



pe-5. 繰り返し計算

(Pascal プログラミング入門)

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



金子邦彦



内容



- 例題 1. 自然数の和
- 例題 2. 最大公約数の計算
- 例題 3. ベクトルの長さ

while 文

- 例題 4. 九九の表

for 文と繰り返しの入れ子

- 例題 5. ド・モアブルの公式

計算誤差の累積

目標

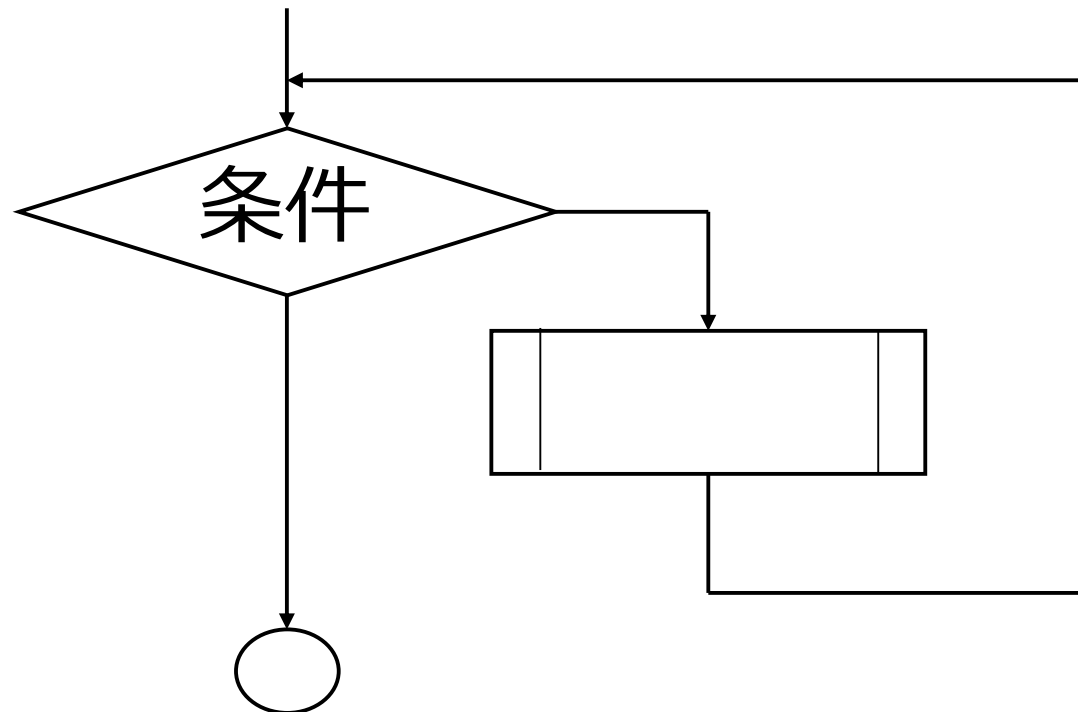


- **繰り返し**（**while 文, for 文**）を使って，**繰り返し計算**を行えるようになること
- **ループカウンタ**として，**整数の変数**を使うこと
- 見やすいプログラムを書くために，**字下げ**を行う

繰り返しとは



- **繰り返し**とは、ある条件が満たされるまで、同じことを繰り返すこと。
- **繰り返し**を行うための文として**while文**, **for 文**などがある。



繰り返しの例



- ユークリッドの互助法
 - m と n の最大公約数を求めるために、「割った余りを求めること」を、余りが 0 になるまで繰り返す。
- 九九の表
 - 九九の表を求めるために、掛け算を 81 回繰り返す



- 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 main content area.



③ 「Language」 のところで、「Pascal」 を選ぶ

The screenshot shows the GDB Online web 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 it, a list of programming languages is displayed, with 'Pascal' highlighted by a red box. The main area shows a code editor with a C program 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 . 自然数の和

- 整数データ（Nとする）を読み込んで、**1からNまでの和を求めるプログラムを作る**
- ここでは、練習のため、自然数の和の公式は使わずに、**while文を用いる**

例) 100 → 5050



```
program sum;  
var N, i, s: integer;  
begin  
  write('sum [1..N] Program. Please Enter N: ');  
  readln(N);  
  s := 0;  
  i := 1;  
  while i <= n do begin  
    s := s + i;  
    i := i + 1;  
  end;  
  writeln('sum[1..n] = ', s:8);  
  readln  
end.
```

