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

CC BY NC SA

内容

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

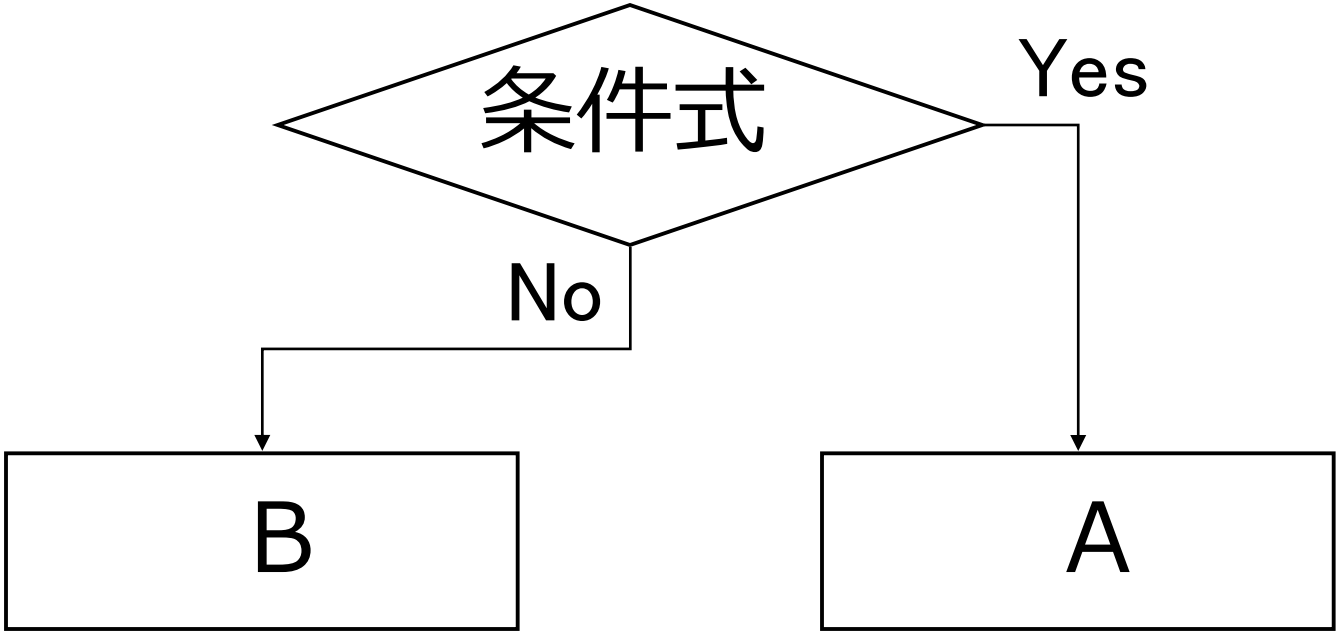
例題３．うるう年の判定

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

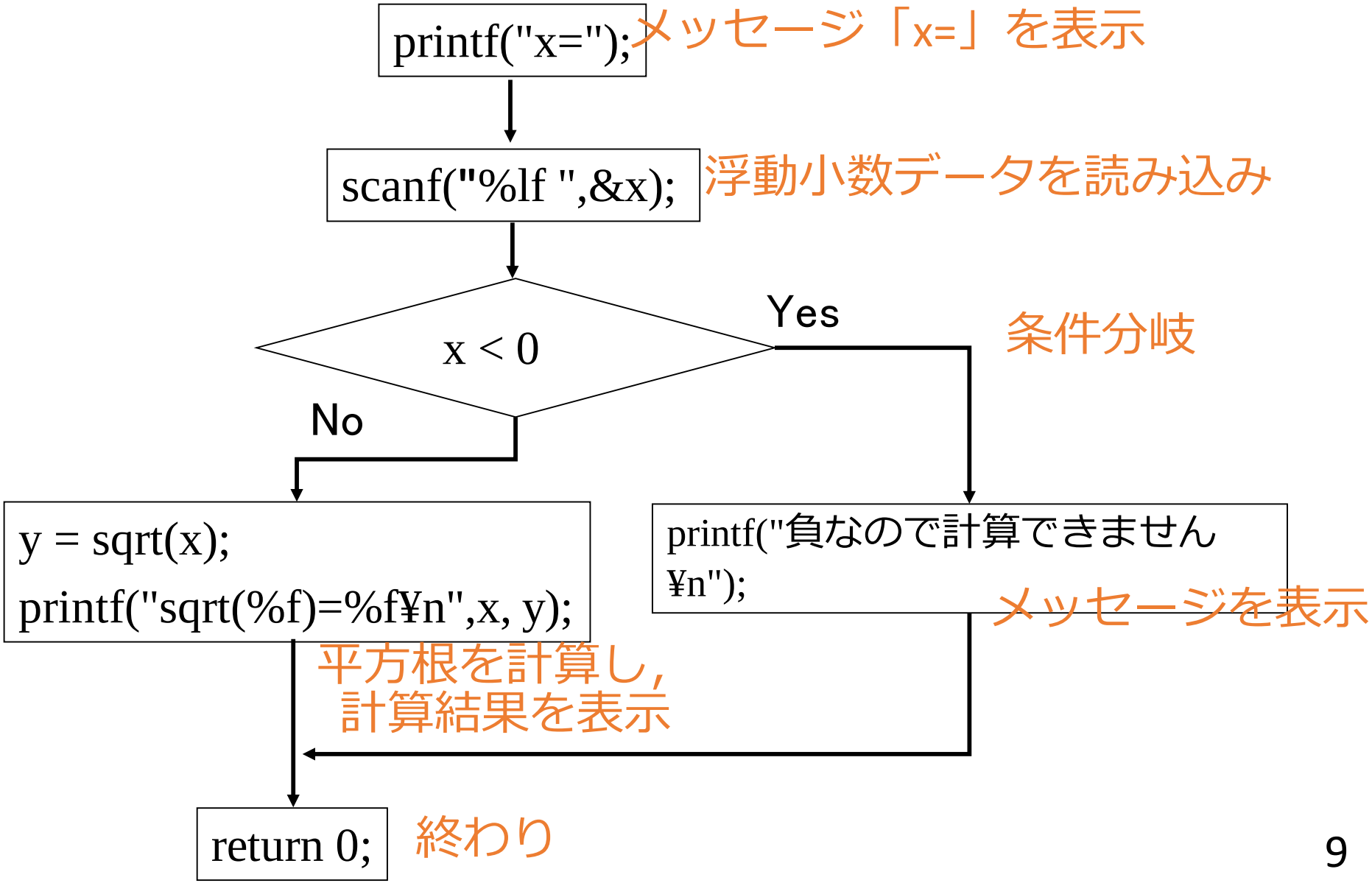
```
x=9

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

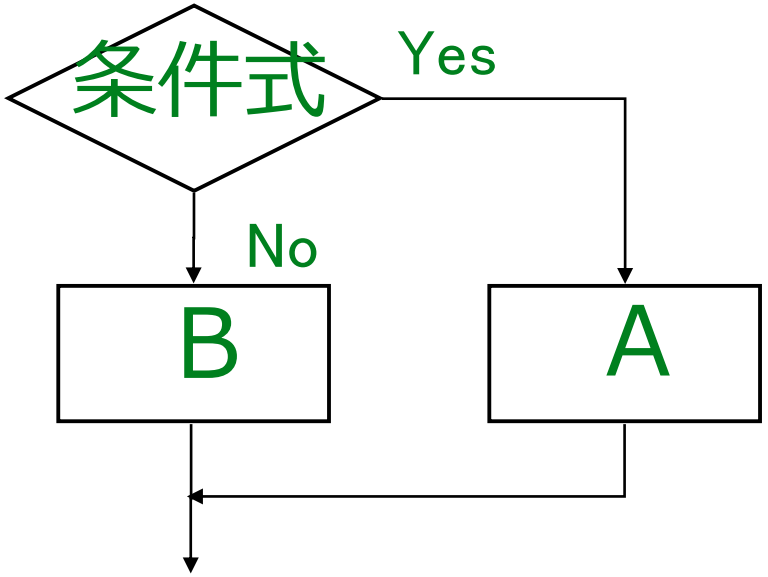

printf("x=");
メッセージ「x=」を表示
