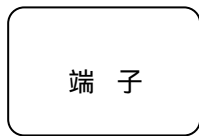
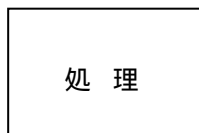


フローチャート（流れ図）の基本



: 流れ図の始まりと終りを表す



: あらゆる種類の処理や行動を表す



: 条件を判断し、それ以降の流を分岐させるために用いる



: 入出力機能一般を表す

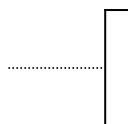


: 結合子



流れ線

: 流れ図記号を結び付け、実行の順序を示す。
・上から下、左から右への時計回り
・必要に応じて矢印をつける
・美しく、流が良く分りやすいように



: 流れ図記号の説明や、注釈書きに用いる

とにかく書き込み実践して、慣れること。

1.2 簡単なプログラム例 (Pascal プログラミング pp.2-4)

例題 1.1 四則演算

二つの整数データを読み込んで、それらの和、差、積、商の値を計算して出力するプログラムを作れ。

```

program arithmetic(input, output);
  var i, j, s, d, p, q : integer;
begin
  read(i, j);
  s := i + j; d := i - j; p := i * j; q := i div j;
  writeln(s, d, p, q)
end.

```

プログラム <例題 1.1>

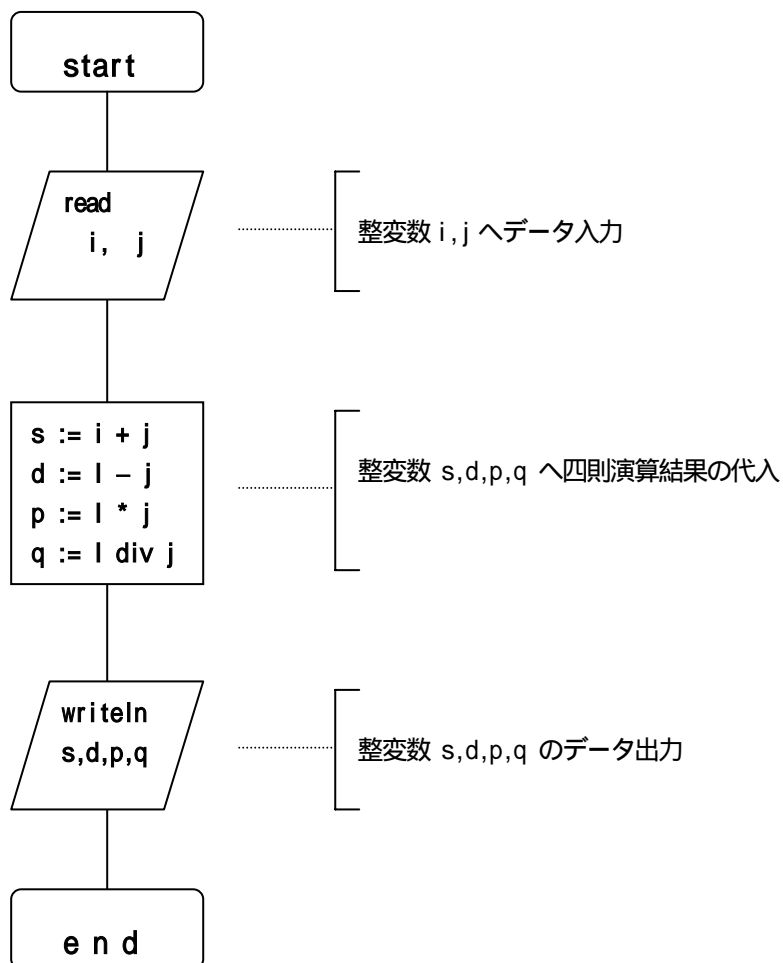
```

1000 16
      1016      984      16000      62

```

実行結果 <例題 1.1>

フローチャート



1.2 簡単なプログラム例 (Pascal プログラミング pp.4-6)

例題 1.2 最大値

五個の正の実数を順に呼んで、それらのうちの最大値を出力せよ。

```

program maximum(input, output);
  const n = 5;
  var x, max : real; i : integer;
begin
  max := 0;
  for i := 1 to n do
  begin
    read(x);
    if x > max then max := x
  end;
  writeln;
  writeln('maximum = ', max)
end.
    
```

