



pe-3. 条件分岐と場合分け

(Pascal プログラミング入門)

URL: <https://www.kkaneko.jp/cc/pascal/index.html>



金子邦彦



前回のまとめ



Pascalには、四則演算と各種ライブラリ関数（三角関数、指数・対数関数、平方根など）の機能がある

```
s := (a + b + c) / 2;
```

$(a+b+c)/2$ を計算し、s に格納

```
A := sqrt(s * (s - a) * (s - b) * (s - c));
```

$\sqrt{s(s-a)(s-b)(s-c)}$ を計算し、A に格納

内容



例題 1. 平方根の計算

例題 2. 対数の計算

if文, else 文, 条件分岐, 比較演算

例題 3. 定形郵便物の料金

例題 4. 2次方程式

多分岐

例題 5. 直線と原点の距離

比較演算と論理演算の組み合わせ

目標

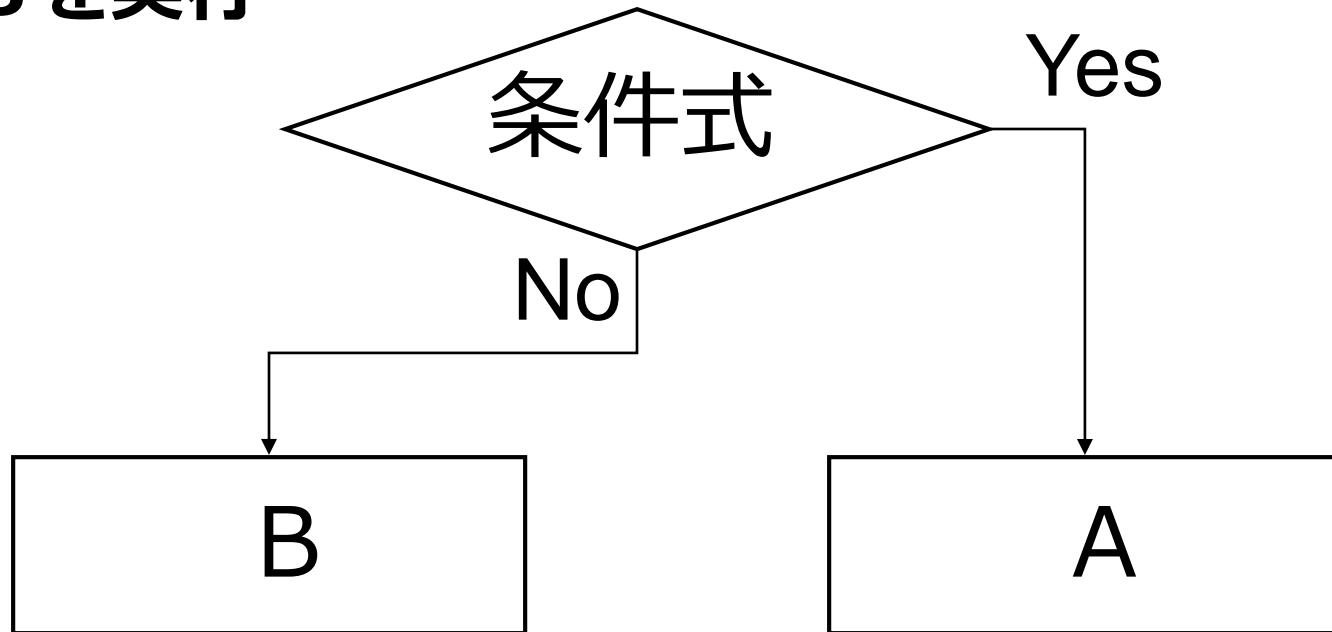


- **条件分岐 (if 文)** を使って, より役に立つプログラムを作ることができるようになる.
- **比較演算** ($<$, $<=$, $>$, $>=$, $=$, $<>$) の使い方を理解する
- 前回習った**四則演算**, **ライブラリ関数** (三角関数, 指数・対数関数など) も使う

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





- プログラミングを行えるオンラインのサービス

<https://www.onlinegdb.com>

- ウェブブラウザを使う

- たくさんの言語を扱うことができる

Pascal, Python3, Java, C/C++, C#, JavaScript,
R, アセンブリ言語, SQL など

- オンラインなので、「秘密にしたいプログラム」
を扱うには十分な注意が必要

Online GDB で Pascal を動かす手順



① ウェブブラウザを起動する

② 次の URL を開く

<https://www.onlinegdb.com>

A screenshot of a web browser's address bar. The address bar is a light gray rectangle with a thin border. On the left side, there is a magnifying glass icon. To the right of the icon, the text "https://www.onlinegdb.com" is displayed in a dark gray font. Below the address bar, there is a thin horizontal line, and below that, a larger light gray rectangular area representing the browser's content area.



