繰り返し計算

while文, for文

本日の内容

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

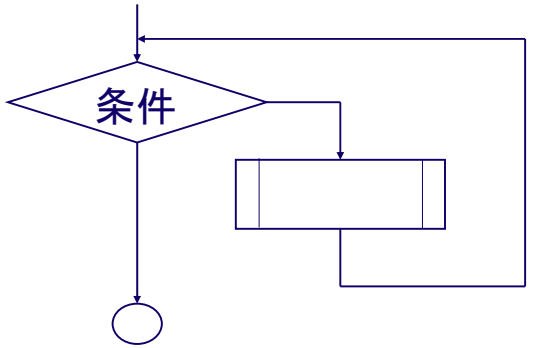
例題5. ド・モアブルの公式

　　計算誤差の累積

今日の到達目標

- 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
- 今回も，見やすいプログラムを書くために，ブロック単位での字下げを行う

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

など

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;
{$APPTYPE CONSOLE}
uses SysUtils;
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

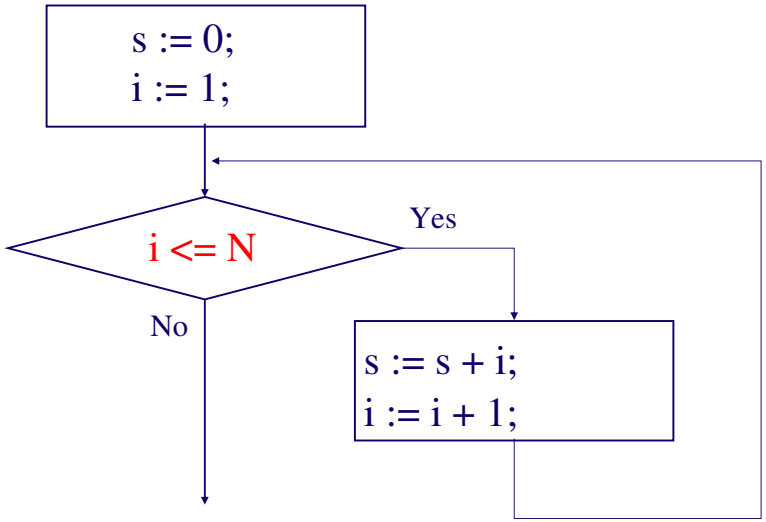
```
sum [1..N] Program. Please Enter N:  100
sum [1..N] =    5050
```

```
sum [1..N] Program. Please Enter N:  1200
sum [1..N] =    720600
```

実行結果画面（例）

```
sum [1..N] Program. Please Enter N: 1200
sum[1..n] =   720600
```

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

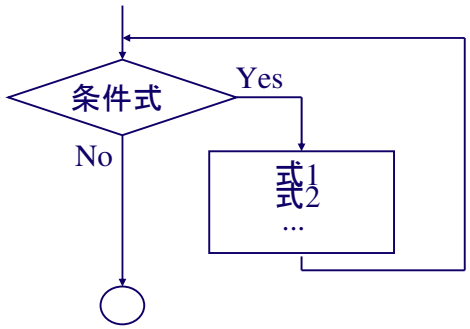
自然数の和

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

```
while 条件式 do begin
    式1;
    式2;
    …
end
```



