

cp-4. 条件分岐と場合分け

(C プログラミング入門)

URL: <https://www.kkaneko.jp/pro/adp/index.html>

金子邦彦



例題 1. 平方根の計算

if文, else 文, 条件分岐

比較演算

例題 2. 多分岐の例

else if 文

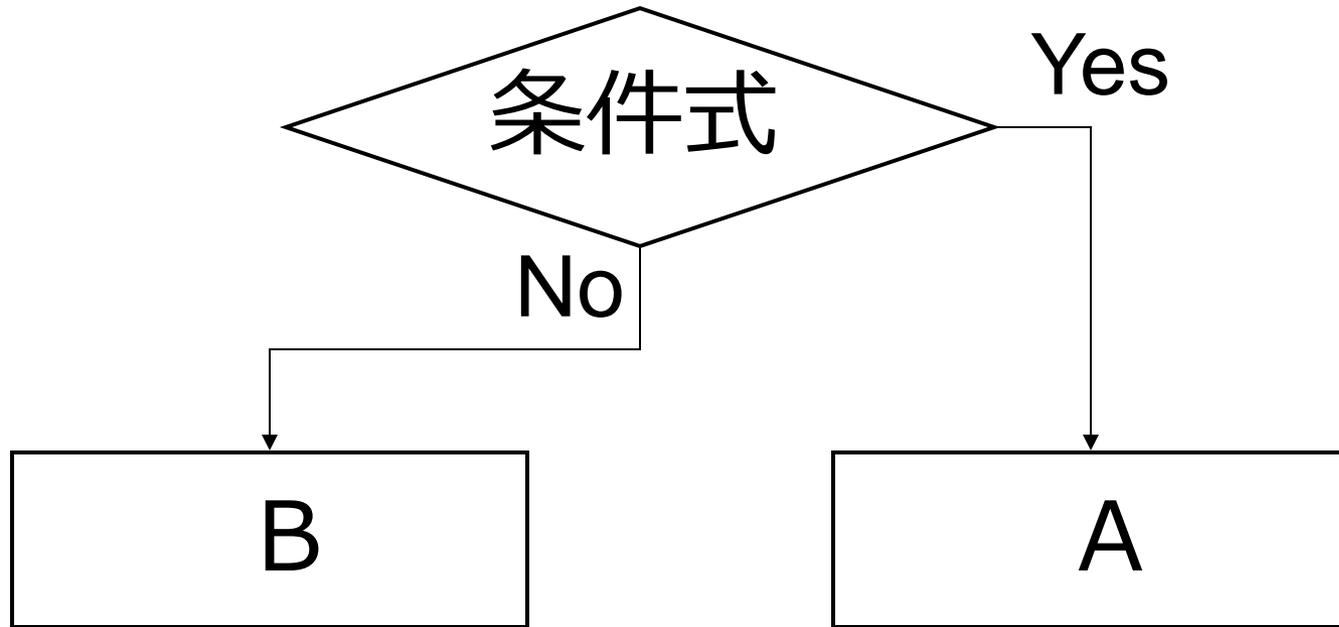
例題 3. うるう年の判定

比較演算と論理演算の組み合わせ

目標

- 条件分岐 (if 文) を使って, より役に立つプログラムを作れるようになる.
 - 比較演算 (<, <=, >, >=, ==, !=) の使い方を理解する
 - 比較演算と論理演算 (&&, ||, !) の組み合わせ
 - 前回習った四則演算, ライブラリ関数 (三角関数, 指数・対数関数など) も使うこと

条件分岐とは



- 「ある条件式」が成り立てばAを、成り立たなければBを実行

計算における条件分岐の例

- 平方根の計算
 - 正または0ならば → `sqrt()` を使って計算できる
 - 負の数ならば → 平方根は計算できない
- 対数の計算
 - 正ならば → `log()` を使って計算できる
 - 0または負の数ならば → 対数は計算できない

など

例題 1 . 平方根の計算

- 浮動小数データを読み込んで、平方根の計算と表示を行うプログラムを作る。
 - 但し、負の数の場合には、メッセージを表示すること。
 - 負の数であるかどうかによって条件分岐を行うために if 文を使う。

