# pe-6. 配列

（Pascal プログラミング入門）

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

金子邦彦

# 内容

- 例題１．月の日数
  - 配列とは．配列の宣言．配列の添字．
- 例題２．ベクトルの内積
- 例題３．棒グラフを描く
- 例題４．Horner 法による多項式の計算
- 例題５．エラトステネスのふるい
  - 配列と繰り返し計算の関係

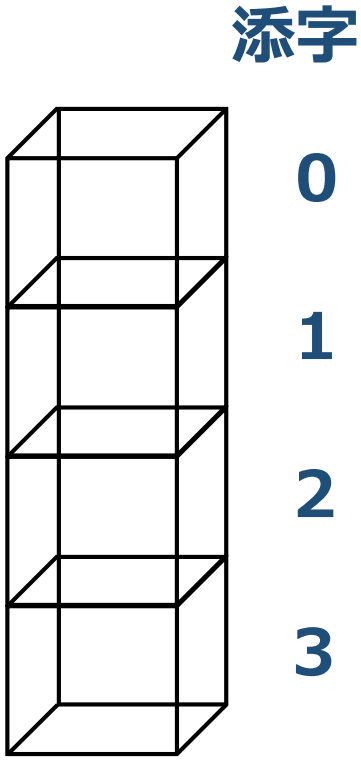
# 目標

- **配列**とは何かを理解し，**integer, real の配列を使ったプログラム**を書けるようになる

- **配列**と**繰り返し文**を**組み合わせて，多量のデータ**を扱えるようになる

# 配列

- **データの並び**で，番号（添字）が付いている



# 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

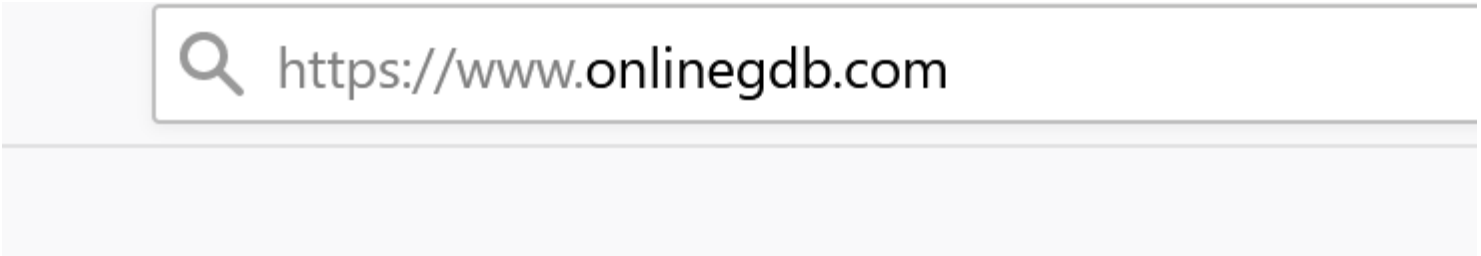
