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

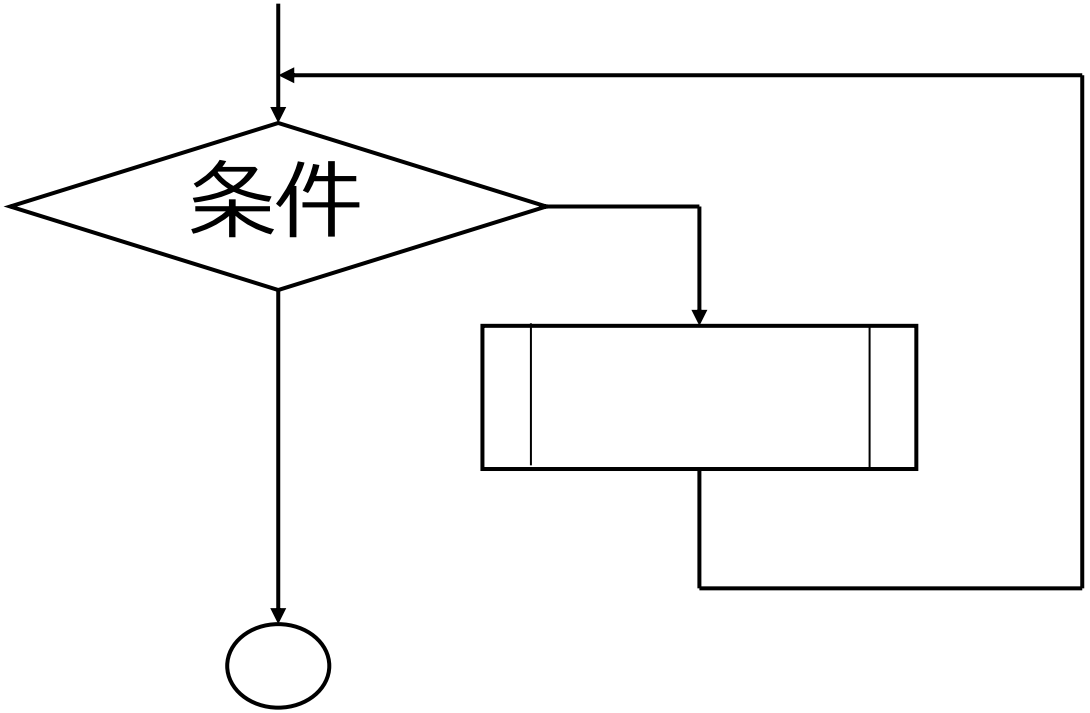
  計算誤差の累積

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

オンライン開発環境 Online GDB

- **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

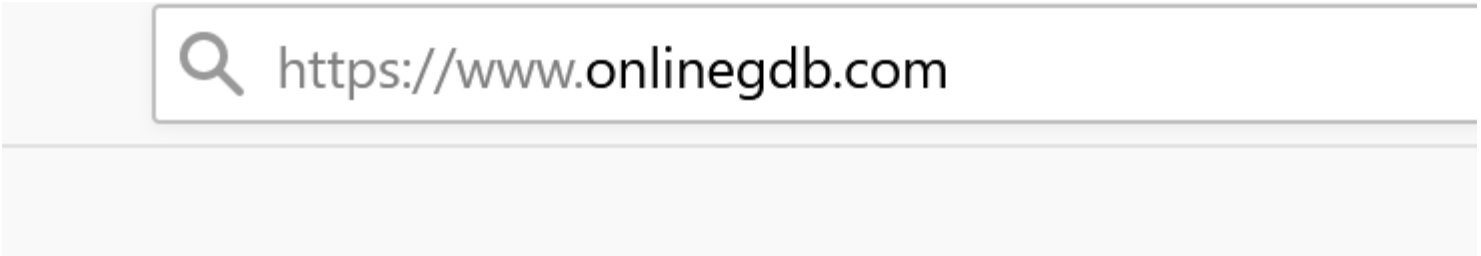
- オンラインなので、「秘密にしたいプログラム」を扱うには十分な注意が必要