条件式

**繰り返し実行される
部分**

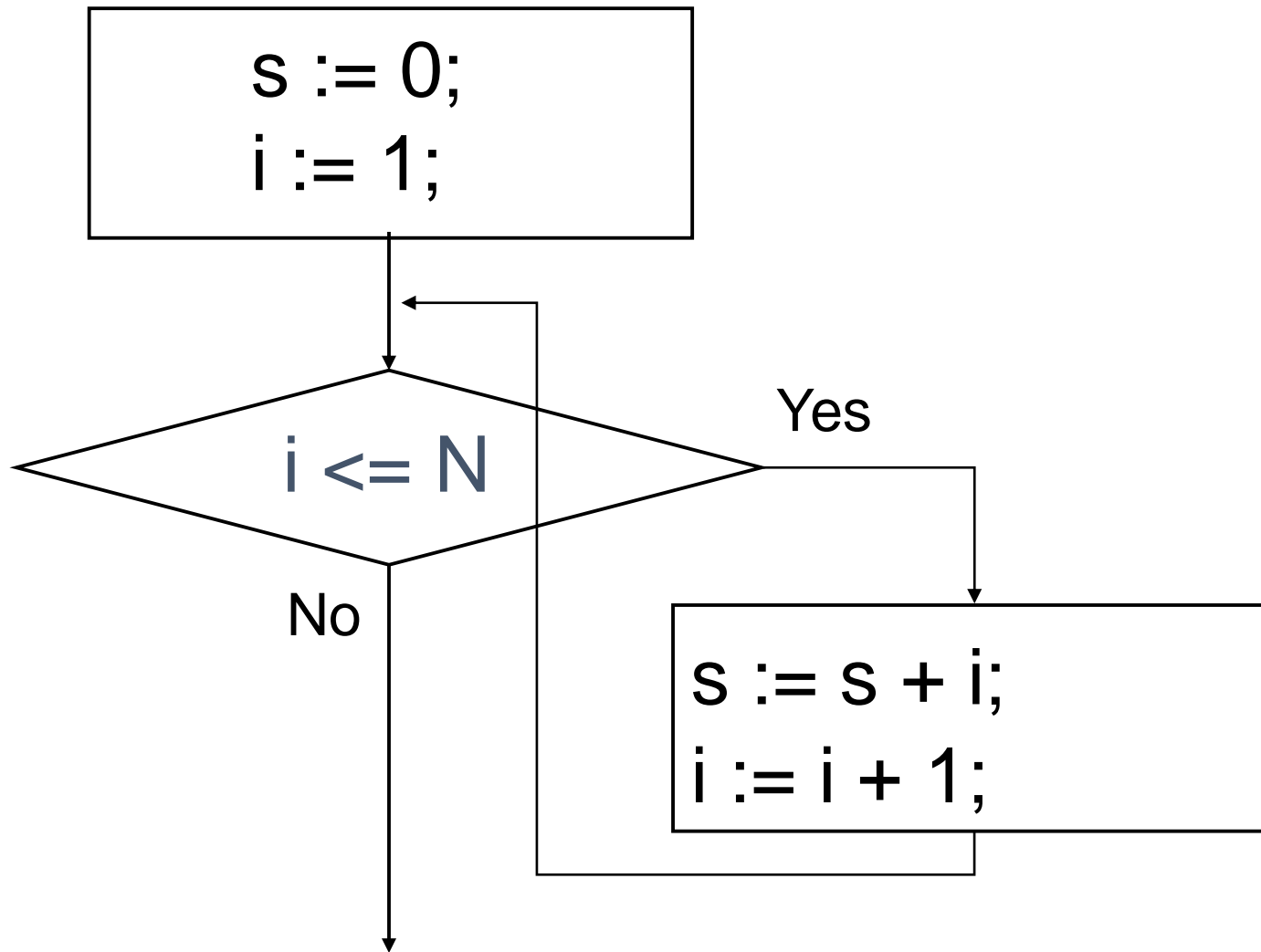


実行結果の例

```
sum [1..N] Program. Please Enter N: 100  
sum[1..n] =      5050
```

```
sum [1..N] Program. Please Enter N: 200  
sum[1..n] =     20100
```

プログラム実行順



繰り返し処理の中身



- **繰り返しの前**

$i := 1$ と $S := 0$ を実行

- **繰り返しの各ステップ**でなされること

1. 「S」に「i」を足しこむ

→ 「S」には、その時点での「1からi」までの和が入る

2. 「i」の値を1増やす

- **繰り返しの終了条件**

$i \leq N$ が成り立たなくなったら終了

→ つまり $i > N$ になったら終了

自然数の和



N = 7 とすると

s の値

i の値

はじめは $s = 0$

$i = 1$

繰り返し 1回目	$i \leq 7$ が成立する	$s = 0 + 1$	$i = 2$
繰り返し 2回目	$i \leq 7$ が成立する	$s = 1 + 2$	$i = 3$
繰り返し 3回目	$i \leq 7$ が成立する	$s = 3 + 3$	$i = 4$
繰り返し 4回目	$i \leq 7$ が成立する	$s = 6 + 4$	$i = 5$
繰り返し 5回目	$i \leq 7$ が成立する	$s = 10 + 5$	$i = 6$
繰り返し 6回目	$i \leq 7$ が成立する	$s = 15 + 6$	$i = 7$
繰り返し 7回目	$i \leq 7$ が成立する	$s = 21 + 7$	$i = 8$
繰り返し 8回目	$i \leq 7$ が成立しない		

while 文



- 何かの処理の**繰り返し**
- **繰り返し**のたびに while 文で書かれた**条件式の真偽**が判定され, **真である限り**, while のあとに続く文が**実行され続ける**.