③ 「Language」 のところで、「Pascal」 を選ぶ

The screenshot shows a web-based IDE interface. At the top, there is a navigation bar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a 'Language' dropdown menu. The 'Language' dropdown is highlighted with a red box. Below the dropdown, a list of programming languages is displayed, including C, C++, C++ 14, C++ 17, Java, Python, PHP, C#, VB, HTML,JS,CSS, Ruby, Perl, Pascal, R, Fortran, Haskell, Assembly(GCC), Objective C, and SQLite. The 'Pascal' option is highlighted with a red box. The main editor area shows a C program with a 'printf' statement that prints 'Hello World'.

```
1 - /*****  
2  
3 Welcome to GDB OnLine.  
4 GDB online is an online compiler and debugger tool for C, C++, Python  
5 C#, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML, CSS, JS  
6 Code, Compile, Run and Debug online from anywhere in world.  
7  
8 *****/  
9 #include <stdio.h>  
10  
11 int main()  
12 {  
13     printf("Hello World");  
14  
15     return 0;  
16 }  
17
```




実行ボタン

```
main.pas
1 {
2
3 Welcome to GDB Online.
4 GDB online is an online compiler and debugger tool for C, C++, Python, Java,
5 C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS,
6 Code, Compile, Run and Debug online from anywhere in world.
7
8 }
9 program Hello;
10 begin
11     writeln ('Hello World')
12 end.
13
```

Copyright (c) 1993-2017 by Florian Klaempfl and others
Target OS: Linux for x86-64
Compiling main.pas
Linking a.out
/usr/bin/ld.bfd: warning: link.res contains output sections; did you forget -T?
12 lines compiled, 0.1 sec
Hello World

...Program finished with exit code 0
Press ENTER to exit console.

エディタ画面

プログラムを
書き換えること
ができる

例題 1. 平方根の計算



- 浮動小数点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、「**計算できないこと**」を意味するメッセージを表示する
 - 正または0のとき： `sqrt()` を使って計算できる
 - 負のとき： 平方根は計算できない
 - **負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



```
program sum;
```

```
var x: real;
```

```
begin
```

```
  write('Please Enter x: ');
```

```
  readln(x);
```

```
  if x < 0 then begin
```

```
    writeln('x < 0, then x is out of range');
```

```
  end
```

```
  else begin
```

```
    writeln('sqrt(', x:8:3, ') =', sqrt(x):8:3);
```

```
  end;
```

```
  readln
```

```
end.
```

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

実行結果の例

```
Please Enter x: 9  
sqrt( 9.000) = 3.000
```

```
Please Enter x: -5  
x < 0, then x is out of range
```