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

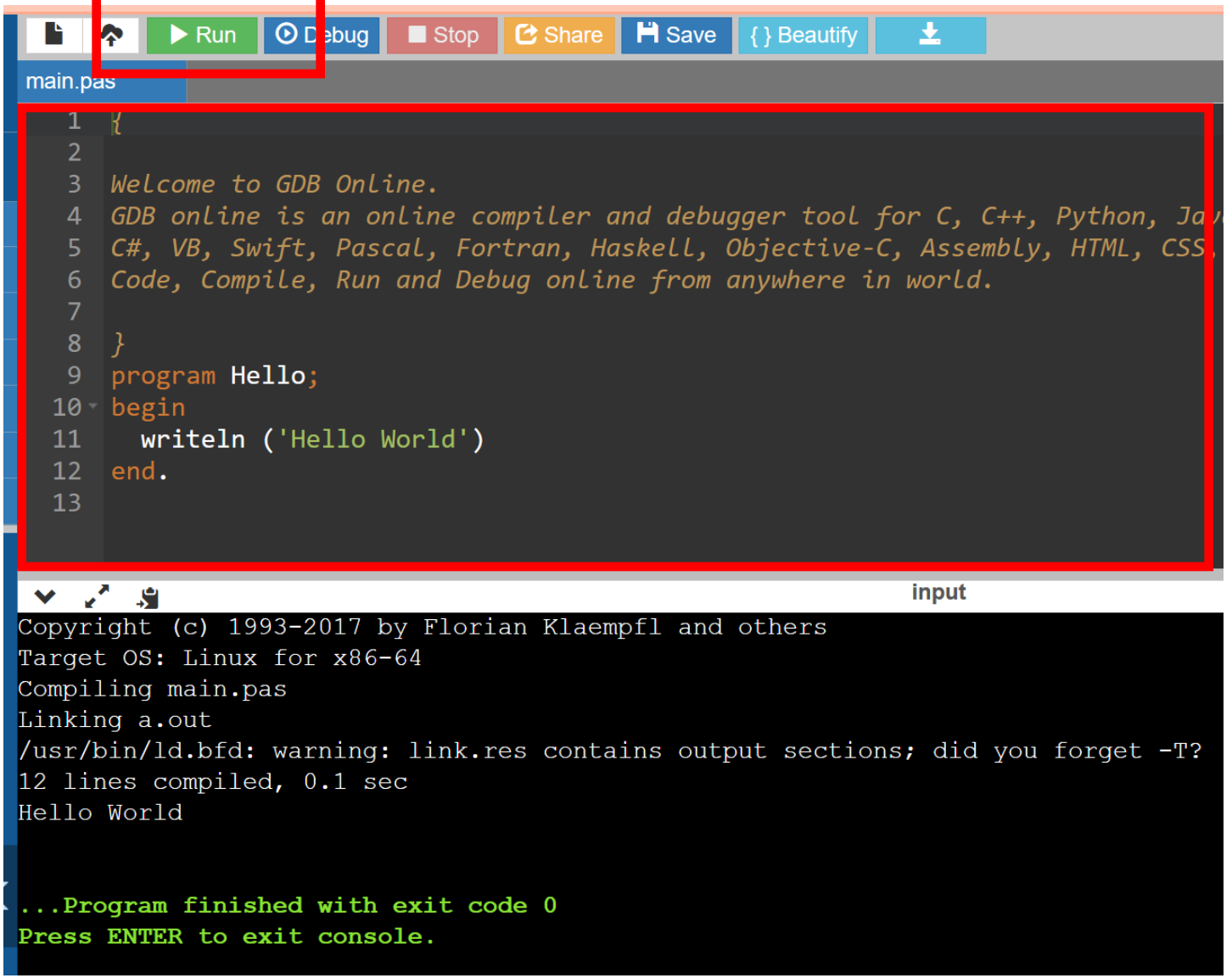


③「Language」のところで，「Pascal」を選ぶ

実行ボタン

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

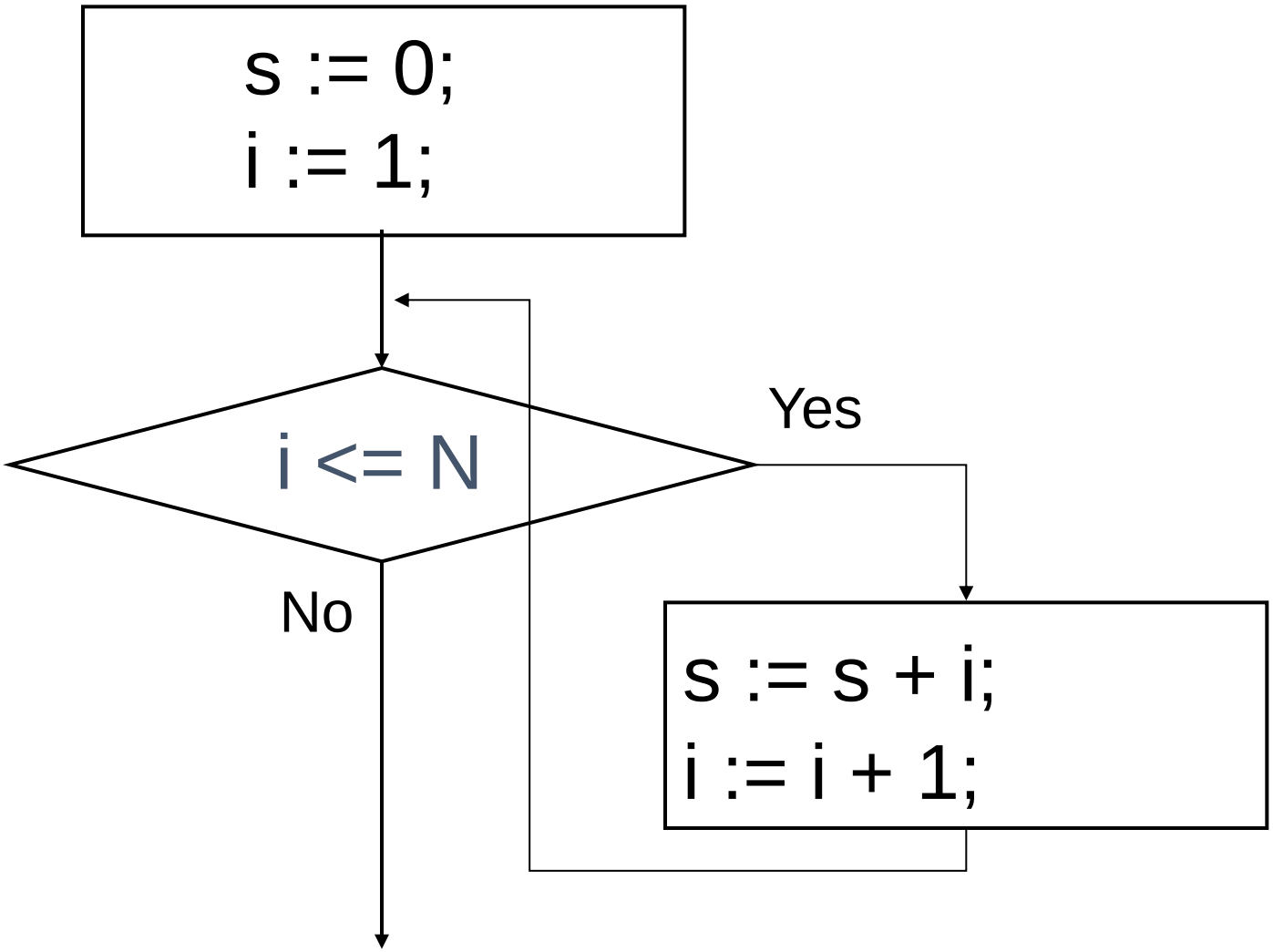
自然数の和

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 <= 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