- 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．

練習1．例題1を実行せよ

① 例題1のプログラム

② Borland Delphi 6 の起動

「スタート」→「プログラム」→「Borland Delphi 6」→「Delphi 6」

③ コンソールアプリケーションの新規作成

「ファイル」→「新規作成」→「その他」⇒ ウインドウが現れる

「コンソールアプリケーション」→「OK」 ⇒ ウインドウが現れる

④ ①のプログラムコードを「コピー」して「貼り付け」

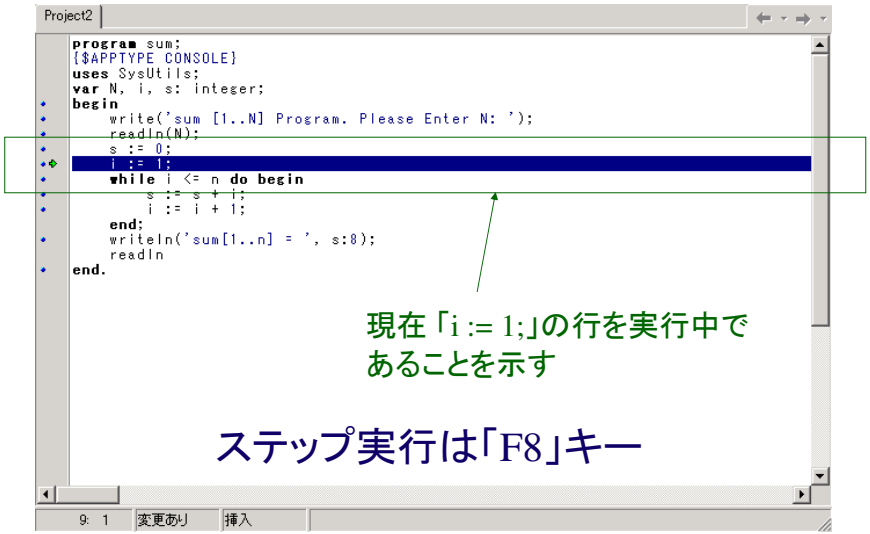
⑤ コンパイル

「プロジェクト」→「〇〇をコンパイル」

⑥ 実行

「実行」→「実行」 ☆余裕がある人は練習2に進んでください

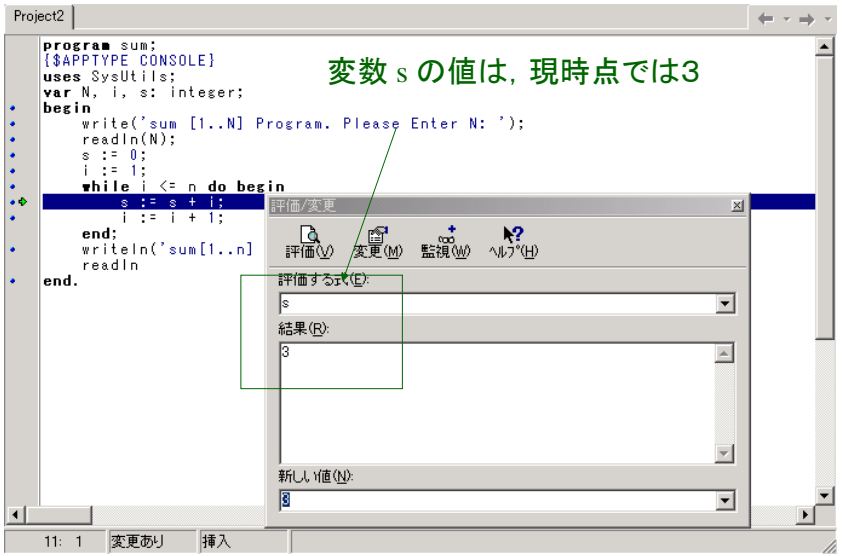
ステップ実行画面



現在「i := 1;」の行を実行中であることを示す

ステップ実行は「F8」キー

ステップ実行での変数の値の確認



変数の値の確認は「CTRL+F7」キー

ステップ実行の終わり方

- 「F9」キーを押すと，ステップ実行が終わって，通常の実行に切り替わる

ステップ実行での注意事項

- read, readln 文では，何かの値を入れるまでプログラムは止まる
  - この間，「F8」,「F9」キーなどは利かない

練習2．例題1をステップ実行せよ

① すでに「練習1」を終えていること
正確には，練習1の「⑤コンパイル」を終えた状態であること

② ここで，「F8」を押す
F8はステップ実行の意味
→ 実行画面が現れる．実行中の行が「青色」で表示される

③ さらに「F8」を2回押す
→ readln の行に達したことを確認すること

④「実行画面」のウインドウで N の値を入れる
Nの値を入れるまでプログラムは止まっている（「F8」などは受け付けられない）

⑤ さらに「F8」を押しつづける

⑥ ステップ実行に飽きたら，「F9」を押すとプログラムの最後の行まで1度に実行される

ステップ実行の意味

- プログラムの「間違い」を探すのに便利
  → ステップ実行によって，プログラムの実行の流れを確認できる
- プログラム実行の途中で，変数の値などを確認できる
