

プログラミングI 数理物理, 総合理学等向け

2018年10月22日

海谷 治彦

目次

- 演習3の解説
 - 日本語表示しないで.
 - scanf はシンプルに使う.
- if-else if文 (教科書は4章)
- 条件の組み合わせ
- 演習4

本日の演習3

- 身長と体重を入力し、身長から体重を引いた値が、
 - 100以下の場合、「やせろ」
 - それ以外は「ふとれ」

と生意気に助言するプログラムを作成せよ。

期待される結果の例

```
$ cc -o smbi sbmi.c
$ ./smbi
175
80
height = 175 weight = 80
Lose your weight!
$
```

これはユーザーが
175と80を入力して、
それぞれエンターを押す
ものとしてください。

演習3のポイント

- 条件式の中で計算式を扱うことができます.
- 別に身長と体重の差分を改めて別の変数に保存してもOKです.

- scanfについて 別紙
- 日本語表示について 別紙

scanfについて

- 単純に `scanf("%d", &x);` 等で値をとってください.
- 複数の値をとる場合, 上記を複数行書いてください.
- `scanf` は多機能なのですが, それを覚える必要はないので, 最も単純な使い方のみで押し通して.
 - いまどき, 実践的にはGUIでしょう.
- 教科書[レ]のやり方 `gets + atoi` はさらに複雑なので, 無視してください.
 - 加えて `gets` は利用が非推奨です(セキュリティ上).

日本語を表示させないで！

- **printf**文等を用いて、**日本語**を表示しようとし**ない**でください！
- **notepad** (メモ帳)は**機能が乏しい**ので、端末に対応した日本語ファイルを作成できません。
- 加えて、プログラム中の (や ; 等, **本来, 半角でなければいけない文字**の誤りも出ます。
 - これを目視で探すのは、かなり大変！
- **本格的な多言語対応**は高度なプログラミングになります。単一言語対応なら英語がベター。
- **日本語は表示させないで！**

if else if 文

else if は沢山かける

- 条件が複数通りある場合，else if を沢山書くことで，場合分けを細かくできる。

ある条件aが成り立ては，
命令群Aを実行する。
そうでなくて，条件bが成り立ては，
命令群Bを実行する
そうでなくて条件cが成り立てば，
命令群Cを実行する。
そうでなければ，
命令群Dを実行する。

```
// 一般形
if(条件a) {
    命令群A
}else if(条件b){
    命令群B
}else if(条件c){
    命令群C
}else{
    命令群D
}
```


例 tax4.c

ポイント:

if else は書いてある順番に判定されるので、
ココは $10000000 > \text{income} \geq 8000000$ の範囲となる。

```
// 実例 tax4.c
if(income >= 10000000){ // 収入が一千万以上なら
    tax = income *30/100; // 税率は30%だ
}else if (income >= 8000000){ // 1000~800
    tax = income *25/100; // 800万 税率25%
}else if (income >= 5000000){ // 800~500
    tax = income *22/100; // 税率22%
}else{ // 500万未満
    tax = income *20/100; // 税率は20%だ.
}
```

```
#include <stdio.h>
```

```
int main(void){ // list4-2x.c  
    int n, ret=0;
```

List 4-2改 p.85[レ]

```
    printf("Input the probability of rain: ");  
    scanf("%d", &n);  
    printf("It's %d %%. ¥n", n);
```

```
    if(n>100){ // ゼロ未満もチェックすべきですが, それは次に  
        printf("Incorrect probability, it should be 0 to 100! ¥n");  
        ret=1; // 例外に相当するので, 1を返すことにしました.  
    }else if(n>=50){  
        printf("You have to bring your umbrella. ¥n");  
    } else {  
        printf("You don't have to bring your umbrella. ¥n");  
    }  
    printf("Have a nice day!¥n");  
  
    return ret; // 変数の値利用可  
}
```

```
bash-3.1$ ./a  
Input the probability of rain: 108  
It's 108 %.  
Incorrect probability, it should be 0 to 100!  
bash-3.1$ ./a  
Input the probability of rain: 66  
It's 66 %.  
You have to bring your umbrella.  
Have a nice day!  
bash-3.1$
```