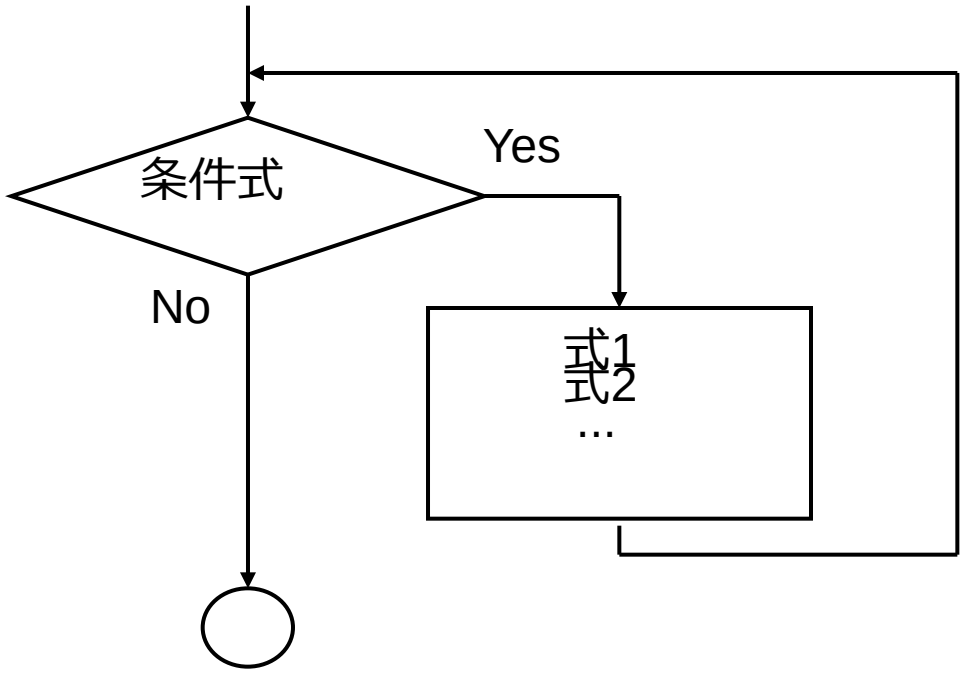
N = 7 とすると

		s の値	i の値
	はじめは s = 0		i = 1
繰り返し 1 回目	i <= 7 が成立する	s = 0 + 1	i = 2
繰り返し 2 回目	i <= 7 が成立する	s = 1 + 2	i = 3
繰り返し 3 回目	i <= 7 が成立する	s = 3 + 3	i = 4
繰り返し 4 回目	i <= 7 が成立する	s = 6 + 4	i = 5
繰り返し 5 回目	i <= 7 が成立する	s = 10 + 5	i = 6
繰り返し 6 回目	i <= 7 が成立する	s = 15 + 6	i = 7
繰り返し 7 回目	i <= 7 が成立する	s = 21 + 7	i = 8
繰り返し 8 回目	i <= 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 ≧ 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