例) 9 のとき : 3

- 1 のとき : メッセージを表示

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    printf("x=");
    scanf("%lf",&x);
    if ( x < 0 ) {
        printf("負なので計算できません\n");
    }
    else {
        y = sqrt(x);
        printf("sqrt(%f)=%f\n", x, y);
    }
    return 0;
}
```

条件式

条件が成り立つ
場合に実行され
る部分

条件が成り立た
ない場合に実行
される部分

平方根の計算

実行結果の例

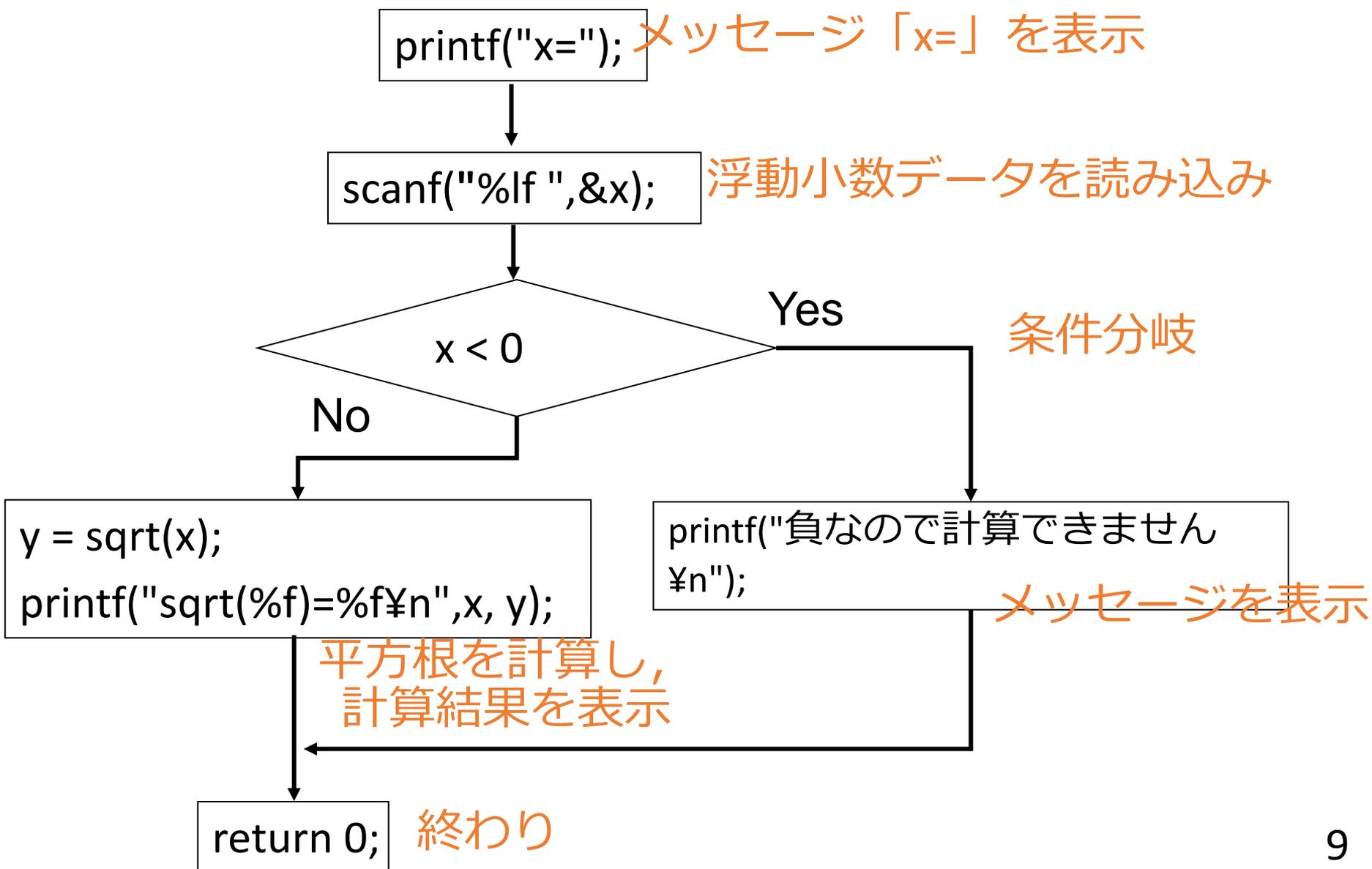
x=9

`sqrt(9.000000)=3.000000`

x= - 1

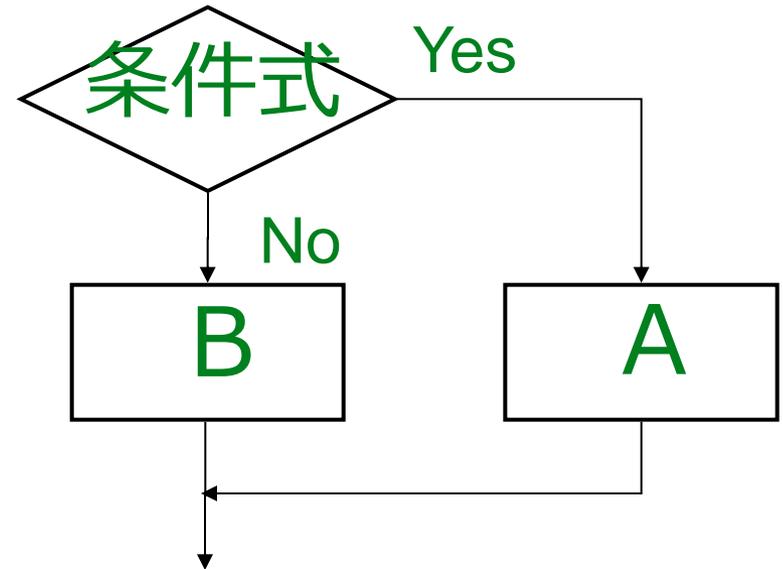
負なので計算できません

プログラム実行順



if 文と else 文

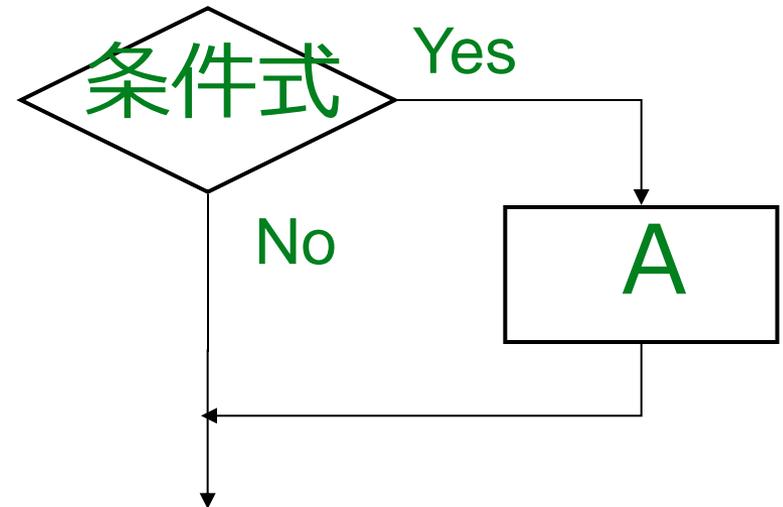
```
if ( 条件式 ) {  
    式;  
    ... ;  
} A  
else {  
    式;  
    ... ;  
} B  
}
```



- 「条件式」が成り立てばAを、成り立たなければBを実行

if 文