プログラム実行順

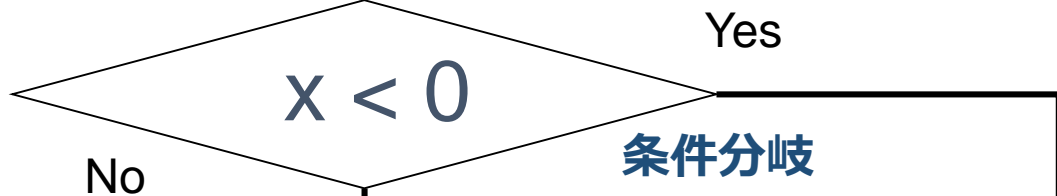


```
write('Please Enter x: ');
```

メッセージ
「Please Enter x:」を表示

```
readln(x);
```

浮動小数点数データを読み込み



Yes

条件分岐

No

```
writeln('sqrt(', x:8:3, ') =', sqrt(x):8:3);
```

平方根を計算し、計算結果を表示

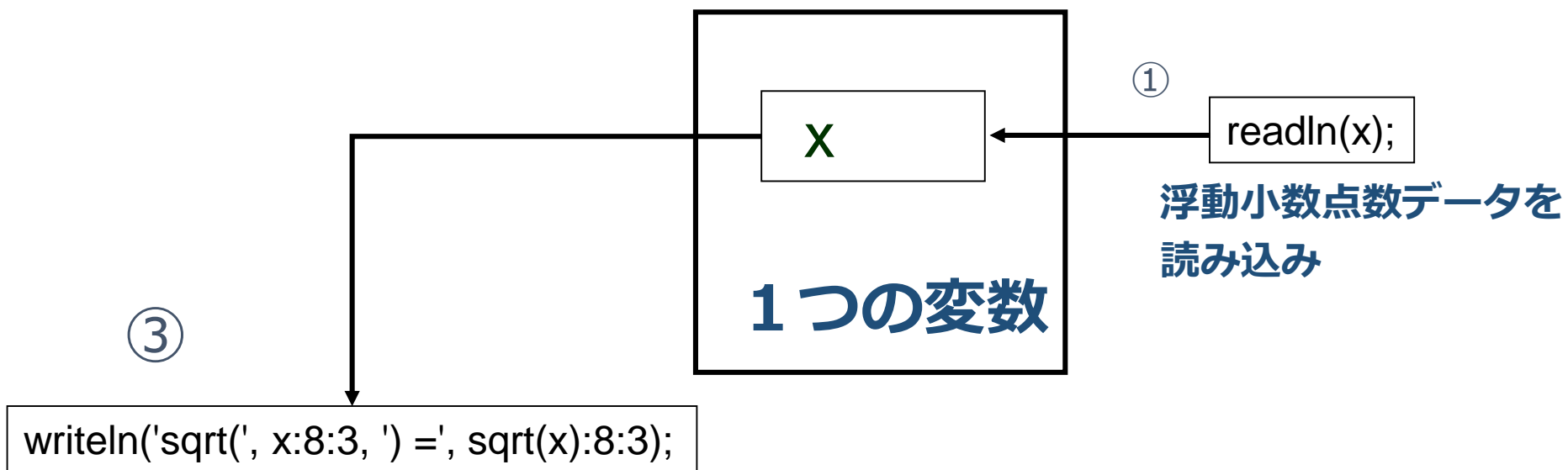
```
writeln(' x < 0, then x is out of range');
```

メッセージを表示

```
readln
```

終わりの readln

メモリ



計算と表示

if ... else 文

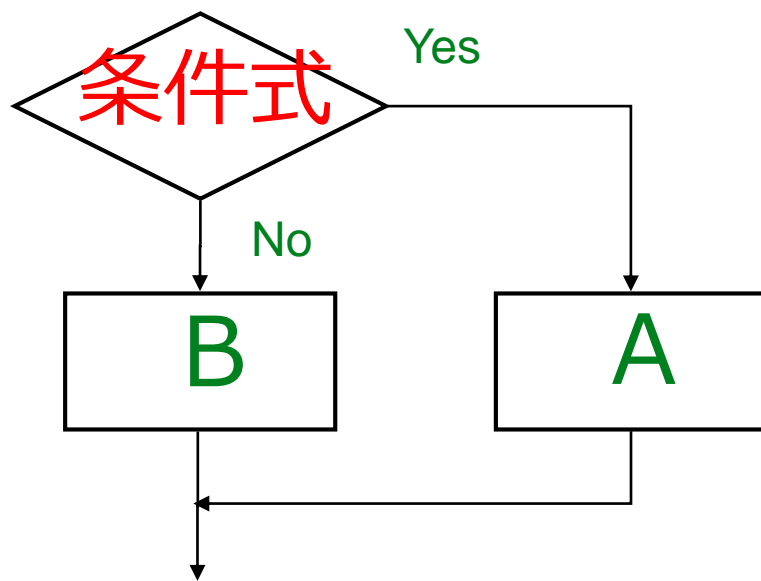


- 「条件式」が成り立てばAを、成り立たなければBを実行

```
if 条件式 then begin
  文;
  文;
  ...
end
else begin
  文;
  文;
  ...
end;
```

A

B



if ... else 文



「if ... then begin ... end else begin ... end;」で
1つの文であり、**文の末端**には文の区切りとして、
セミコロンが必要

```
if 条件式 then begin
```

```
  文;
```

```
  文;
```

```
  ...
```

```
end
```

```
else begin
```

```
  文;
```

```
  文;
```