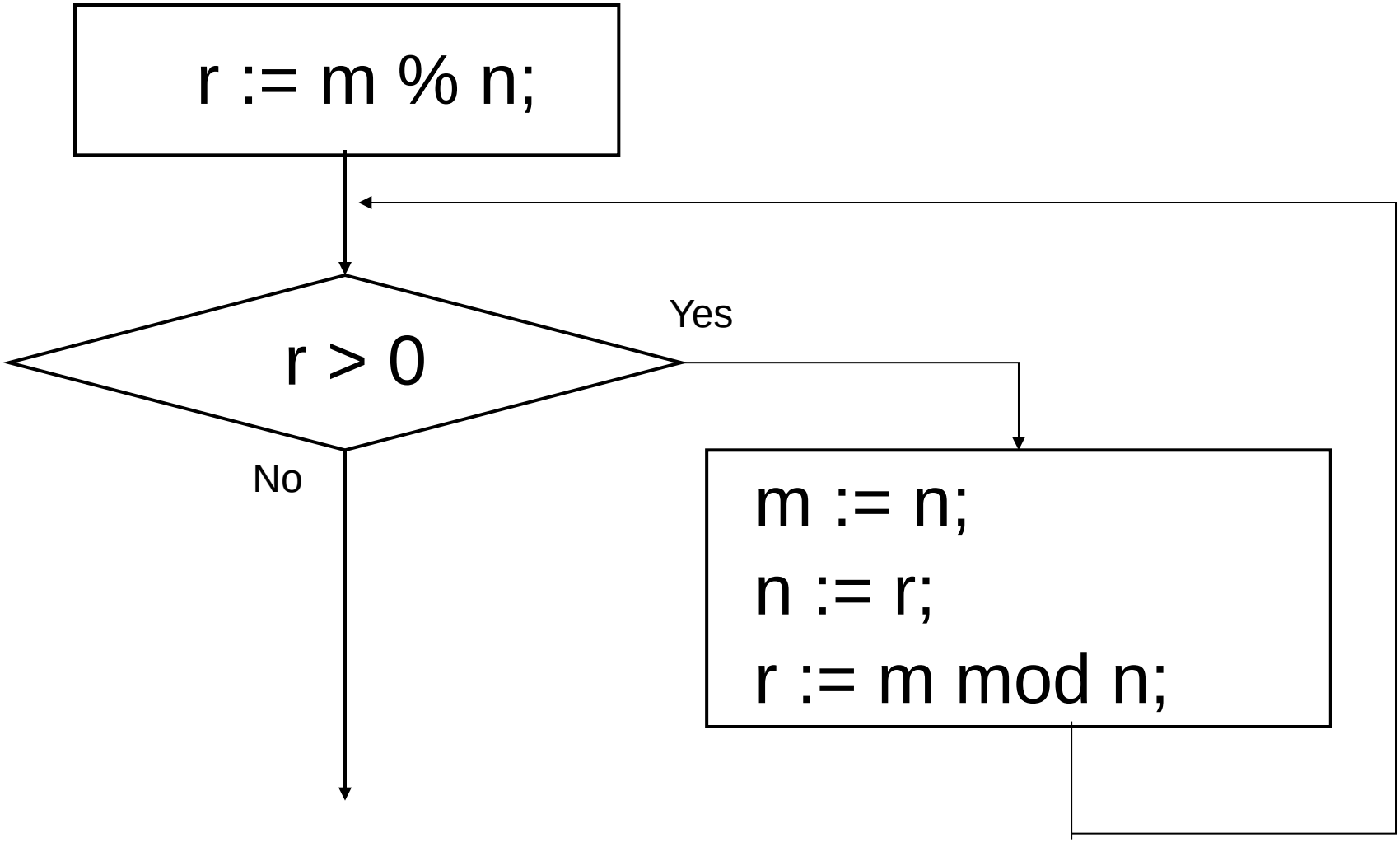
条件が成り立つ限り，
実行されつづける部分

最大公約数の計算

実行結果の例

```
GCD Program. Please Enter m: 80
Please Enter n: 35
GCD =         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 mod 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 mod 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 mod 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

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

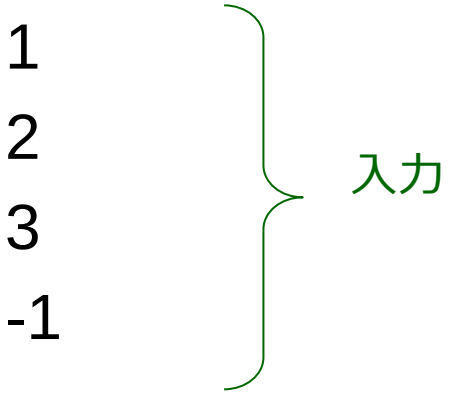
35, 10 の最大公約数は