```
if ( 条件式 ) {  
    式;  
    ... ;  
} A
```



- if 文のみを書いて, else 文を書かないこともできる
- 「ある条件」が成り立つときに限り A を実行

比較演算

- 条件式の中には、ふつう、比較演算を書く

- 演算子

意味

<

左辺が右辺より小さい

<=

左辺が右辺以下

>

左辺が右辺より大きい

>=

左辺が右辺以上

= =

左辺が右辺と等しい

! =

左辺が右辺と等しくない

比較演算の例

「左辺が右辺以上」の意味

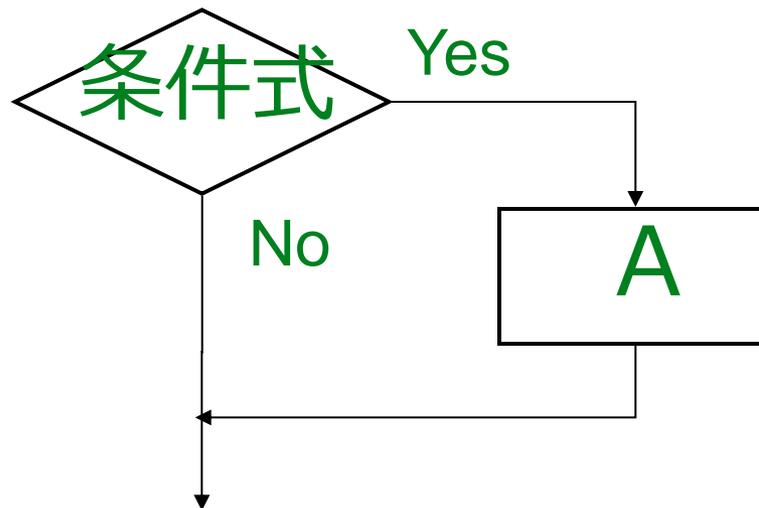
```
if (age >= 20 ){  
    printf("You may drink alcoholic beverage. ");  
}  
else{  
    printf("You may not drink alcoholic beverage.");  
}
```

ここまでのまとめ