- オンラインなので、「秘密にしたいプログラム」を扱うには十分な注意が必要

# Online GDB で Pascal を動かす手順

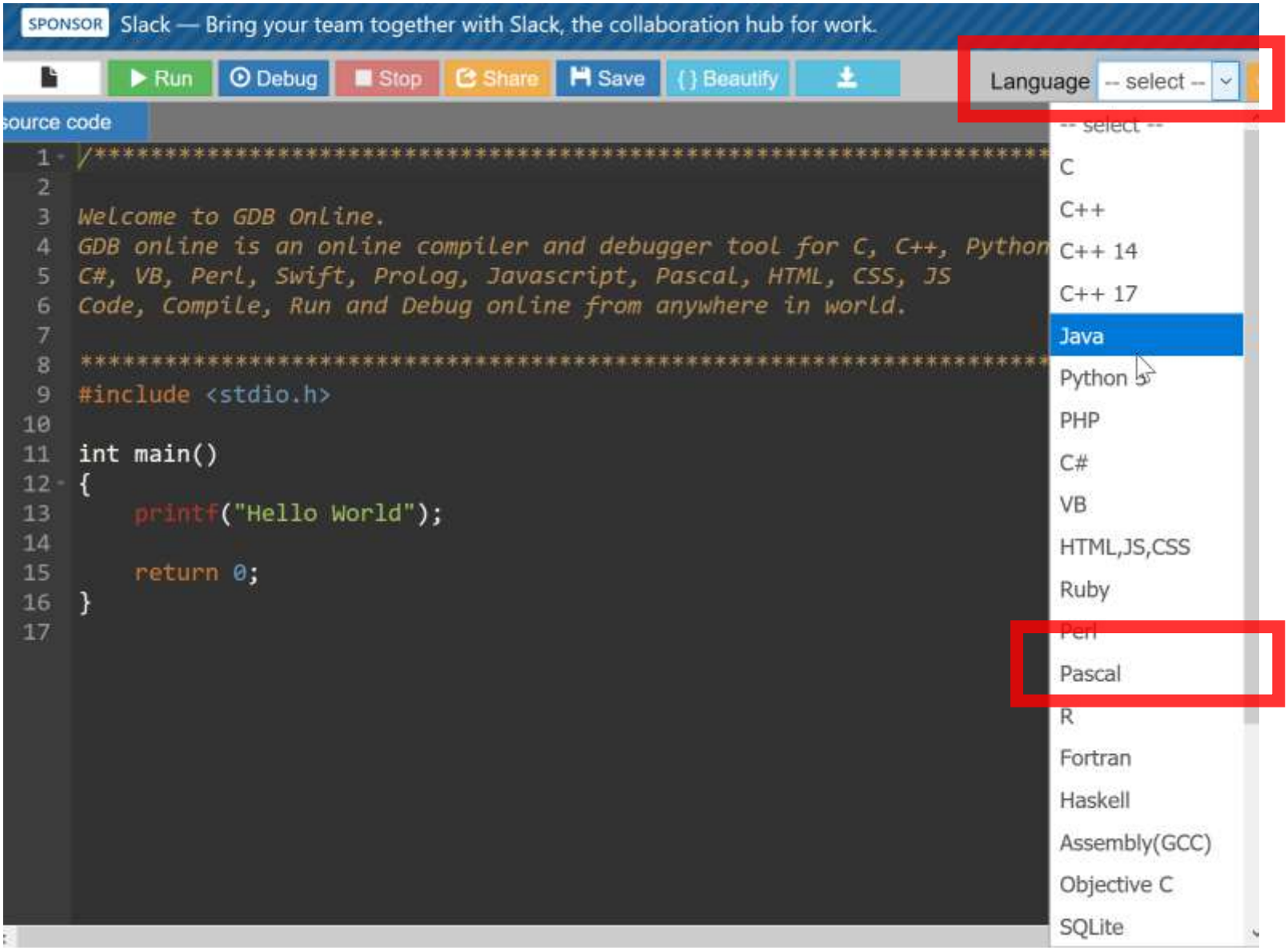
① ウェブブラウザを起動する

② 次の URL を開く

https://www.onlinegdb.com



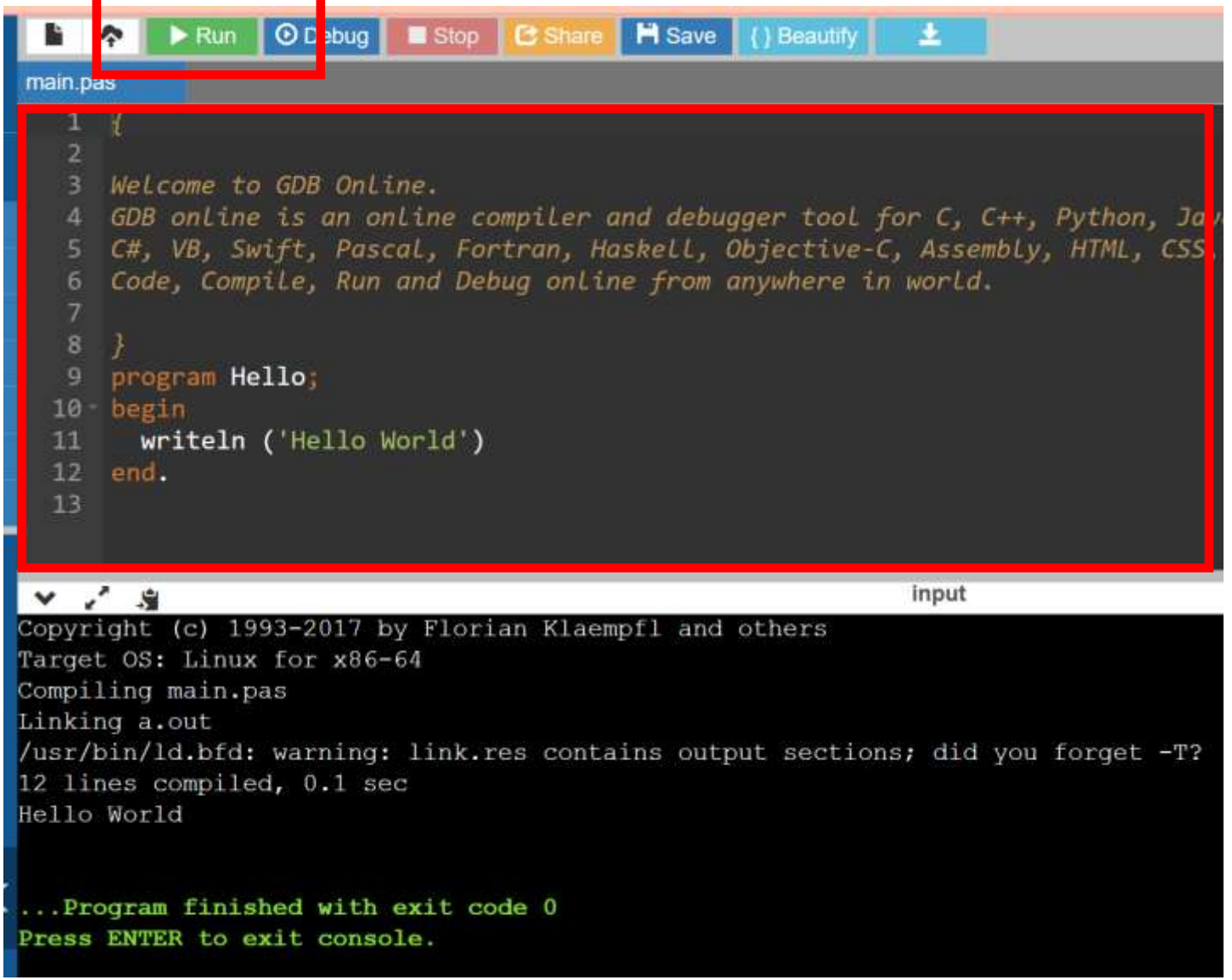
# ③「**Language**」のところで，「**Pascal**」を選ぶ



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．月の日数

- 年と月を読み込んで，日数を求めるプログラムを作る
  - **うるう年の２月ならば２９**
  - **日数を求めるために，サイズ１２（１から１２まで）の integer の配列を使う**

例） 2021 年 11 月 → 30

```
program sum;
var num : array [1..12] of integer;
var y, m : integer;
begin
    num[1] := 31; num[2] := 28; num[3] := 31; num[4] := 30;
    num[5] := 31; num[6] := 30; num[7] := 31; num[8] := 31;
    num[9] := 30; num[10] := 31; num[11] := 30; num[12] := 31;
    write('Please Enter y(year): ');
    readln(y);
    write('Please Enter m(month): ');
    readln(m);
    if (m=2) and (((y mod 400)=0) or (((y mod 100)<>0) and ((y mod 4)=0))) then begin
        writeln('number of days(', y, '/', m, ') is 29');
    end
    else begin
        writeln('number of days(', y, '/', m, ') is ', num[m]);
    end;
    readln
end.
```