条件の組み合わせ

p.60-
(明)

- 二つの条件を組み合わせた条件を書くことができます。

× = 成り立たない(満たさな)
○ = 成り立つ(満たす)

条件1	条件2	条件1 かつ 条件2	条件1 または 条件2
×	×	×	×
×	○	×	○
○	×	×	○
○	○	○	○

○になるのは、
2つの条件が○のとき

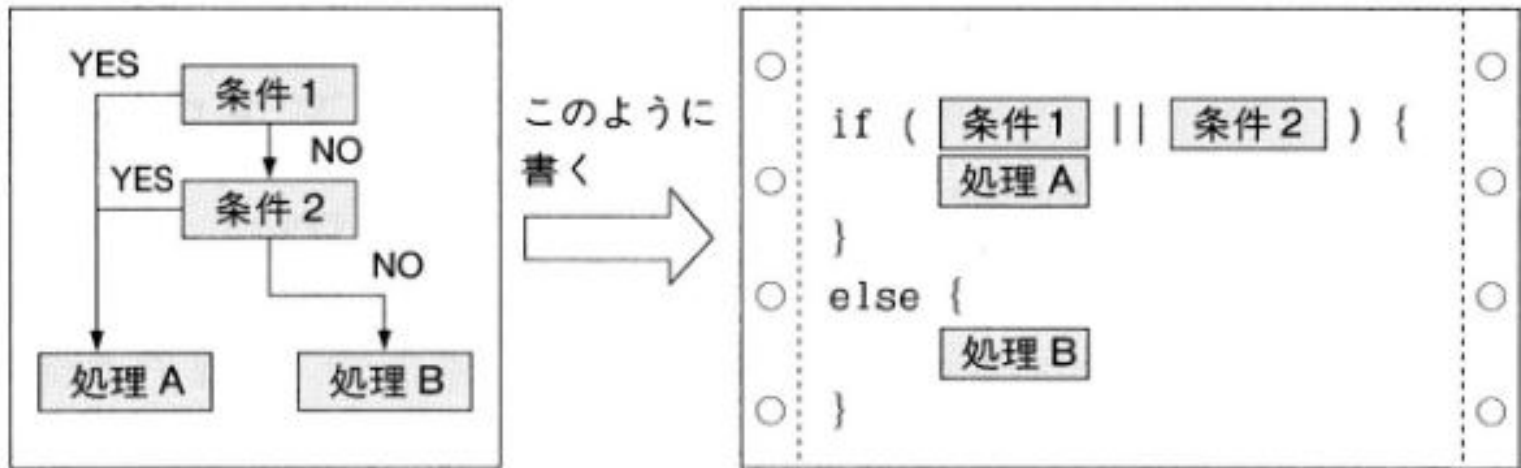
×になるのは、
2つの条件が×のとき

Cでは
条件1 && 条件2
と表記

Cでは
条件1 || 条件2
と表記

- 条件の否定を示す条件をかけます。
 - 「条件」に対して「!(条件)」と表記, ¬条件の意味.
 - 例 xは0より大きくない ! (x > 0)
 - すなわちxは0以下 (x ≤ 0)

図のイメージ



```
#include <stdio.h>
```

```
int main(void){ // list4-3x.c  
    int n;
```

```
    printf("Input the probability of rain¥n");  
    scanf("%d", &n);  
    printf("It's %d %%. ¥n", n);
```

以下のほうが自然かも
`if(!(0<=n && n<=100))`
「(0以上かつ100以下)では無い」

```
if( n<0 || n>100 ){ // ゼロ未満もチェックしました  
    printf("Incorrect probability, it should be 0 to 100! ¥n");  
    return 1; // 例外に相当するので, 1を返して終了してみました  
}else if(n>=50){  
    printf("You have to bring your umbrella. ¥n");  
} else {  
    printf("You don't have to bring your umbrella. ¥n");  
}  
  
printf("Have a nice day!¥n");  
return 0;  
}
```