```
if ( 条件式 ) {  
  式;  
  ... ;  
}
```

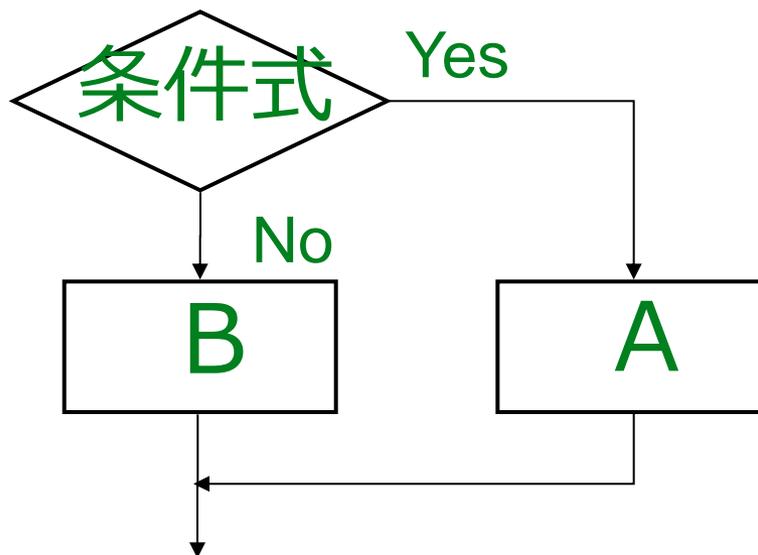
A



```
if ( 条件式 ) {  
  式;  
  ... ;  
}  
else {  
  式;  
  ... ;  
}
```

A

B



字下げとセミコロンを忘れないこと



- セミコロンを忘れると

- プログラムは動かない

字下げ

```
if (条件式) {  
  文;  
  ...;  
}
```

セミコロン

- 字下げを忘れると

- プログラムは動くが、読みづらい

字下げ

字下げ