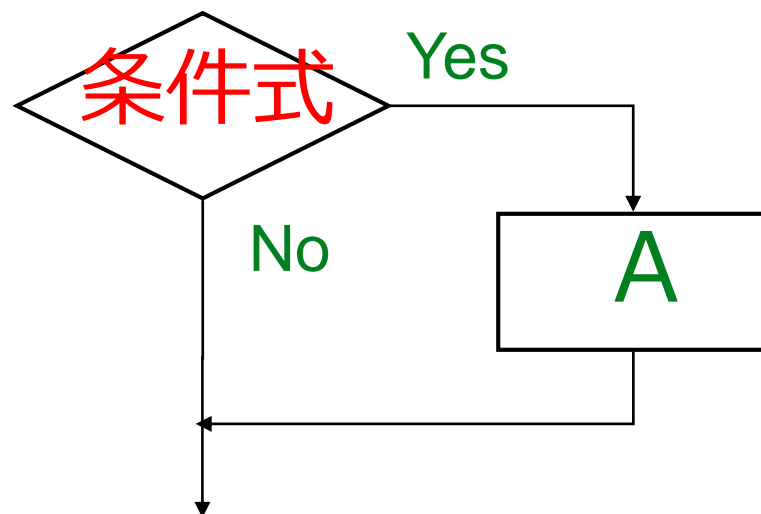
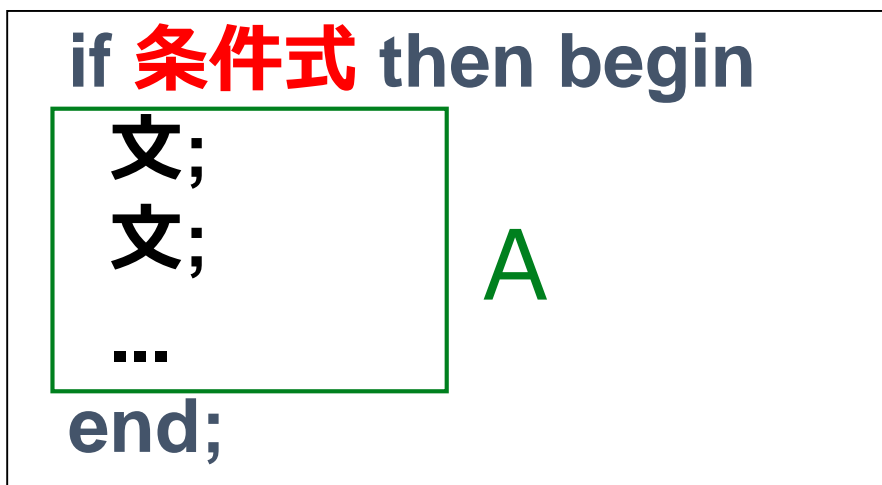
```
  ...
```

```
end;
```


if 文



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



if 文



「if ... then begin ... end;」 で 1つの文であり, 文の末端には文の区切りとして, セミコロンが必要

```
if 条件式 then begin
```

```
  文;
```

```
  文;
```

```
  ...
```

```
end;
```



- **比較演算**は、条件式を書くときなどに使用

- 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



```
if age >= 20 then begin
  writeln('成人です');
end
else begin
  writeln('未成年です');
end;
```

「左辺が右辺以上」の意味

例題 2. 対数の計算



- 浮動小数点数データを読み込んで、**e を底とする対数の計算と表示**を行うプログラムを作る。
 - 但し、**0 または負の数の場合**には、「計算できないこと」を意味する**メッセージを表示する**
 - 正のとき： $\ln()$ を使って計算できる
 - 0 または負のとき： 対数は計算できない
- **負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。



- a を底とする指数関数

$$y = a^x$$

- a を底とする y の対数

$$x = \log_a y \Leftrightarrow y = a^x$$



```
program sum;
```

```
var x: real;
```

```
begin
```

```
  write('Please Enter x: ');
```

```
  readln(x);
```

```
  if x <= 0 then begin
```

```
    writeln('x <= 0, then x is out of range');
```

```
  end
```

```
  else begin
```

```
    writeln('ln(', x:8:3, ') =', ln(x):8:3);
```

```
  end;
```

```
  readln
```

```
end.
```