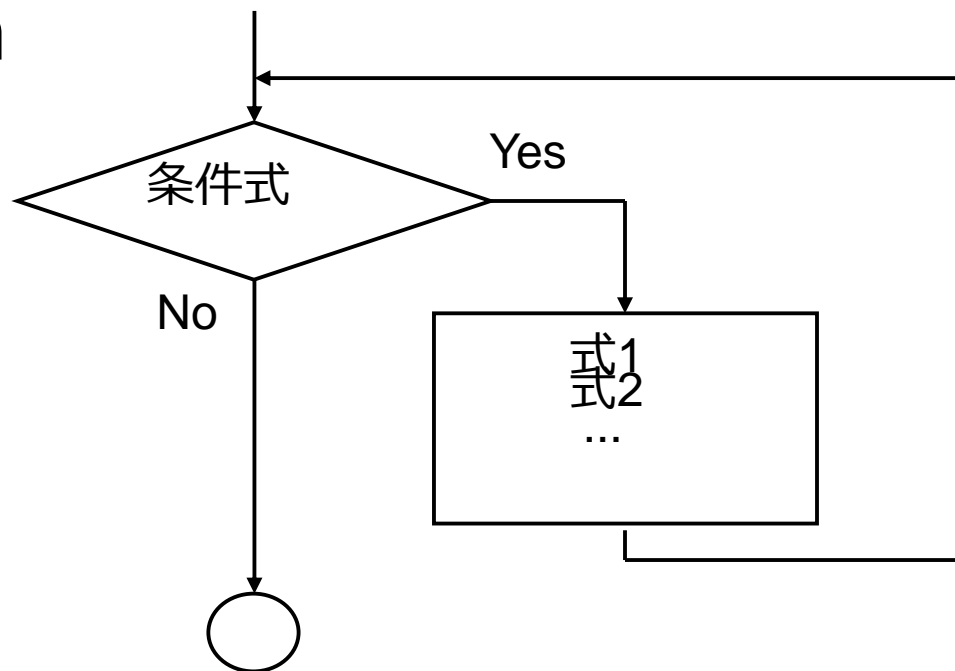
```
while 条件式 do begin
```

```
  式1;
```

```
  式2;
```

```
  ...
```

```
end
```



例題 2. 最大公約数の計算



- 2つの整数データを読み込んで、**最大公約数を求めるプログラム**を作る。
 - ユークリッドの互助法を用いること
 - ユークリッドの互助法を行うために while 文を書く
- 例) 20, 12 のとき : 4

ユークリッドの互助法

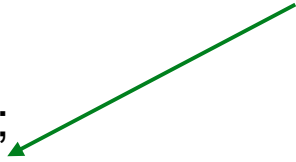


- **最大公約数を求めるための手続き**
- m, n の最大公約数は,
 - $m \geq n$ とすると,
 - 「 m を n で割った余り」 = 0 なら, 最大公約数は n
 - 「 m を n で割った余り」 > 0 なら, m と n の最大公約数は, 「 m を n で割った余り」と n の最大公約数に等しい
(なお, $n >$ 「 m を n で割った余り」 が成り立つ)



```
program sum;  
var r, m, n: integer;  
begin  
    write('GCD Program. Please Enter m: ');  
    readln(m);  
    write('Please Enter n: ');  
    readln(n);  
    r := m mod n;  
    while r > 0 do begin  
        m := n;  
        n := r;  
        r := m mod n;  
    end;  
    writeln('GCD = ', n:8);  
    readln  
end.
```