```
if (条件式) {  
  文;  
  ...;  
}  
else {  
  文;  
  ...;  
}
```

セミコロン

例題 2 . 多分岐の例



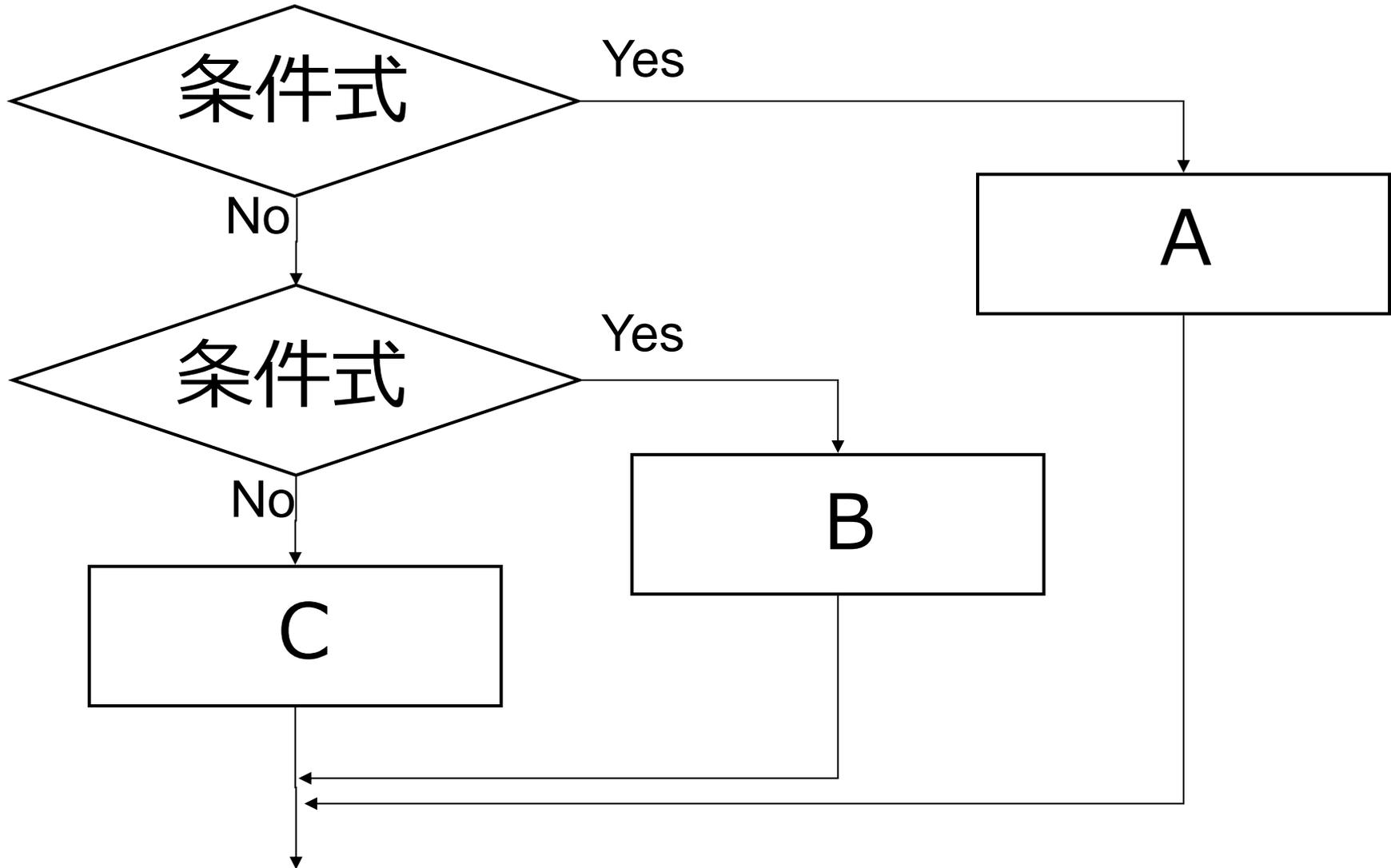
- キーボードから数値を読み込んで,
 - 1 3 以上なら 「100 Yen」
 - 6 以上なら 「50 Yen」
 - それ以外なら 「0 Yen」と表示するプログラムを作る
- 多分岐を行うために, else if 文を使う