Cでの単純な条件式

- 変数や定数の一致や, 大小関係
 - $i > 0$
 - $3 * r * r \geq 10$
 - $i == s$
 - $1 == 1$
 - 1が1と等しいのは当たり前なので常に真
- C言語では, 値 0 は, 「条件が成り立たないことを示す値」と定義されている.
 - $if(0)$ も文法的には正しい, 意味ないけど.
- 一方, 値 0 以外 (1, -1, 100 等) は, 条件が常に成り立つことを示す.
 - $if(1)$ 等も文法的には正しい, 意味ないけど.
 - 繰り返しの話が出てこないという意味がない.

条件の合成規則

- 恐らく高校数学で習ったと思うが、

- \wedge (and, かつ, Cでの `&&`)
- \vee (or, または, Cでの `||`)
- \neg (not, 否定, Cでの `!`)

の意味は以下のように定義されている. ちなみに、

- T (true, 条件が成り立つ値, Cでの0以外)
- F (false, 条件が成り立たない値, Cでの0) とする.

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

A	$\neg A$
T	F
F	T

条件式の合成

- A, B, C, D 等を「 x が10より大きい」等の条件とする.
 - C言語の場合 $x \geq 10$ 等
 - 実は $A = (x \geq 10)$; 等, 条件判断の「値」を整数として, 変数に保存してもよい.
- $\wedge \vee \neg$ を階層的に組み合わせて, 複雑な条件を構成することができる (高校数学で習ったと思う)
 - 例 $\neg(A \wedge (B \vee (C \wedge \neg D)))$
- Cの条件式も同様のことが可能である.
 - 例 $!(A \ \&\& \ (B \ || \ (C \ \&\& \ !D)))$
- if文での条件には, 上記のような条件を書くことができる.
 - 無論, 単純な条件, 例えば 数値1 や 0 等も書ける.

数学の条件とC言語の条件の違い

- 条件判断は、左から右に順番に行われる、結果、
- \wedge で繋がっている条件で、より左のものがfalseになれば、続く条件の評価は行われない。
 - $A \ \&\& \ B \ \&\& \ C \ \&\& \ D$ では、A, B, C, D の順で評価を行い、もし、Bが成り立たなければ、C, D の評価はしない。
- \vee で繋がっている条件で、より左のものがtrueになれば、続く条件の評価は行われない。
 - $A \ || \ B \ || \ C \ || \ D$ では、A, B, C, D の順で評価を行い、もし、Bが成り立たれば、C, D の評価はしない。
- $A \ \&\& \ B$ と $B \ \&\& \ A$ の意味が変わる場合がある。
 - 数学では、 $A \wedge B$ と $B \wedge A$ の意味は常に同じ。

例 tax4.c

ポイント:

if else は書いてある順番に判定されるので、
ココは $10000000 > \text{income} \geq 8000000$ の範囲となる。

```
// 実例 tax4.c
if(income >= 10000000){ // 収入が一千万以上なら
    tax = income *30/100; // 税率は30%だ
}else if (income >= 8000000){ // 800万以上
    tax = income *25/100; // 800万 税率25%
}else if (income >= 5000000){ // 500万以上
    tax = income *22/100; // 税率22%
}else{ // その他は
    tax = income *20/100; // 税率は20%だ.
}
```

同じ意味

上のほうが、
記述が少ないでしょ？

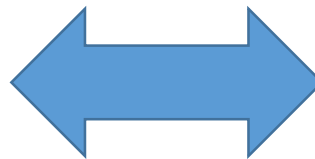
```
// 実例 tax4.c
if(income >= 10000000){ // 収入が一千万以上なら
    tax = income *30/100; // 税率は30%だ
}else if (income >= 8000000){ // 800万以上
    tax = income *25/100; // 800万 税率25%
}else if (income >= 5000000){ // 500万以上
    tax = income *22/100; // 税率22%
}else{ // その他は
    tax = income *20/100; // 税率は20%だ.
}
```

```
// 実例 tax5.c
if(income >= 10000000 ){ // 収入が一千万以上なら
    tax = income *30/100; // 税率は30%だ
}
if (10000000 > income && income >= 8000000){ // 1000万~800万
    tax = income *25/100; // 税率25%
}
if (8000000 > income && income >= 5000000){ // 800万~500万
    tax = income *22/100; // 税率22%
}
if(5000000 > income){ // 500万未満
    tax = income *20/100; // 税率は20%だ.
}
```