10, 5 の最大公約数に等しい

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 x1, x2, ... xk を１つづつ読み込んで，**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



```
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

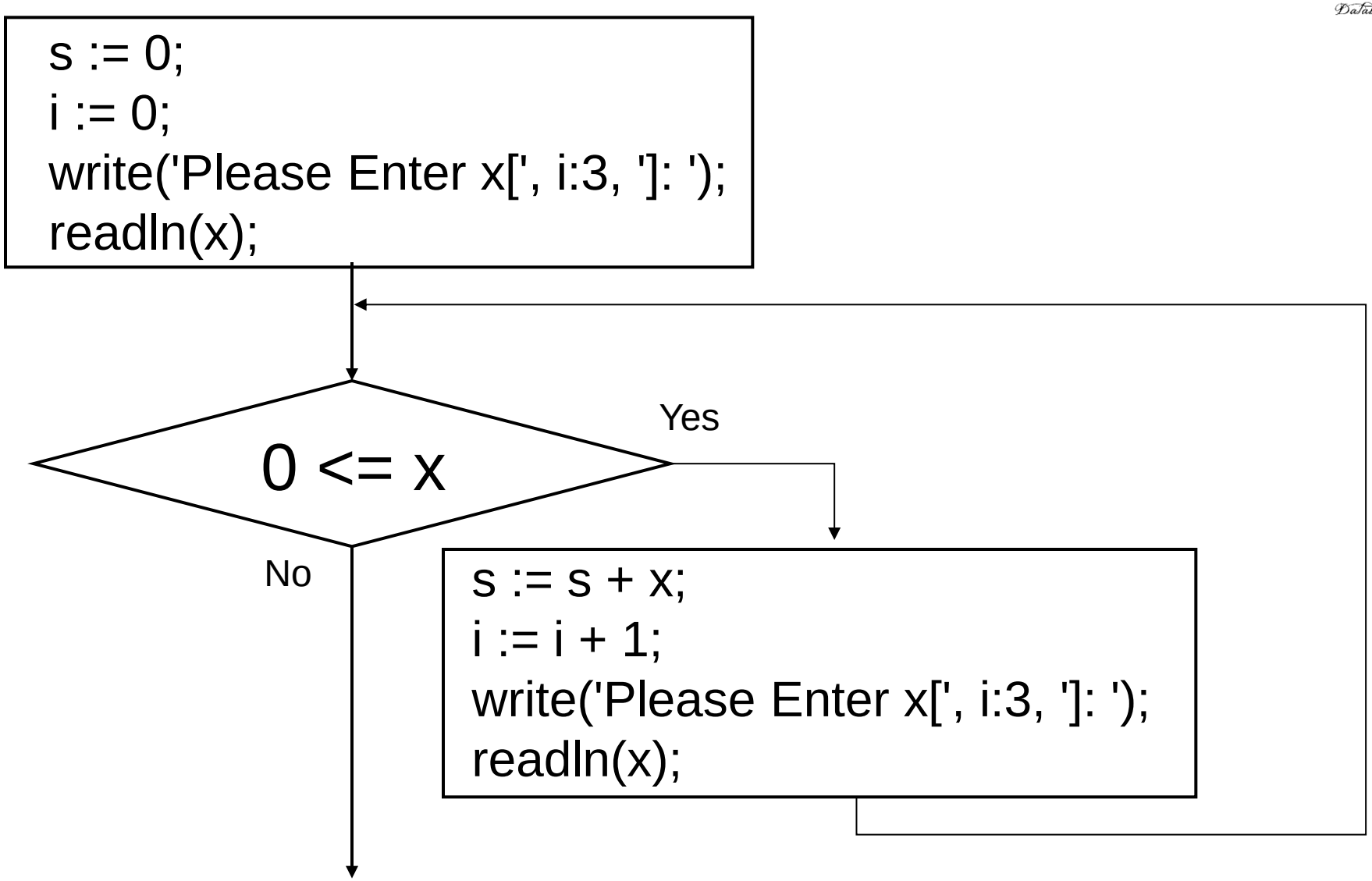
条件が成り立つ限り，
実行されつづける部分

総和と平均

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


s := 0;
i := 0;
write('Please Enter x[', i:3, ']: ');
readln(x);
0 <= x
Yes
No
s := s + x;
i := i + 1;
write('Please Enter x[', i:3, ']: ');
readln(x);