scanf("%lf ",&x);
浮動小数データを読み込み
x < 0
Yes
条件分岐
No
y = sqrt(x);
printf("sqrt(%f)=%f¥n",x, y);
平方根を計算し，
計算結果を表示
printf("負なので計算できません
¥n");
メッセージを表示
return 0;
終わり

if 文と else 文

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

A

B

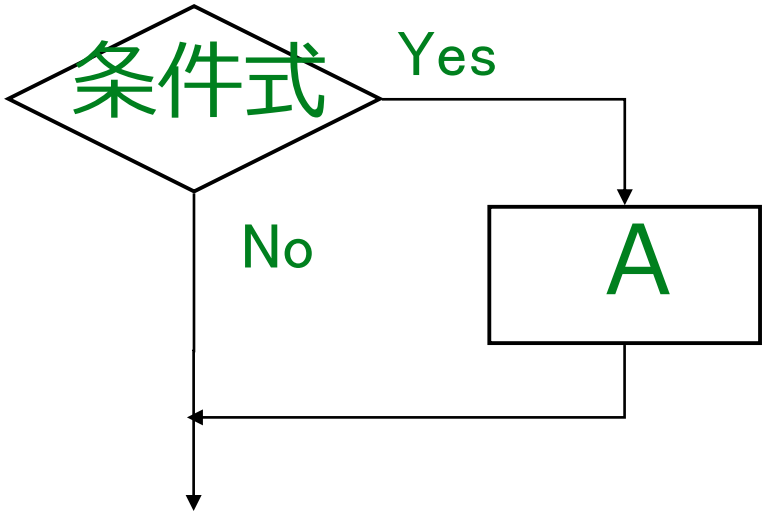


- 「条件式」が成り立てばＡを、成り立たなければＢを実行

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

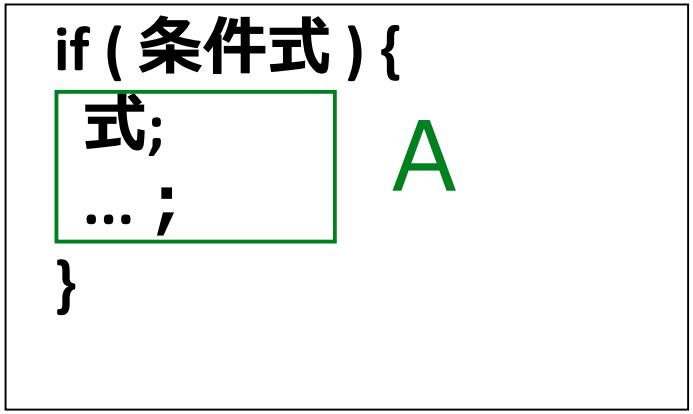
＜	左辺が右辺より小さい
＜＝	左辺が右辺以下