```
#include <stdio.h>
#pragma warning(disable:4996)
int main()
{
    int a;
    printf("a=");
    scanf("%d",&a);
    if ( a > 13 ) {
        printf("100 Yen¥n");
    }
    else if( a > 6 ) {
        printf("50 Yen¥n");
    }
    else {
        printf("0 Yen¥n");
    }
    return 0;
}
```

このうちどれか1つ
が実行される

多分岐の例

A, B, Cのうちどれか1つを実行

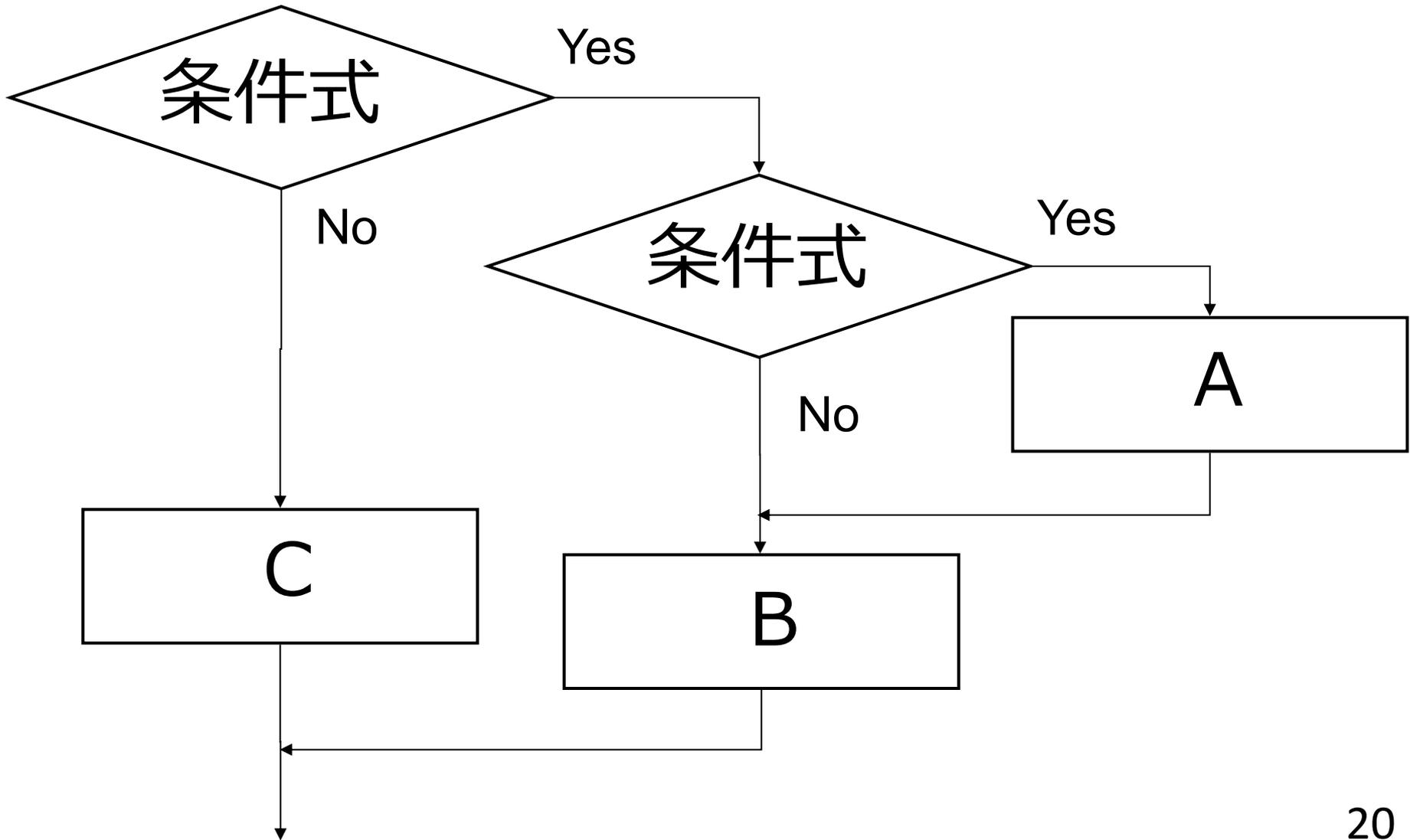


if 文の入れ子