組み合わせ条件 vs ifの入れ子

- 条件をANDで組み合わせる処理は、if文のブロックを入れ子に組み合わせることで記述することと意味が同じ.
- ただし、組み合わせ条件の場合、事前に条件判定のための情報が全て必要.

```
if(条件A && 条件B) {  
    何かやる;  
}
```



```
if(条件A) {  
    if(条件B) {  
        何かやる;  
    }  
}
```

組み合わせ条件の例

```
// freeice1.c
// 年齢が15歳以下で支払い価格が2000円以上ならアイスクリームをサービス
// && による複合条件
#include <stdio.h>

int main(void){
int age, charge;
    printf("input your age: ");
    scanf("%d", &age);
    printf("input your total payment: ");
    scanf("%d", &charge);

    if( age <= 15 && charge >= 2000 ){
        printf("free ice cream for you!¥n");
    }

    return 0;
}
```

if文の入れ子の例

```
// freeice2.c
// if文の入れ子
#include <stdio.h>

int main(void){
int age, charge;
    printf("input your age: ");
    scanf("%d", &age);
    printf("input your total payment: ");
    scanf("%d", &charge);

    if( age <= 15 ){
        if( charge >= 2000 ){
            printf("free ice cream for you!¥n");
        }
    }

    return 0;
}
```

if文の入れ子の例2

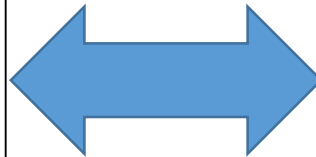
```
// freeice3.c
// 支払い価格が2000円以上でないなら年齢は聞かない,
// プライバシーに配慮っぽいかも.
#include <stdio.h>

int main(void){
int charge;
    printf("input your total payment: ");
    scanf("%d", &charge);
    if( charge >= 2000 ){
int age;
        printf("input your age: ");
        scanf("%d", &age);
            if( age <= 15 ){
                printf("free ice cream for you!¥n");
            }
        }
    }
return 0;
}
```

組み合わせ条件 vs else if

- 条件をORで組み合わせる処理は, if else文で記述することと意味が同じ.
- 同じ内容の「何かやる」を else if でも書くのでちよいと冗長. (プログラムとしてはとても良くない)

```
if(条件A || 条件B){  
    何かやる;  
}
```



```
if(条件A){  
    何かやる;  
}else if(条件B){  
    上と同じ何かやる;  
}
```


注意: 代入文も条件式になるが...

- 実は**代入文自体も条件式**の一種とみなせます.
 - この使い方は難しいので**当面は使わないこと**.
 - 結果, cc がエラーとして検出することは無い.
- ただし, 単なる**値の比較**とは意味が違う.
 - 代入された値が0か否かで真偽を判断
- 以下の二つのような例は意味がまったく異なります, 厳重に注意してください. (**違いは, たった1文字**の =)

```
scanf("%d", &x);  
if(x==10){  
    //xは入力時点のまま  
    何かやる;  
}
```

```
scanf("%d", &x);  
if(x=10){ // 常に成り立つ  
    //xは10になってしまっている  
    何かやる;  
}
```

本日の演習5

- 身長と体重を入力し、身長から体重を引いた値が、
 - 100以下の場合、「やせろ」
 - 120以上の場合、「ふとれ」
 - それ以外は無言

と生意気に助言するプログラムを作成せよ。

期待される結果の例

```
$ cc -o sbmi2 sbmi2.c
$ ./sbmi2
175
80
height = 175 weight = 80
Lose your weight!
$
```

これはユーザーが
175と80を入力して、
それぞれエンターを押す
ものとしてください。

以上