条件式

条件が成り立つ
場合に実行され
る部分

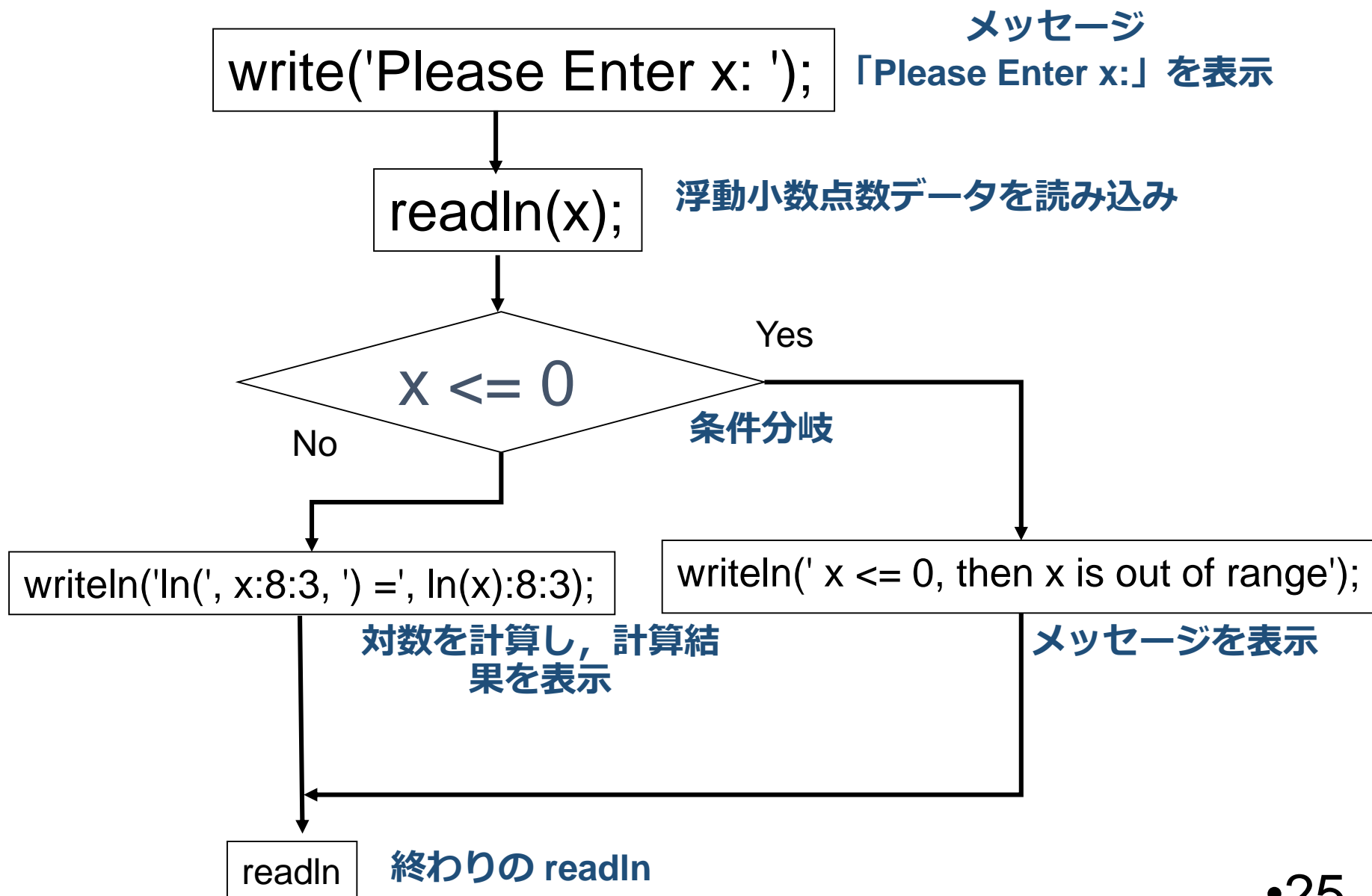
条件が成り立た
ない場合に実行
される部分

実行結果の例

```
Please Enter x: 1  
ln( 1.000) = 0.000
```

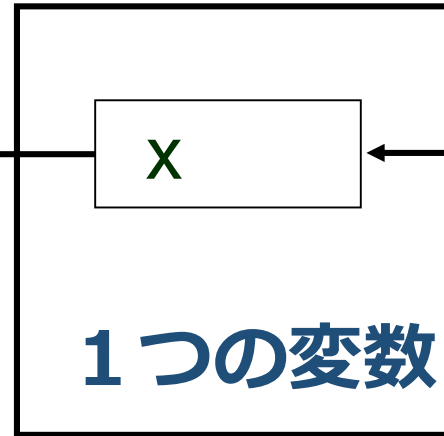
```
Please Enter x: 0  
x <= 0, then x is out of range
```


プログラム実行順





メモリ



①

```
readln(x);
```

浮動小数点数データを
読み込み

③

```
writeln('ln(', x:8:3, ') =', ln(x):8:3);
```

計算と表示

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 begin
  文;
  文;
  ...
end;
```

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