＞	左辺が右辺より大きい
＞＝	左辺が右辺以上
＝＝	左辺が右辺と等しい
！＝	左辺が右辺と等しくない

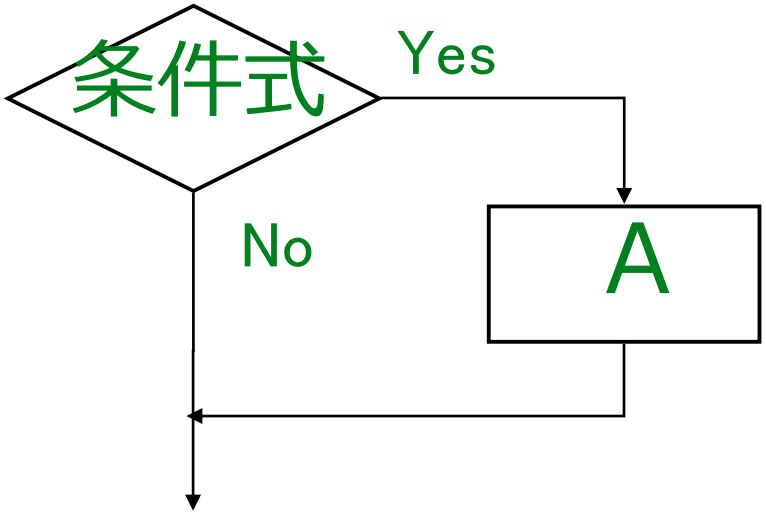
比較演算の例

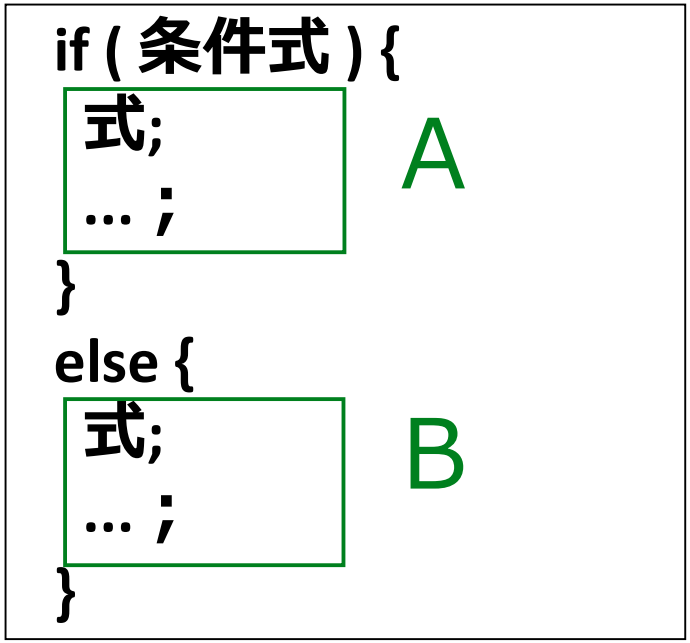
「左辺が右辺以上」の意味

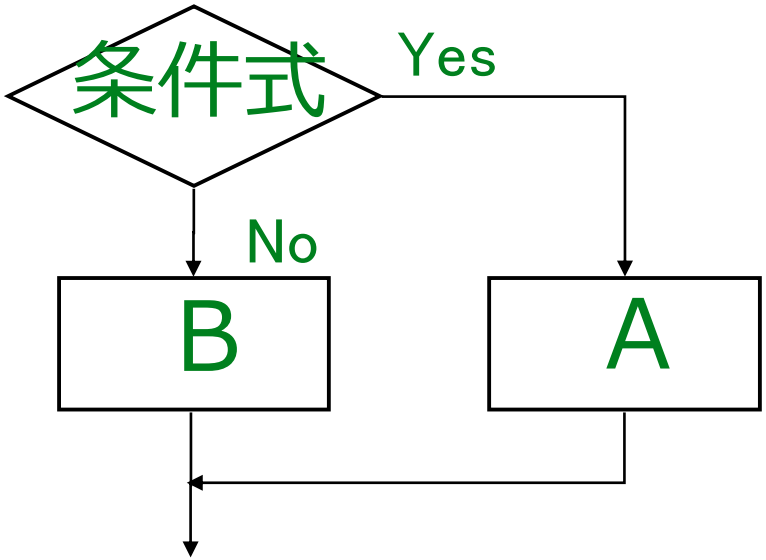
```
if  (age >= 20 ){
    printf("You may drink alcoholic beverage. ");
}
else{
    printf("You may not drink alcoholic beverage.");
}
```

ここまでのまとめ









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