配列への書き込み

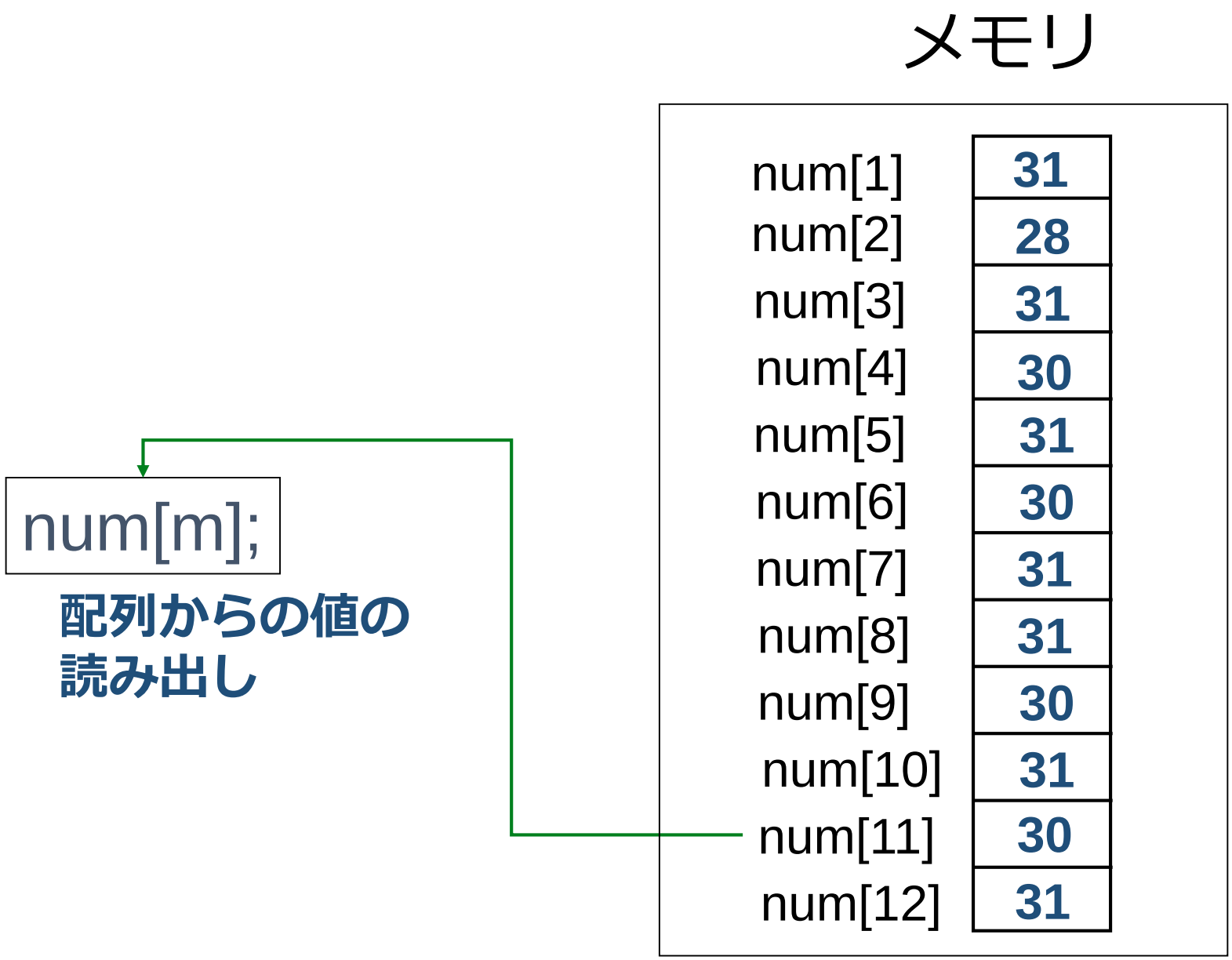
配列からの読み出し

# 月の日数

実行結果の例

```
Please Enter y(year): 2022
Please Enter m(month): 6
number of days(2022/6) is 30
```