条件式



条件が成り立つ限り,
実行されつづける部分

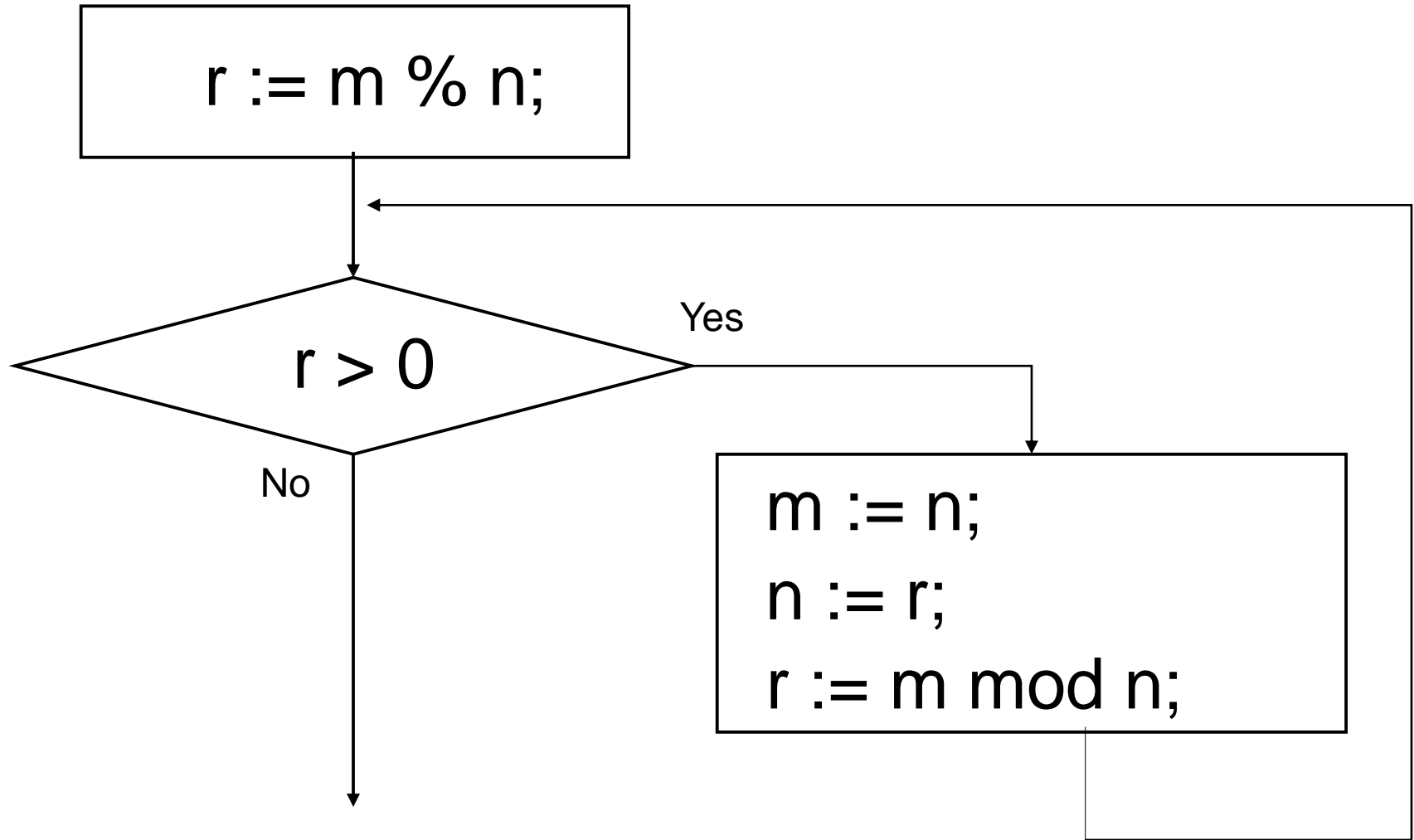
最大公約数の計算



実行結果の例

```
BCD Program. Please Enter m: 80  
Please Enter n: 35  
BCD =          5
```

プログラム実行順



繰り返し処理の中身



- **繰り返しの前**

$r := m \bmod n$ を実行 (m を n で割った余りが r に入る)

- **繰り返しの各ステップ**でなされること

$m := n;$

$n := r;$

$r := m \bmod n;$

を実行 (m, n, r の値は小さくなっていく)

- **繰り返しの終了条件**

r が 0 になったら終了

最大公約数の計算



$m = 80, n = 35$ とすると,
最初の「 $r = m \bmod n;$ 」で, $r = 10$ になる

	m の値	n の値	r の値
繰り返し 1 回目	$r > 0$ が成立する $m = 35$ $n = 10$ $r = 5$		

80, 35 の最大公約数は
35, 10 の最大公約数に等しい

繰り返し 2 回目	$r > 0$ が成立する $m = 10$ $n = 5$ $r = 0$		
--------------	--	--	--

35, 10 の最大公約数は
10, 5 の最大公約数に等しい

繰り返し 3 回目	$r > 0$ が成立しない		
--------------	----------------	--	--

例題 3 . 総和と平均



- データ x_1, x_2, \dots, x_k を 1 つずつ読み込んで、**合計**と**平均を求めるプログラム**を作成す
- **負の数が入力されたら終了する**