```
if 条件式 begin
  文;
  文;
  ...
end
else begin
  文;
  文;
  ...
end;
```

セミコロン

セミコロン



例題 3 . 定形郵便物の料金

- キーボードから数値を読み込んで,

25 以下なら	「84 Yen」
25 より大きくて50以下なら	「94 Yen」
50より大きいなら	「Too heavy」

と表示するプログラムを作る
- **多分岐**を行うために, **if 文**を使う



```
program sum;  
var x: real;  
begin  
  write('juuryou?: ');  
  readln(x);  
  if x <= 25 then begin  
    writeln('84 Yen');  
  end  
  else if x <=50 then begin  
    writeln('94 Yen');  
  end  
  else begin  
    writeln('Too heavy');  
  end;  
  readln  
end.
```

**x <= 25 が成り立つ
場合に実行される部分**

**x <= 25 が成り立たず,
x <= 50 が成り立つ
場合に実行される部分**

**x <= 25 も成り立たず,
x <= 50 も成り立たない
場合に実行される部分**

セミicolon

end.

定形郵便物の料金



実行結果の例

```
juuryou?: 20  
84 Yen
```

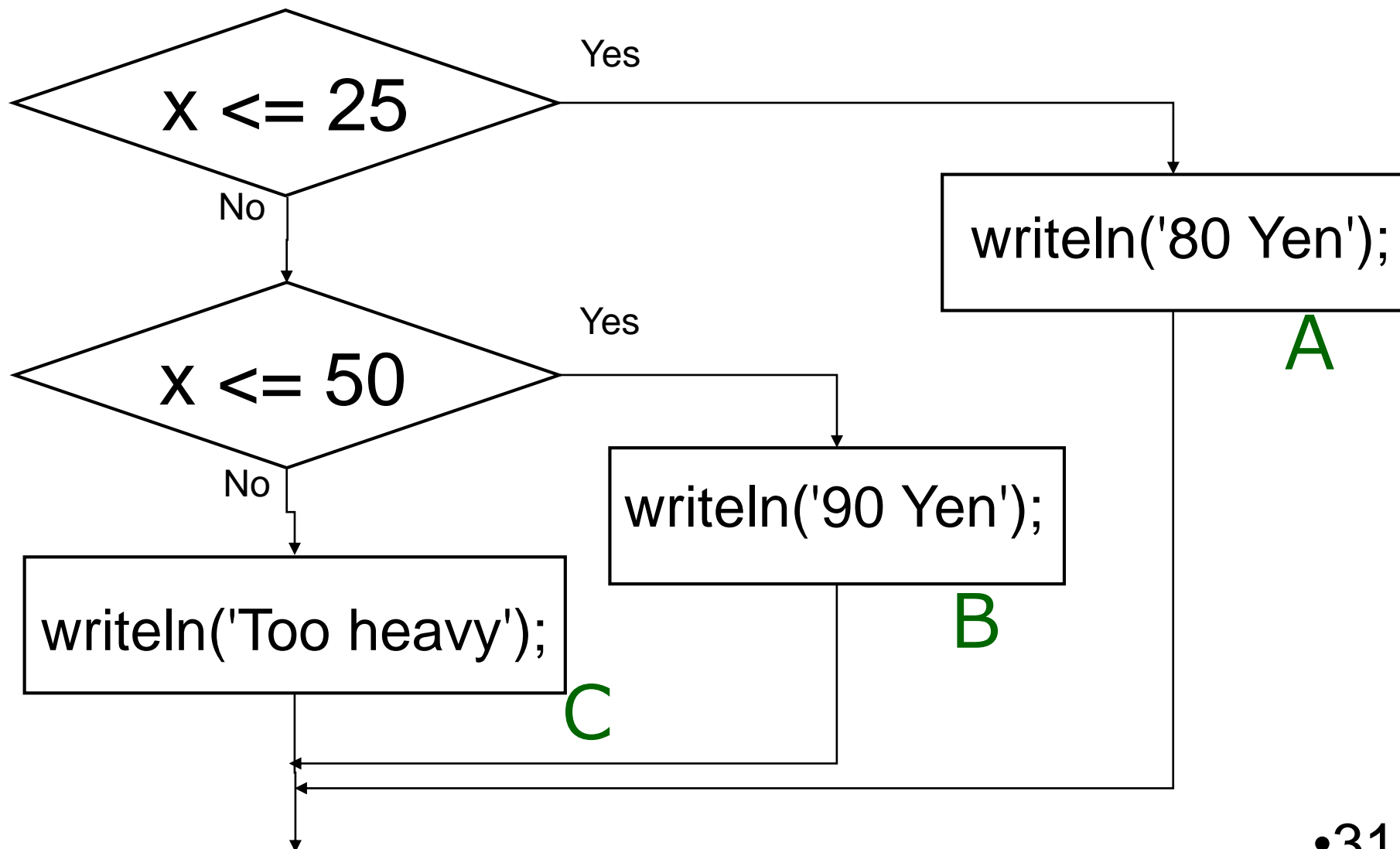
```
juuryou?: 40  
94 Yen
```