  - Borland Delphi 6 では
    F9: 通常の実行
    F8: ステップ実行
    CTRL＋F7: 変数の値などの表示

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;
{$APPTYPE CONSOLE}
uses SysUtils;
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

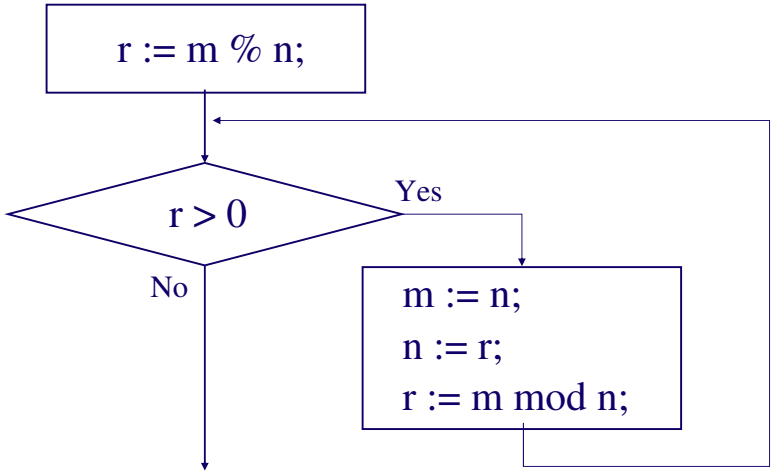
実行結果の例

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

実行結果画面(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			
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			

練習3. 例題2を実行せよ

① 例題2のプログラム

② Borland Delphi 6 の起動

「スタート」→「プログラム」→「Borland Delphi 6」→「Delphi 6」

③ コンソールアプリケーションの新規作成

「ファイル」→「新規作成」→「その他」⇒ ウインドウが現れる

「コンソールアプリケーション」→「OK」 ⇒ ウインドウが現れる

④ ①のプログラムコードを「コピー」して「貼り付け」

⑤ コンパイル

「プロジェクト」→「○○をコンパイル」

⑥ 実行

「実行」→「実行」 ☆余裕がある人は「ステップ実行」してください

例題3. 総和と平均

- データ $x_1, x_2, \ldots x_k$ 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る
  - 負の数が入力されたら終了する

例） 整数のデータ 1, 2, 3 に対して

1
2
3
-1

} 入力

```
program sum;
{$APPTYPE CONSOLE}
uses SysUtils;
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

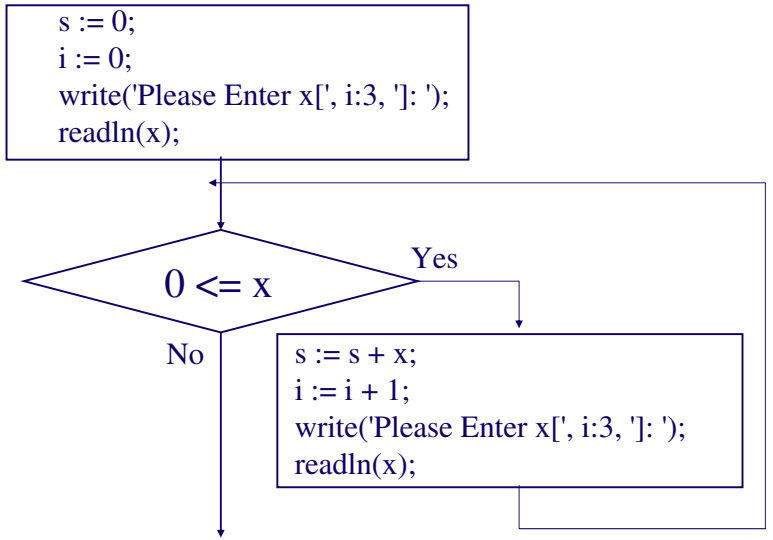
実行結果の例