整数のデータ 1, 2, 3 に対して

1	}	入力
2		
3		
-1		



```
program sum;  
var x, s: real;  
var i: integer;  
begin
```

```
  s := 0;
```

```
  i := 0;
```

```
  write('Please Enter x[', i:3, ']: ');
```

```
  readln(x);
```

```
  while 0 <= x do begin
```

```
    s := s + x;
```

```
    i := i + 1;
```

```
    write('Please Enter x[', i:3, ']: ');
```

```
    readln(x);
```

```
  end;
```

```
  writeln('sum =', s:8:3);
```

```
  writeln('average =', (s/i):8:3);
```

```
  readln
```

```
end.
```

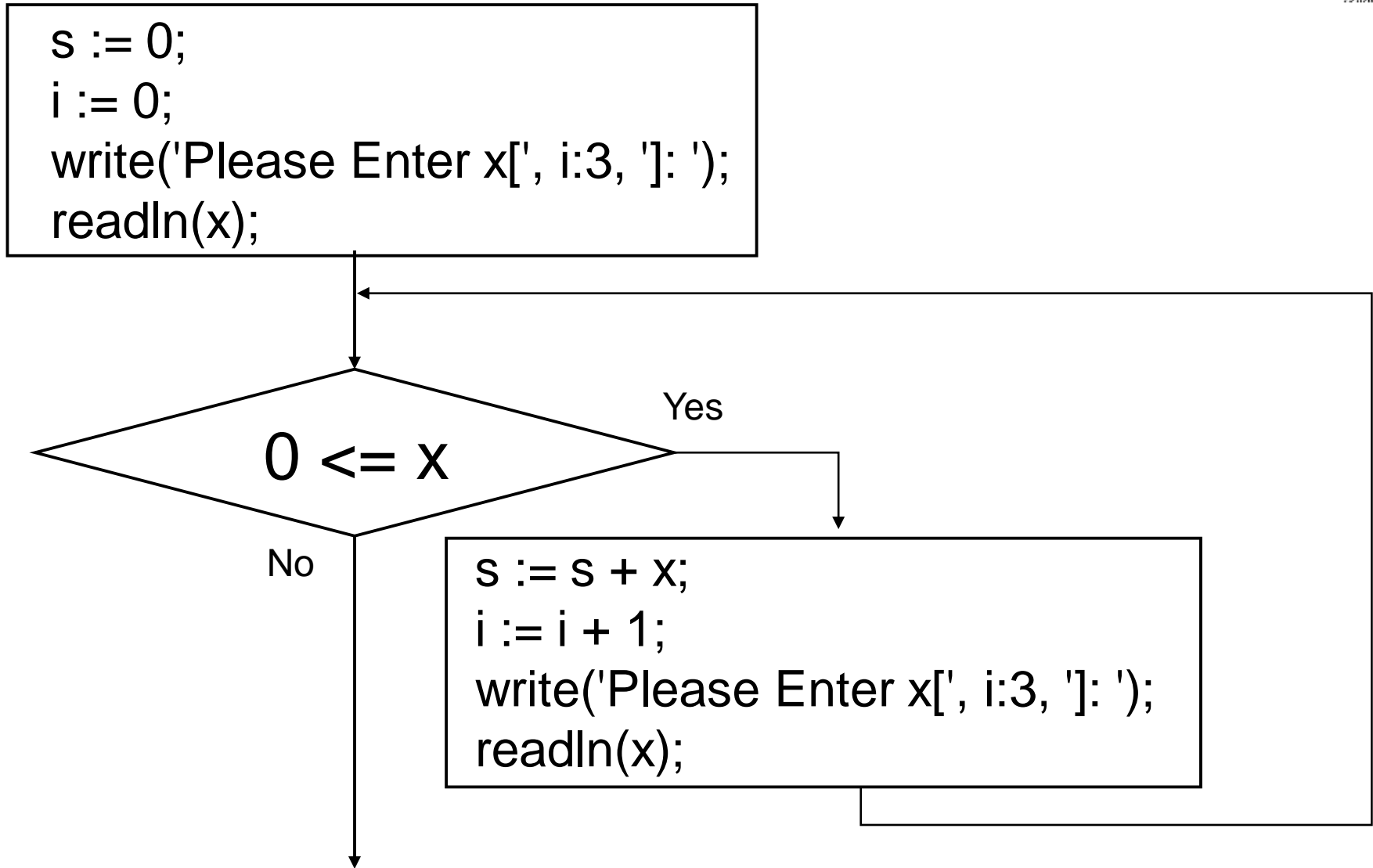
条件式

条件が成り立つ限り,
実行されつづける部分

実行結果の例

```
Please Enter x[ 0]: 1
Please Enter x[ 1]: 2
Please Enter x[ 2]: 3
Please Enter x[ 3]: -1
sum =      6.000
average =   2.000
```