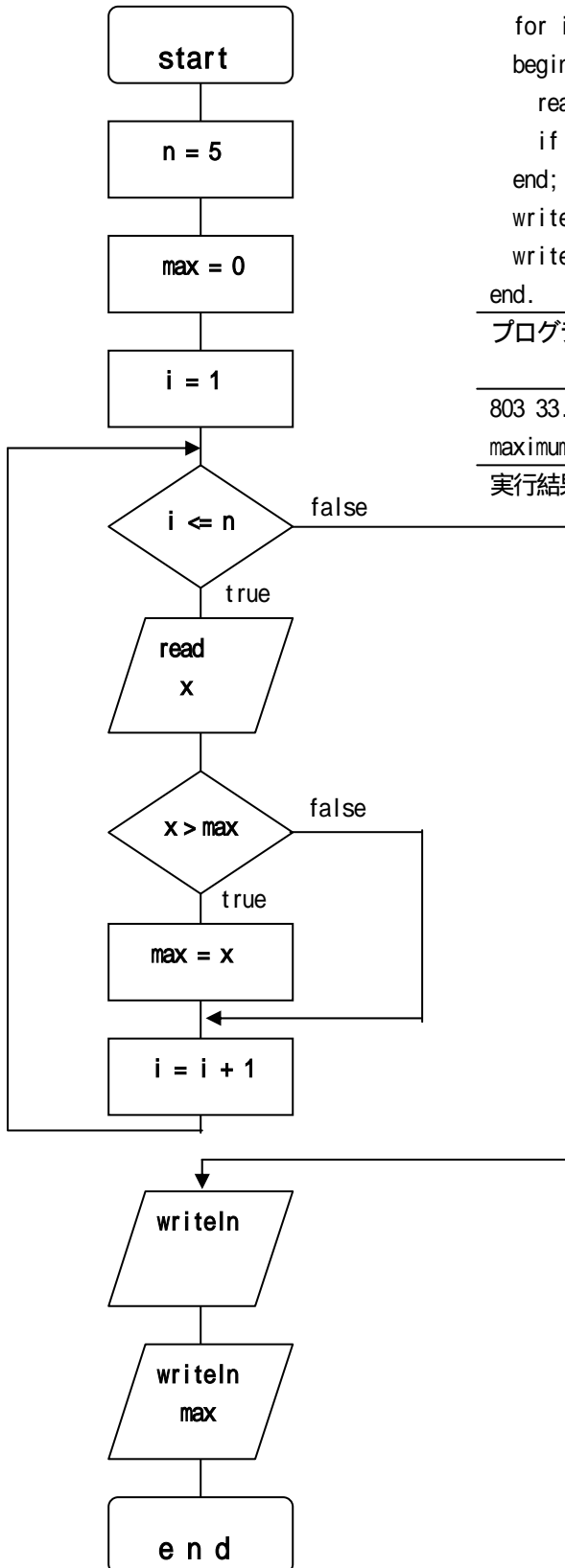
プログラム <例題 1.2>

803 33.5 9.01 10.34 22.57

maximum = 3.350000000000000e+01

実行結果 <例題 1.2>

フローチャート

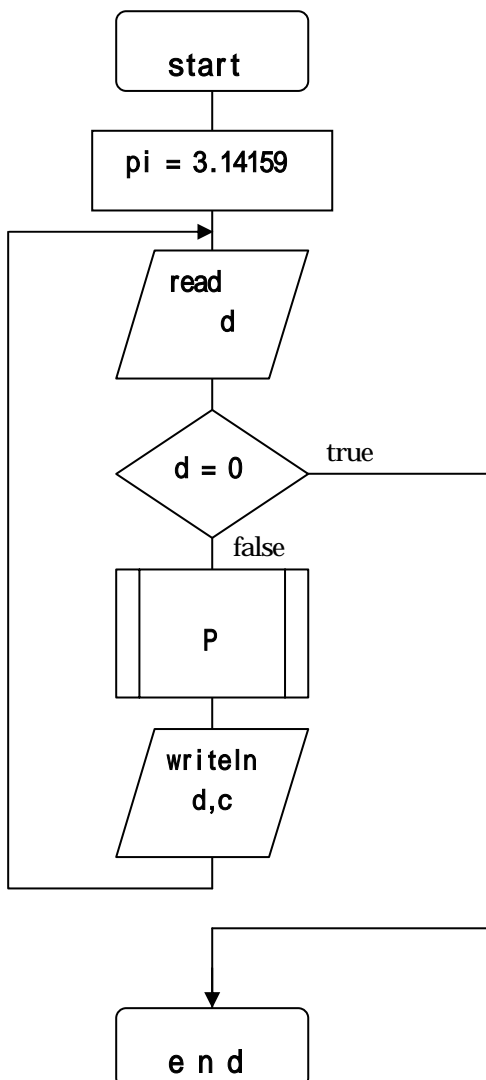


3 プログラムの構造 (Pascal プログラミング pp.12-16)

Pascal プログラムを構成する7つの部分

program example(input, output);	{プログラム頭部 }
label 1,90;	{ラベル宣言部 }
const pi = 3.14159;	{定数定義部 }
type data = real;	{型定義部 }
var d : data ; c : real;	{変数宣言部 }
procedure p ; begin c := pi * d end;	{手続き関数宣言部 }
begin	{実行文部 }
1: read(d);	{ 々 }
if d = 0 then goto 90;	{ 々 }
p;	{ 々 }
writeln(d, c);	{ 々 }
goto 1;	{ 々 }
90:	{ 々 }
end.	{ 々 }

フローチャート(Main)



フローチャート(手続き関数 P)