```
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
```

実行結果画面(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「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

練習4. 例題3を実行せよ

① 例題3のプログラム

② Borland Delphi 6 の起動
「スタート」→「プログラム」→「Borland Delphi 6」→「Delphi 6」

③ コンソールアプリケーションの新規作成
「ファイル」→「新規作成」→「その他」⇒ ウインドウが現れる
「コンソールアプリケーション」→「OK」 ⇒ ウインドウが現れる

④ ①のプログラムコードを「コピー」して「貼り付け」

⑤ コンパイル
「プロジェクト」→「〇〇をコンパイル」

⑥ 実行
「実行」→「実行」 ☆余裕がある人は「課題1」に進んでください

課題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
  - 必ず，動作確認まで行うこと
  - (余裕がある人のみ)和だけでなく平均も表示させよ

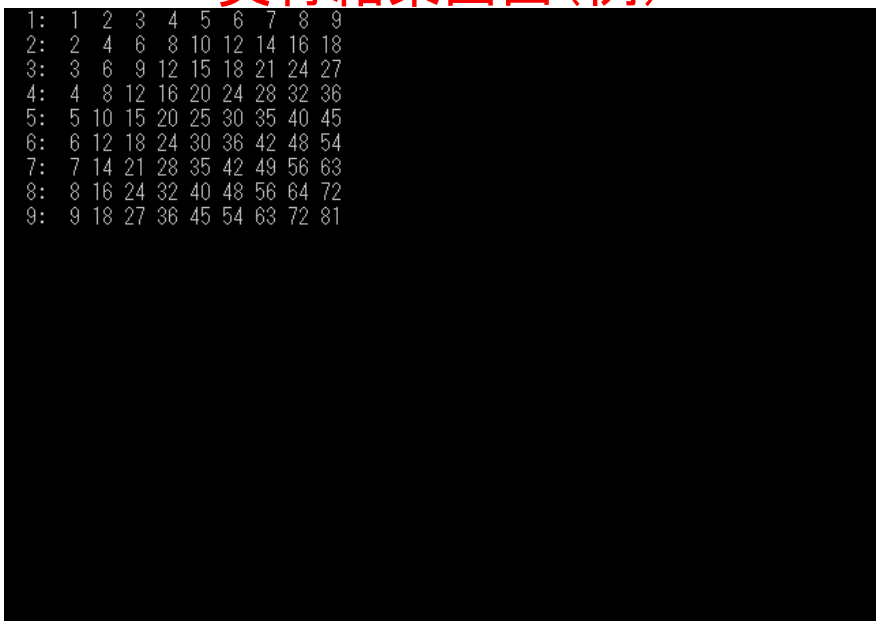
```
program sum;
{$APPTYPE CONSOLE}
uses SysUtils;
var i, j: integer;
begin
  for i := 1 to 9 do begin
    write( i:3, ':' );
    for j := 1 to 9 do begin