```
if ( 条件式 ) {  
    if ( 条件式 ) {  
        式;  
        ... ;  
    }  
    式;  
    ... ;  
}  
else {  
    式;  
    ... ;  
}
```

- if 文は, 上のように, 入れ子にすることもできる

if 文の入れ子



課題 1. 2次方程式

- 2次方程式 $ax^2 + by + c = 0$ の解を求めるプログラムを作りなさい
 - 重解, 虚数解も正しく求めなさい (判別式 $b^2 - 4ac$ の値で条件分岐する)
 - 余裕があれば, (1) $a=0$, (2) $a=0$ かつ $b=0$, (3) $a=0$ かつ $b=0$ かつ $c=0$ の場合にも, 正しく解を求めるようにしなさい
 - 「`#include <math.h>`」を忘れない
 - 複素数は, 実数部と虚数部に分けて扱う (実数部用と虚数部用の変数を使う)
 - 解が2つあるので, そのための変数を使う

課題 2. if 文の入れ子

価格と重量（ともに浮動小数データ）を読み込んで、

価格 1000 以上で、重さ 100 以上 → Unnecessary

価格 1000 以上で、重さ 100 未満 → Expensive

価格 1000 未満で、重さ 100 以上 → Reasonable

価格 1000 未満で、重さ 100 未満 → Cheap

と表示するプログラムを作成せよ

例題 3. うるう年の判定



- 「西暦年」を読み込んで、うるう年かどうか表示するプログラムを作る。
 - うるう年の判定のために、比較演算と論理演算を組み合わせる

例) 2 0 0 1 → 2001 is not a leap year.

2 0 0 4 → 2004 is a leap year.

うるう年の判定



```
#include <stdio.h>
#pragma warning(disable:4996)
int main()
{
    int y;
    printf("year=");
    scanf("%d",&y);
    if (((y % 400) == 0) || (((y % 100) != 0) && ((y % 4) == 0))) {
        printf(" %d is a leap year.¥n ",y);
    }
    else {
        printf("%d is not a leap year.¥n ",y);
    }
    return 0;
}
```

条件式

条件が成り立つ
場合に実行され
る部分

条件が成り立た
ない場合に実行
される部分

うるう年の判定式



$((y \% 400) == 0) \ || \ (((y \% 100) != 0) \ \&\& \ ((y \% 4) == 0))$

400の倍数である 100の倍数でない 4の倍数である

または

かつ

論理演算



- $A \ \&\& \ B$ A かつ B
- $A \ \|\ B$ A または B
- $!A$ A でない

真, 偽に関する論理的な演算を行う.

比較演算と論理演算の組み合わせ



例)

```
if (( m == 1 ) || ( m == 2 )) {  
    y = y - 1;  
    m = m + 12;  
}
```

論理演算子

比較演算子

m が 1 , または m が 2 の時に限り実行

課題 3. 曜日を求めるプログラム

- ツエラーの公式を使い, 年, 月, 日を読み込んで, 曜日を求めるプログラムを作成しなさい.
 - ツエラーの公式については, 次ページの解説を参照せよ
 - 計算された曜日は, 数字として表示すること

0 : 日曜日

1 : 月曜日

2 : 火曜日

3 : 水曜日

4 : 木曜日

5 : 金曜日

6 : 土曜日

ツイラーの公式

$$(y + (y/4) - (y/100) + (y/400) + ((13 * m + 8) / 5) + d) \% 7$$

- この値が0なら日曜, 1なら月曜 . . .

- 年, 月, 日を表す変数 y, m, d を, 整数データとして宣言すること. つまり, 「/4」, 「/5」などは, 割り算を行って小数点以下切り捨てと考えよ.

- ツイラーの公式では, 「1年の起点を3月とし、月は3月から14月までである」と考えている

- 1月, 2月は, 前年の13, 14月と考えるということ

- ヒント (意味を理解してから使うこと) :

```
if ( ( m == 1 ) || ( m == 2 ) ) {
```

```
    y = y - 1;
```

```
    m = m + 12;
```

```
}
```

より勉強したい人への付録

if 文での {, } の省略

```
if (age >= 20 ) {  
    printf("You can smoke");  
}  
else {  
    printf("You can not smoke");  
}
```



同じ意味

```
if (age >= 20 )  
    printf("You can smoke");  
else  
    printf("You can not smoke");
```

- if文での {, } を省略したプログラムが時々あるので慌てないこと
 - {, } 内に1つの式あるいは文しか書かない場合に限り省略可
- {, } を省略すると, しばしば, プログラムが読みにくくなる