```
juuryou?: 60  
Too heavy
```

定形郵便物の料金



A, B, C のうちどれか 1 つを実行





```
program sum;  
var x: real;  
begin  
  write('juuryou?: ');  
  readln(x);  
  if x <= 25 then begin  
    writeln('80 Yen');  
  end  
  else if x <=50 then begin  
    writeln('90 Yen');  
  end  
  else begin  
    writeln('Too heavy');  
  end;  
  readln  
end.
```

このうちどれか1つ
が実行される

例題 4 . 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る
 - 重解, 虚数解も正しく求めるために, 判別式 $b^2 - 4ac$ の値で条件分岐する
 - この例題 4 では, 「 $a=0$ 」の場合は考えない



$D = b^2 - 4ac$ とする

1) 判別式 $D > 0$ のとき

$$x = \frac{-b + \sqrt{D}}{2a}, \frac{-b - \sqrt{D}}{2a}$$

異なる 2 実数解

2) $D = 0$ のとき

$$x = -\frac{b}{2a},$$

重解

3) $D < 0$ のとき

$$x = \frac{-b + i\sqrt{-D}}{2a}, \frac{-b - i\sqrt{-D}}{2a}$$

異なる 2 虚数解



```
program sum;
var a, b, c, D: real;
begin
  write('Please Enter a: ');
  readln(a);
  write('Please Enter b: ');
  readln(b);
  write('Please Enter c: ');
  readln(c);
  D := b * b - 4 * a * c;
  if D > 0 then begin
    writeln('x =', (- b + sqrt(D))/ (2*a):8:3, ', ', (- b - sqrt(D))/ (2*a):8:3 );
  end
  else if D = 0 then begin
    writeln('x =', - b / (2*a):8:3 );
  end
  else begin
    writeln( 'x =', - b / ( 2*a ):8:3, ' + ', sqrt(-D) / (2*a):8:3, 'i ',
      - b / (2*a):8:3, ' - ', sqrt(-D) / (2*a):8:3, 'i');
  end;
  readln
end.
```

$D > 0$ のとき

$D = 0$ のとき

$D < 0$ のとき

2次方程式



実行結果の例

