8個の場合に、重い順番に並べたい。

天秤を何回使うとできるだろうか？

今日の課題レポート

- 演習問題4, 5, 6を解いて作ったプログラム ex4.c, ex5.c, ex6.c
 - 演習問題7の答え
 - 授業援助システムを使って提出してください
 - 締め切り:9月1日(木)お昼の12時まで
 - 授業時間外でわからないところが出てきたら...
 - 簡単な質問はメールにて対応します
 - 現場にいないと答えにくい質問については, テクニカルアシスタントの方に聞いてみてください(情報科目相談室(M102)にいます)
-