# プログラムとデータ


メモリ
num[1] 31
num[2] 28
num[3] 31
num[4] 30
num[5] 31
num[6] 30
num[7] 31
num[8] 31
num[9] 30
num[10] 31
num[11] 30
num[12] 31
num[m];
配列からの値の
読み出し

# 配列の宣言

- **配列**には，**名前**と**型**（**データの種類**のこと）と**サイズ**がある
  - 整数データ　　**integer**
  - 浮動小数データ　　**real**
- **配列**を使うには，**配列の使用をコンピュータに伝える**こと（宣言）が必要

```
var num : array [1..12] of integer;
```

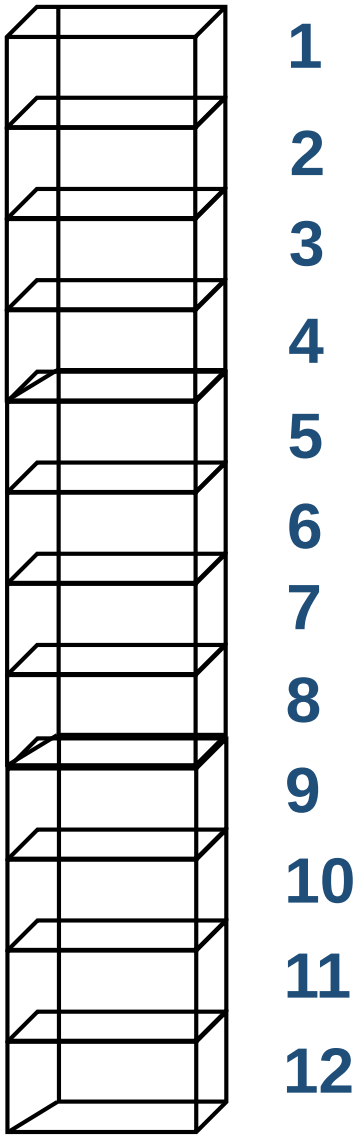
名前は num　　配列の添字は１から１２　　整数データ

# array [ ] of の意味

- [] の中に，**添字の範囲**を書く
  おのずと，配列のサイズが決まる

**array [1..12] of integer のとき**
**添字は，１から１２まで**

# 配列の添字




配列の中身を読み書きするときには，配列の名前と添字を書く

例）　num[m]

**添字**

# 配列の読み書き

- 配列の名前と添字を書く

例）

```
num[1] := 31;
writeln('number of days(' y, '/', m, ') is', num[m] );
```

# 例題２．ベクトルの内積

- **ベクトル（1.9, 2.8, 3.7）**と，**ベクトル（4.6, 5.5, 6.4）**の**内積**を表示するプログラムを作る

  ２つのベクトルの内積の計算のために，サイズ３の配列を２つ使う

# ベクトルの内積

ベクトルの成分から内積を求める

$\vec{a} = (a_0, a_1, a_2)$　$\vec{b} = (b_0, b_1, b_2)$ のとき

$$\vec{a} \bullet \vec{b} = a_0 b_0 + a_1 b_1 + a_2 b_2$$

```
program sum;
var u, v : array [0..2] of real;
var i : integer;
var ip : real;
begin
  u[0] := 1.9; u[1] := 2.8; u[2] := 3.7;
  v[0] := 4.6; v[1] := 5.5; v[2] := 6.4;
        ip := 0;
        for i := 0 to 2 do begin
    ip := ip + u[i] * v[i];
  end;
  writeln('product =', ip:8:3 );
  readln
end.
```