```
Please Enter a: 1
Please Enter b: 5
Please Enter c: 6
x = -2.000, -3.000
```

```
Please Enter a: -2
Please Enter b: -8
Please Enter c: -8
x = -2.000
```

```
Please Enter a: 1
Please Enter b: 1
Please Enter c: 1
x = -0.500 + 0.866i , -0.500 - 0.866i
```

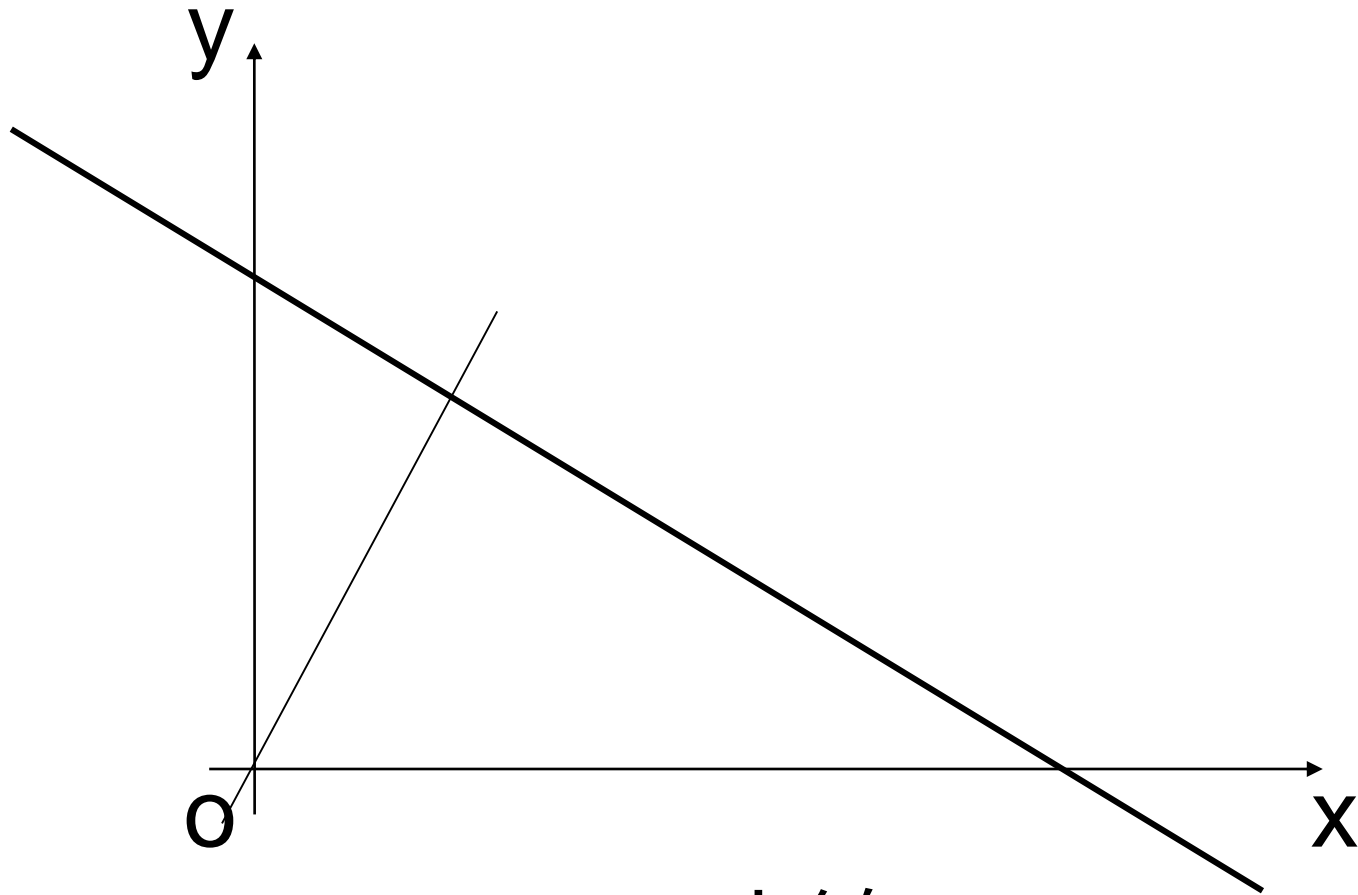
例題 5. 直線と原点の距離



- 平面 : $ax + by + c = 0$ と原点との距離を求めるプログラムを作成する
- 但し, $a = 0$ かつ $b = 0$ のときには計算を行わない

$$d = \frac{|c|}{\sqrt{a^2 + b^2}}$$

直線と原点の距離



直線 $ax + by + c = 0$



```
program sum;
var a, b, c, d: real;
begin
  write('Please Enter a: ');
  readln(a);
  write('Please Enter b: ');
  readln(b);
  write('Please Enter c: ');
  readln(c);
  if ( a = 0 ) and ( b = 0 ) then begin
    writeln('a=0 and b=0!');
  end
  else begin
    d := abs( c ) / sqrt( a * a + b * b );
    writeln('d =', d:8:3 );
  end;
  readln
end.
```

**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**

直線と原点の距離



実行結果の例

```
Please Enter a: 1  
Please Enter b: 1  
Please Enter c: 1  
d = 0.707
```

```
Please Enter a: 0  
Please Enter b: 0  
Please Enter c: 1  
a=0 and b=0!
```


条件式



$(a = 0) \text{ and } (b = 0)$



a=0 である

b=0 である



かつ

論理演算



- A and B A かつ B
- A or B A または B
- not A A でない

真, 偽に関する論理的な演算を行う.

演習1. 論理式に関する演習



(1) 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

例題4のプログラムを書き換えて、 **$a=0$ の場合にも、正しく解を求める**ようにしなさい

(2) 平面 $ax + by + cz + d = 0$ と原点の距離を求めるプログラムを作りなさい



演習 (1) で行うこと

- $a = 0$ かつ $b = 0$ かつ $c = 0$ のとき
すべての x が解である
- $a = 0$ かつ $b = 0$ かつ $c \neq 0$ のとき
解なし
- $a = 0$ かつ $b \neq 0$ のとき
 $x = -c / b$



演習 (1) で行うこと

- $a = 0$ かつ $b \neq 0$ のとき

$$x = -c / b$$

この部分のプログラム例 (正解は1つでは無い)

```
else if ( a = 0 ) and ( b <> 0 ) then begin
    writeln ('x =', - c / b:8:3 );
end
```