プログラム実行順



繰り返し処理の中身



- **繰り返しの前**

$S := 0$ と $i := 0$ を実行

X_0 を読み込む

- **繰り返しの各ステップ**でなされること

1. 「S」 に 「 X_i 」 を足しこむ

→ 「S」 には, その時点での「 X_0 から X_i 」までの和が入る

2. 「i」 の値を1増やす

3. X_i を読み込む

- **繰り返しの終了条件**

$X_i < 0$ ならば終了

演習 1 . m から n までの和



- 2つの整数データ (M, Nとする) を読み込んで, MからNまでの和を求めるプログラムを作りなさい
 - while 文を使いなさい. 例題 1 のプログラムを参考にしなさい

例題 4 . 九九の表



- 九九の表を表示するプログラムを作成する
 - 九九の表を表示するために、繰り返しの入れ子を使う



```
program sum;  
uses SysUtils;  
var i, j: integer;  
begin
```

繰り返し実行される部分

```
  for i := 1 to 9 do begin  
    write( i:3, ':' );  
    for j := 1 to 9 do begin  
      write( (i*j):3 );  
    end;  
    writeln;
```

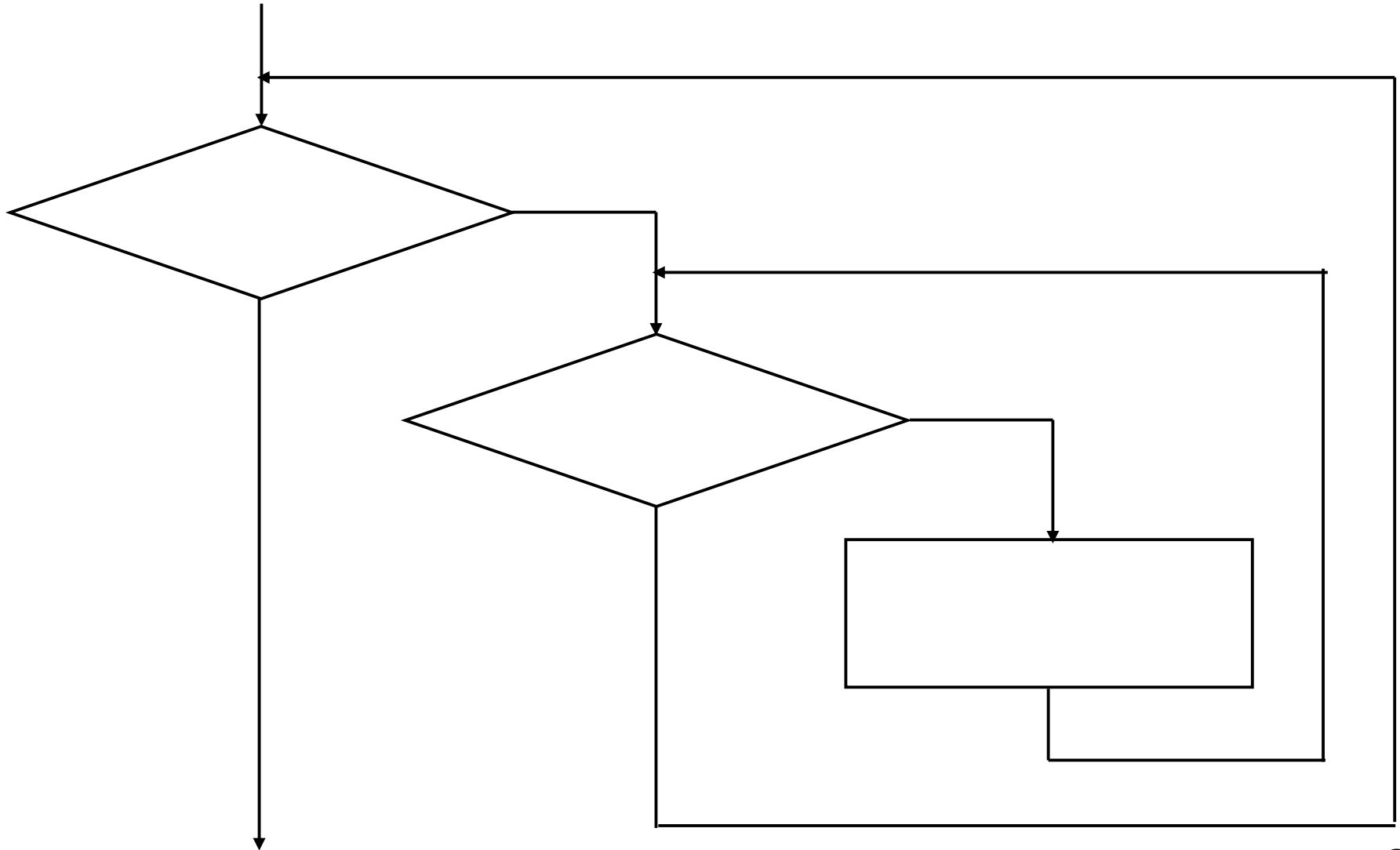
```
  end;  
  readln  
end.
```

実行結果画面 (例)