繰り返し処理の中身

- **繰り返しの前**

  S := 0 と i := 0 を実行

  X0 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 「Xi」 を足しこむ

     → 「S」には，その時点での「X0から Xi 」までの和が入る

  2. 「i」 の値を 1 増やす
  3. Xi を読み込む

- **繰り返しの終了条件**

  Xi < 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

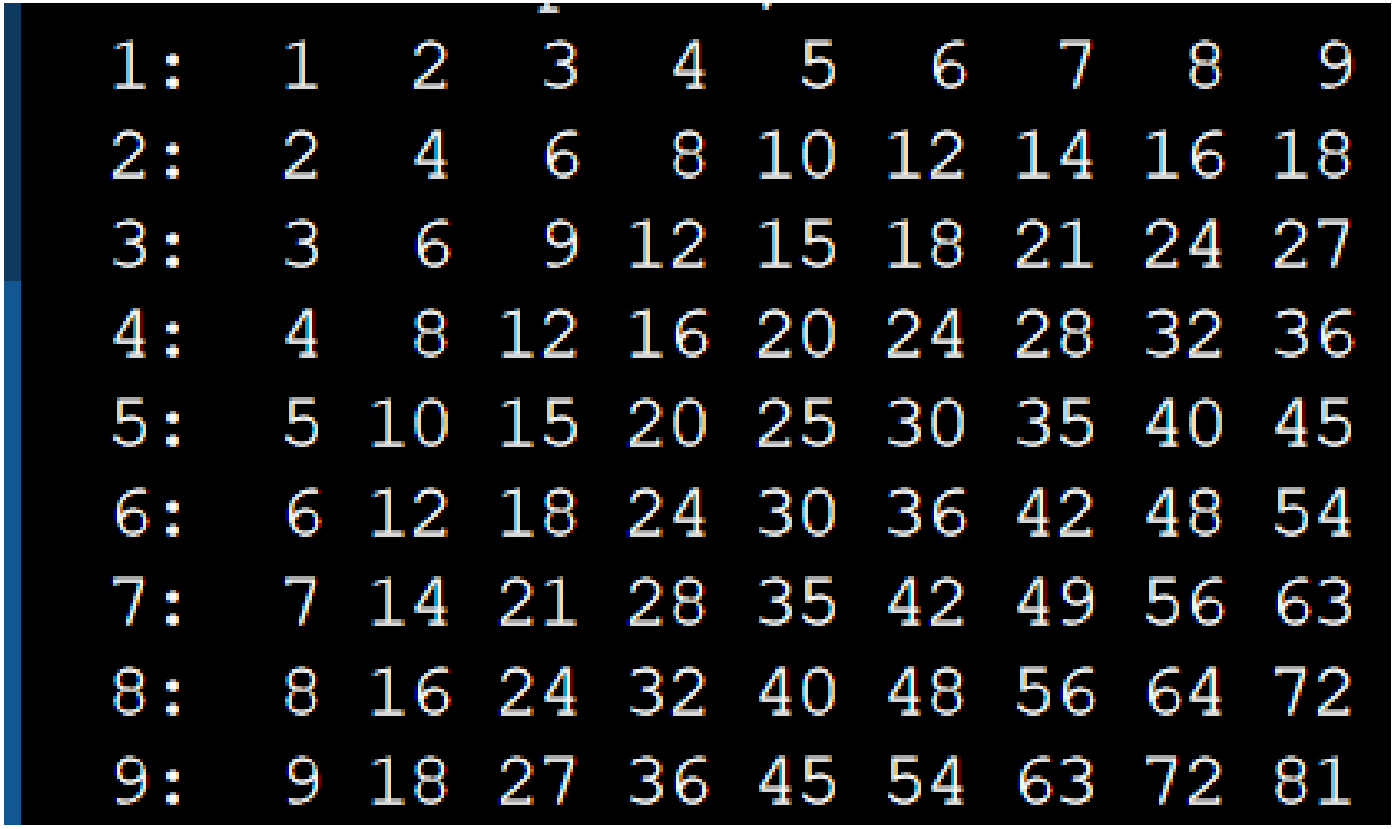
例題４．九九の表