- 字下げを忘れると
  - プログラムは動くが，読みづらい

```
if ( 条件式 ) {
  文;
  … ;
}
```

字下げ

セミコロン

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

字下げ

セミコロン

字下げ

例題２．多分岐の例

- キーボードから数値を読み込んで，

  １３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

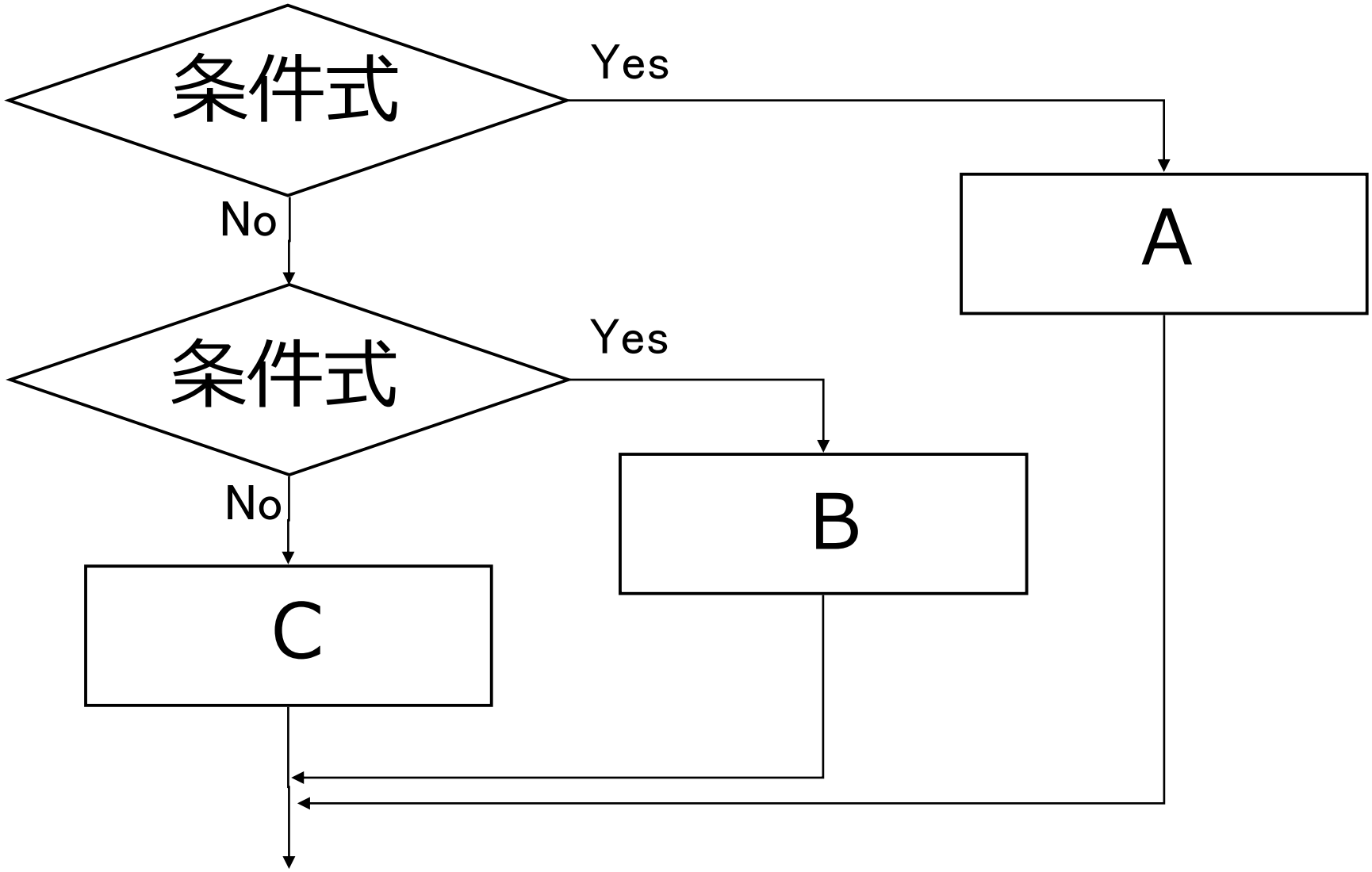
- 多分岐を行うために，else if 文を使う

```
#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

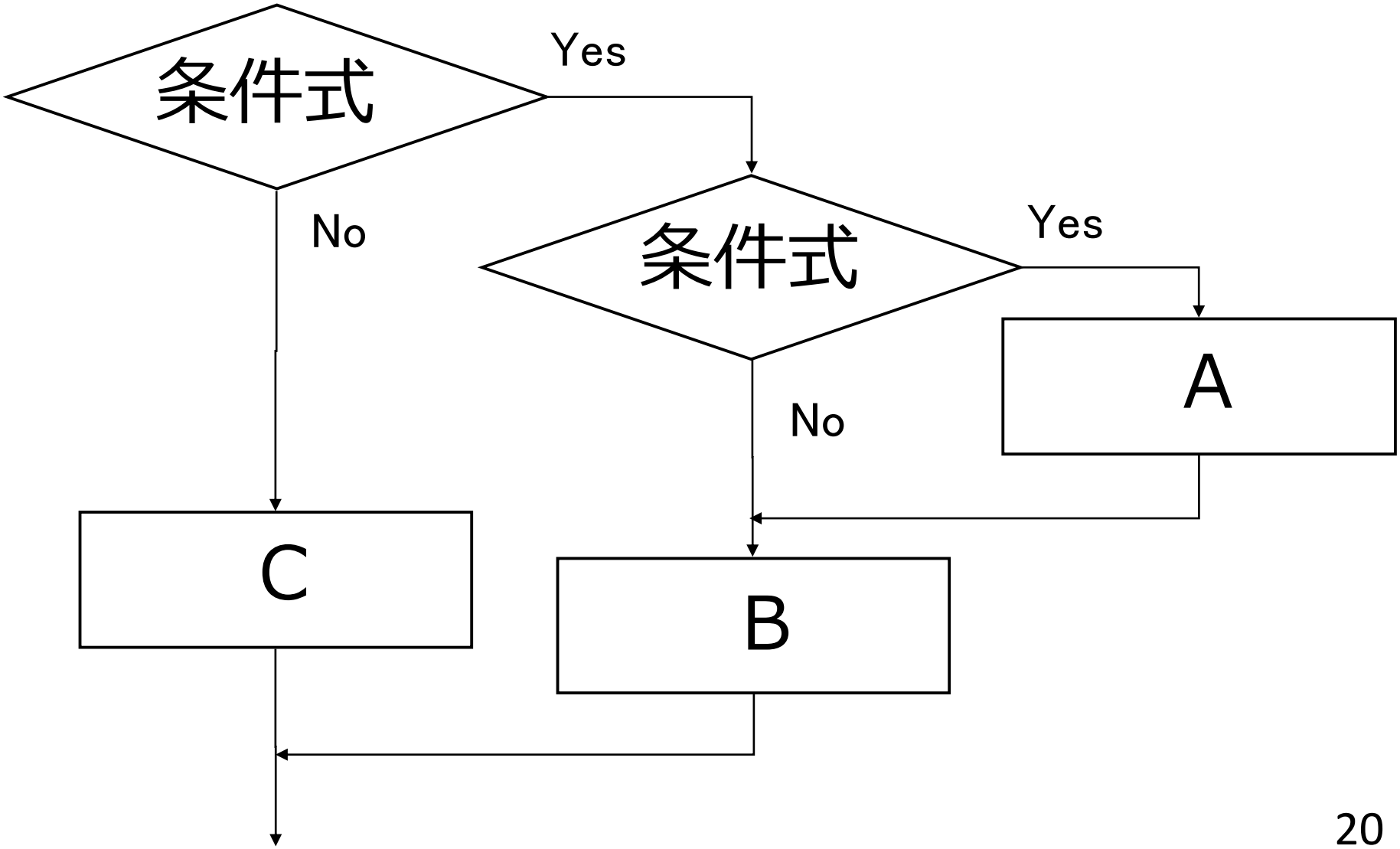


if 文の入れ子

```
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）　２００１　→ 2001 is not a leap year.

　　２００４　→ 2004 is a leap year.

グレゴリオ暦でのうるう年

- うるう年とは：　2月が29日まである年
- うるう年は400年に97回で，1年の平均日数は365.2422日
- うるう年の判定法
  - 年数が4の倍数の年　→ うるう年
  - 但し，100の倍数の年で400の倍数でない年

    → うるう年ではない

    （４の倍数なのだが例外とする）

(例)　２００８年：うるう年（４の倍数）
２００４年：うるう年（４の倍数）
２０００年：うるう年（４の倍数）
１９００年：うるう年ではない
（１００の倍数だが４００の倍数でない）
１８００年：うるう年ではない
（１００の倍数だが４００の倍数でない）

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

$$\underbrace{((y \% 400) == 0)}_{\text{400の倍数である}} \ \| \ \underbrace{((y \% 100) != 0)}_{\text{100の倍数でない}} \ \&\& \ \underbrace{((y \% 4) == 0))}_{\text{4の倍数である}}$$

かつ

または

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

ツエラーの公式

(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.

- ツエラーの公式では，「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