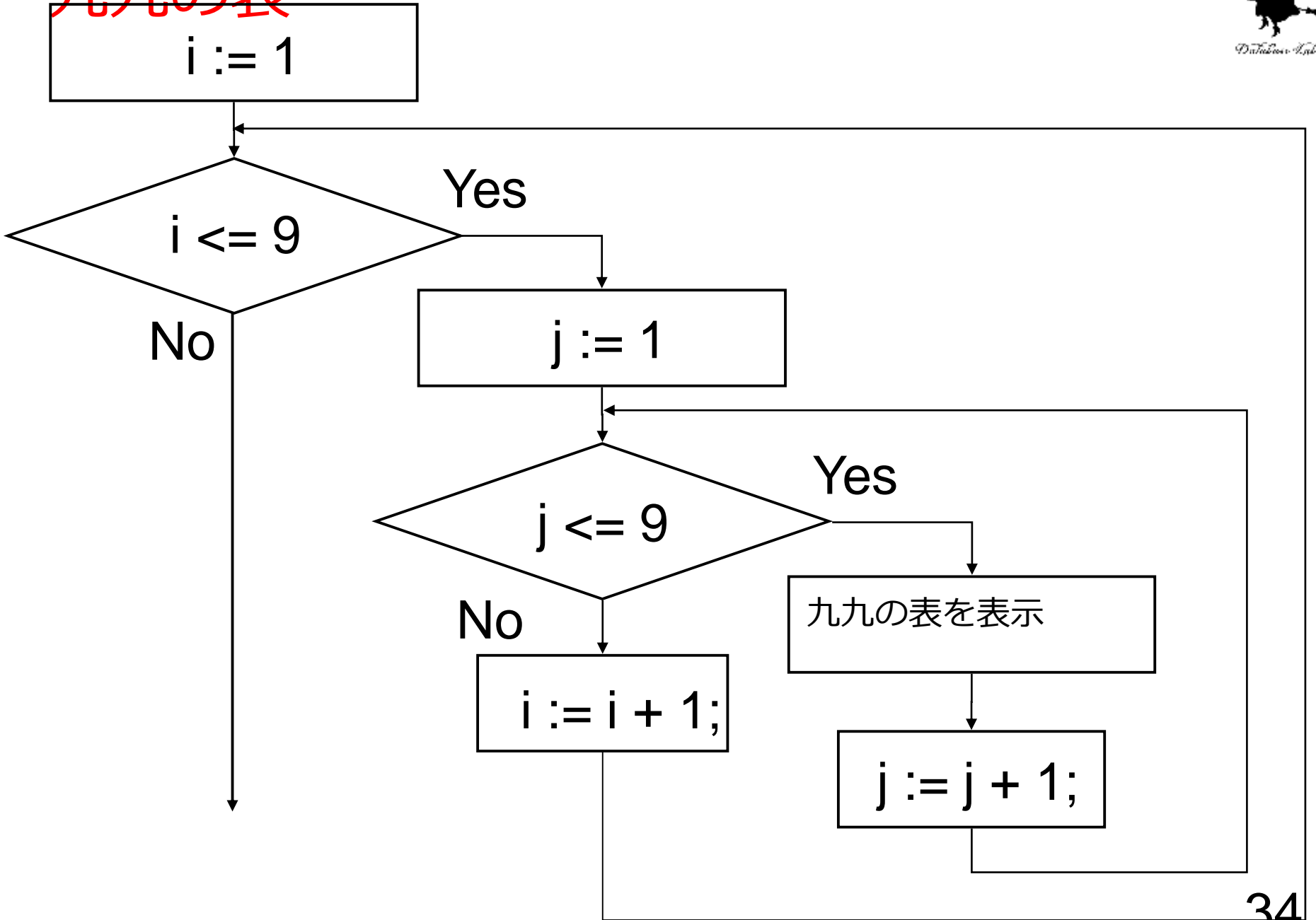


```
1:  1  2  3  4  5  6  7  8  9
2:  2  4  6  8 10 12 14 16 18
3:  3  6  9 12 15 18 21 24 27
4:  4  8 12 16 20 24 28 32 36
5:  5 10 15 20 25 30 35 40 45
6:  6 12 18 24 30 36 42 48 54
7:  7 14 21 28 35 42 49 56 63
8:  8 16 24 32 40 48 56 64 72
9:  9 18 27 36 45 54 63 72 81
```