- 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

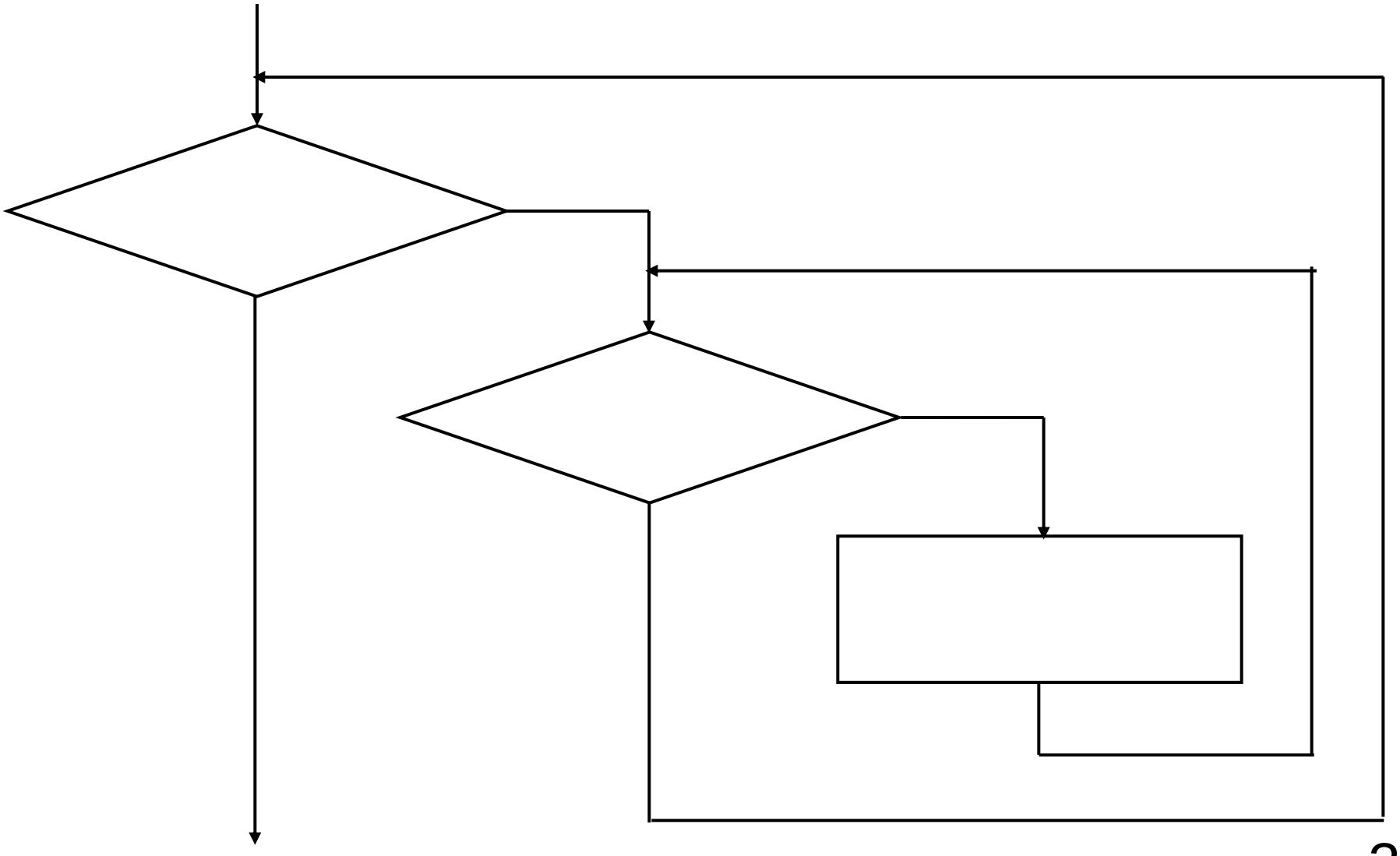
```
program sum;
uses SysUtils;
var i, j: integer;
begin
    for i := 1 to 9 do begin
        write( i:3, ':' );
        for j := 1 to 9 do begin
            write( (i*j):3 );
        end;
        writeln;
    end;
    readln
end.
```