      write( (i*j):3 );
    end;
    writeln;
  end;
  readln
end.
```

繰り返し実行される部分

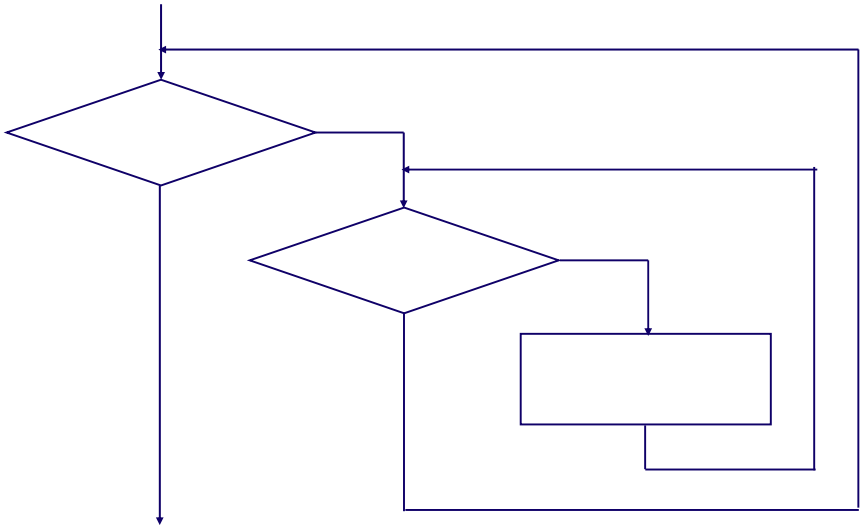
例題4．九九の表

- 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

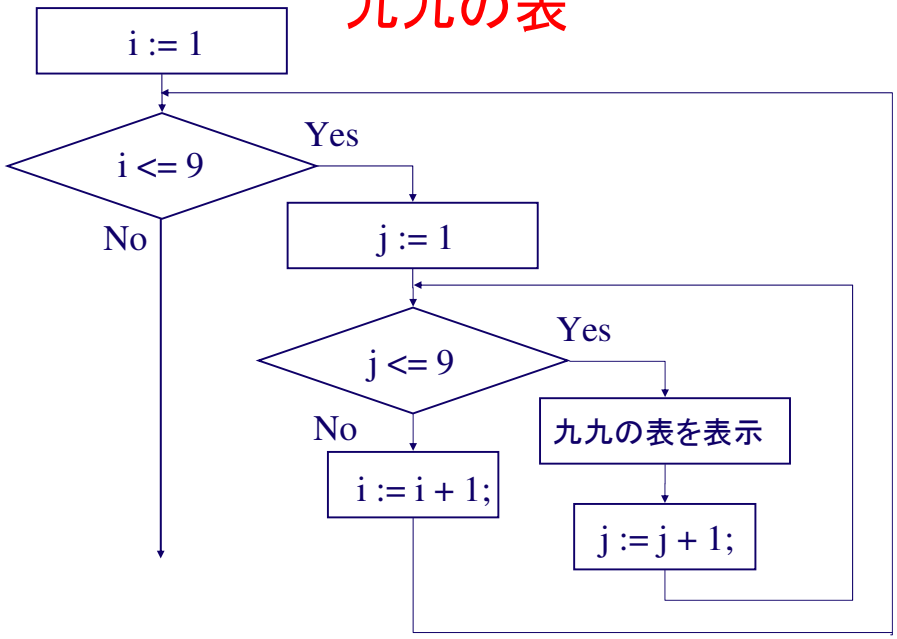
実行結果画面(例)



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

$z_1 = x_1 + i y_1$

$z_2 = x_2 + i y_2$ のとき

$$z_1 z_2 = (x_1 + i y_1)(x_2 + i y_2)$$
$$= x_1 x_2 + x_1 i y_2 + i y_1 x_2 + i y_1 i y_2$$
$$= \underbrace{x_1 x_2 - y_1 y_2}_{\text{実数部}} + \underbrace{i(x_1 y_2 + y_1 x_2)}_{\text{虚数部}}$$

```
program sum;
{$APPTYPE CONSOLE}
uses SysUtils;
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，「浮動小数」の計算は，あくまでも近似計算（有効精度１２桁程度）
- 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