繰り返しの入れ子



九九の表





```
for 変数名 := 初期値 to 終了値 do begin  
    式1;  
    式2;  
    ...  
end
```

例題 5. ド・モアブルの定理



- θ を読み込んで、次の値を計算するプログラムを作る

$$(\cos\theta + i\sin\theta)^n$$

$$\cos n\theta + i\sin n\theta$$

- なお、 i は虚数単位
- ここでは $(\sin\theta + i\cos\theta)^n$ を求めるために、while文を用いた繰り返し計算を行ってみる

複素数の積



$$z_1 = x_1 + iy_1$$

$$z_2 = x_2 + iy_2 \text{ のとき}$$

$$\begin{aligned} z_1 z_2 &= (x_1 + iy_1)(x_2 + iy_2) \\ &= x_1x_2 + x_1iy_2 + iy_1x_2 + iy_1iy_2 \\ &= x_1x_2 - y_1y_2 + i(x_1y_2 + y_1x_2) \end{aligned}$$

$\underbrace{\hspace{10em}}_{\text{実数部}} \quad \underbrace{\hspace{10em}}_{\text{虚数部}}$



```
program sum;  
var j, n: integer;  
var x1, y1, x2, y2, theta: real;  
begin  
  write('Please Enter n: ');  
  readln(n);  
  write('Please Enter theta: ');  
  readln(theta);  
  x1 := cos(theta);  
  y1 := sin(theta);  
  x2 := x1;  
  y2 := y1;  
  j := 1;  
  while j <= n do begin  
    writeln( '(cos theta + i sin theta)', j, '=', x1:8:3, '+ i ', y1:8:3 );  
    writeln( 'cos', j, 'theta + i sin', j, 'theta =', cos( j * theta ):8:3, sin( j * theta ):8:3 );  
    x1 := x1 * x2 - y1 * y2;  
    y1 := x1 * y2 + x2 * y1;  
    j := j + 1;  
  end;  
  readln  
end.
```

条件式

繰り返し実行される
部分

実行結果画面 (例)



```
Please Enter n: 8
Please Enter theta: 0.1
(cos theta + i sin theta)1= 0.995+ i 0.100
cos1theta + i sin1theta = 0.995 0.100
(cos theta + i sin theta)2= 0.980+ i 0.197
cos2theta + i sin2theta = 0.980 0.199
(cos theta + i sin theta)3= 0.955+ i 0.292
cos3theta + i sin3theta = 0.955 0.296
(cos theta + i sin theta)4= 0.922+ i 0.382
cos4theta + i sin4theta = 0.921 0.389
(cos theta + i sin theta)5= 0.879+ i 0.468
cos5theta + i sin5theta = 0.878 0.479
(cos theta + i sin theta)6= 0.828+ i 0.548
cos6theta + i sin6theta = 0.825 0.565
(cos theta + i sin theta)7= 0.769+ i 0.622
cos7theta + i sin7theta = 0.765 0.644
(cos theta + i sin theta)8= 0.703+ i 0.689
cos8theta + i sin8theta = 0.697 0.717
```

繰り返し処理の中身



- 繰り返しの前

$x1 := \cos \theta$

$y1 := \sin \theta$

$x2 := x1$

$y2 := y1$

を実行

- 繰り返しの各ステップでなされること

$x1 := x1 * x2 - y1 * y2$

$y1 := x1 * y2 + x2 * y1;$

を実行 (x1 が**実数部**, y1 が**虚数部**)



$$(\cos\theta + i \sin\theta)^n = \cos n\theta + i \sin n\theta$$

$$\begin{aligned}(\cos\theta + i \sin\theta)^2 &= \cos^2\theta - \sin^2\theta + 2i \cos\theta \sin\theta \\ &= \cos 2\theta + i \sin 2\theta\end{aligned}$$

$$\begin{aligned}(\cos\theta + i \sin\theta)^3 &= (\cos\theta + i \sin\theta)^2 (\cos\theta + i \sin\theta) \\ &= (\cos 2\theta + i \sin 2\theta) (\cos\theta + i \sin\theta) \\ &= \cos 2\theta \cos\theta - \sin 2\theta \sin\theta \\ &\quad + i (\cos 2\theta \sin\theta - \sin 2\theta \cos\theta) \\ &= \cos (2\theta + \theta) + i \sin (2\theta + \theta) \\ &= \cos 3\theta + i \sin 3\theta\end{aligned}$$

(以下同様に考える. 数学的帰納法で証明できる)

計算結果から分かること



- 本来なら「 $(\cos\theta + i \sin\theta)^n = \cos n\theta + i \sin n\theta$ 」が成り立つはず
- しかし、コンピュータでの計算は、近似計算
- **計算を繰り返す**（つまり、計算結果を使った計算）
たびに、**誤差が積み重なる**