繰り返し実行される部分

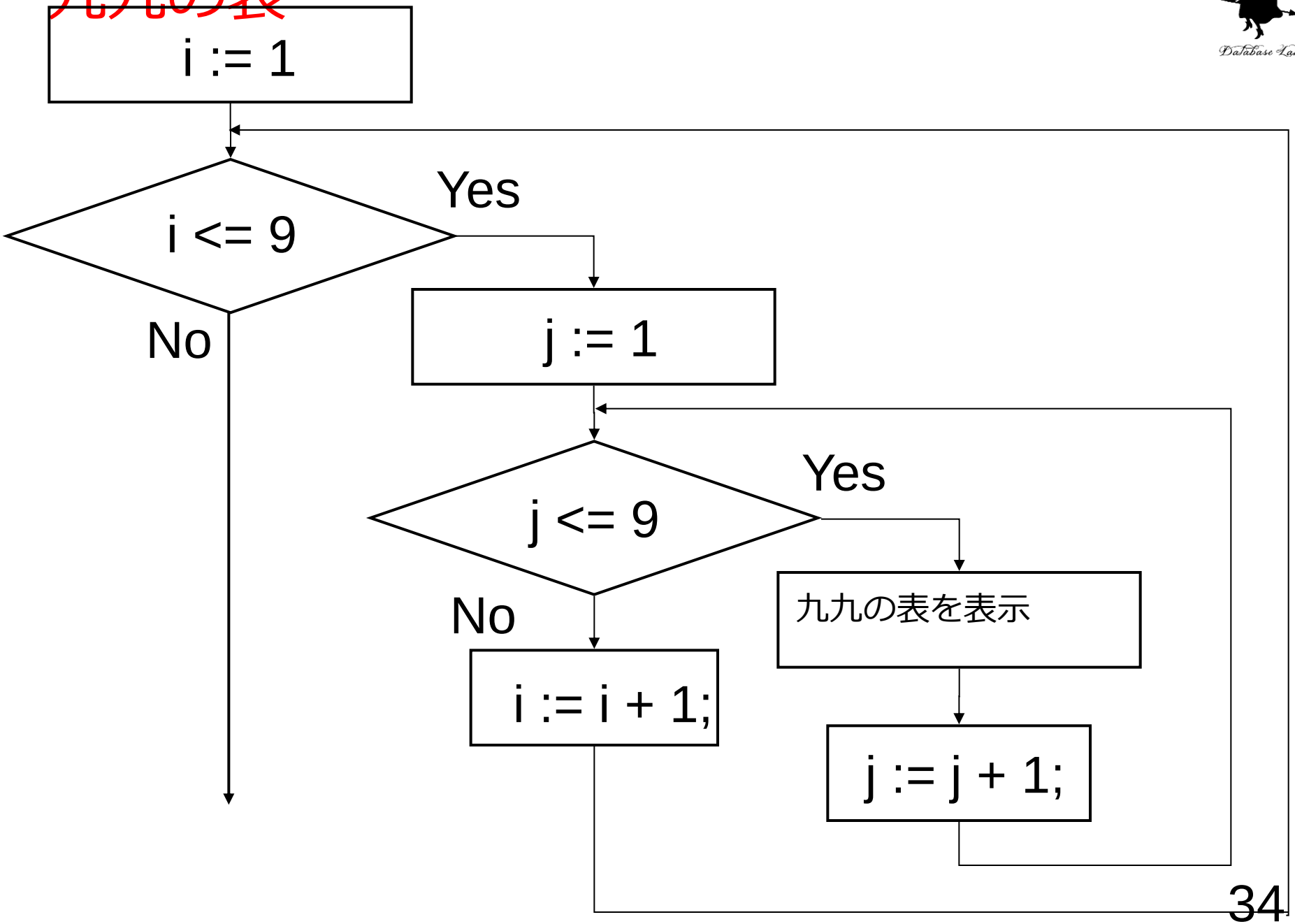
実行結果画面（例）


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 文

```
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 (sinθ+ *i* cosθ)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

z1 = x1 + *i* y1

z2 = x2 + *i* y2 のとき

$$z1\ z2 = (x1 + i\,y1)(x2 + i\,y2)$$
$$= x1x2 + x1i\,y2 + i\,y1x2 + i\,y1i\,y2$$
$$= \underbrace{x1x2 - y1y2}_{\text{実数部}} + \underbrace{i\,(x1y2 + y1x2)}_{\text{虚数部}}$$

```
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 θ

  y1 := sin θ

  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**）

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

$(\cos\theta + i\ \sin\theta)2 = \cos 2\theta - \sin 2\theta + 2i \cos\theta \sin\theta$

$= \cos 2\theta + i \sin 2\theta$

$(\cos\theta + i\ \sin\theta)3 = (\cos\theta + i\ \sin\theta)2\ (\cos\theta + i\ \sin\theta)$

$= (\cos 2\theta + i \sin 2\theta)\ (\cos\theta + i\ \sin\theta)$

$= \cos 2\theta\cos\theta - \sin 2\theta\sin\theta$

$+ i\ (\cos 2\theta\sin\theta - \sin 2\theta\cos\theta)$

$= \cos(2\theta+\theta) + i \sin(2\theta+\theta)$

$= \cos 3\theta + i \sin 3\theta$

(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**