配列への書き込み

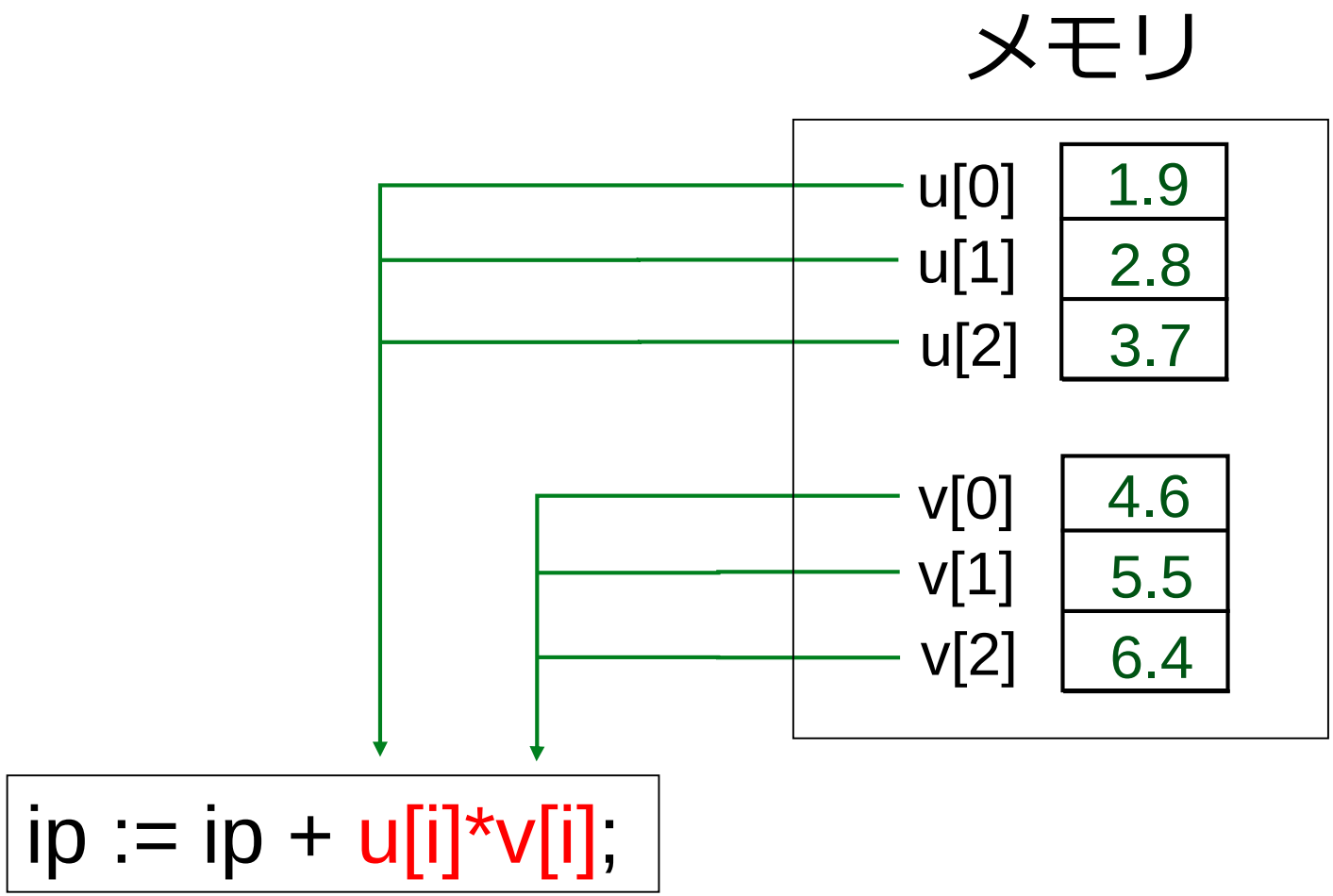
配列からの読み出し

# ベクトルの内積

実行結果の例

```
product =   47.820
```

# プログラムとデータ



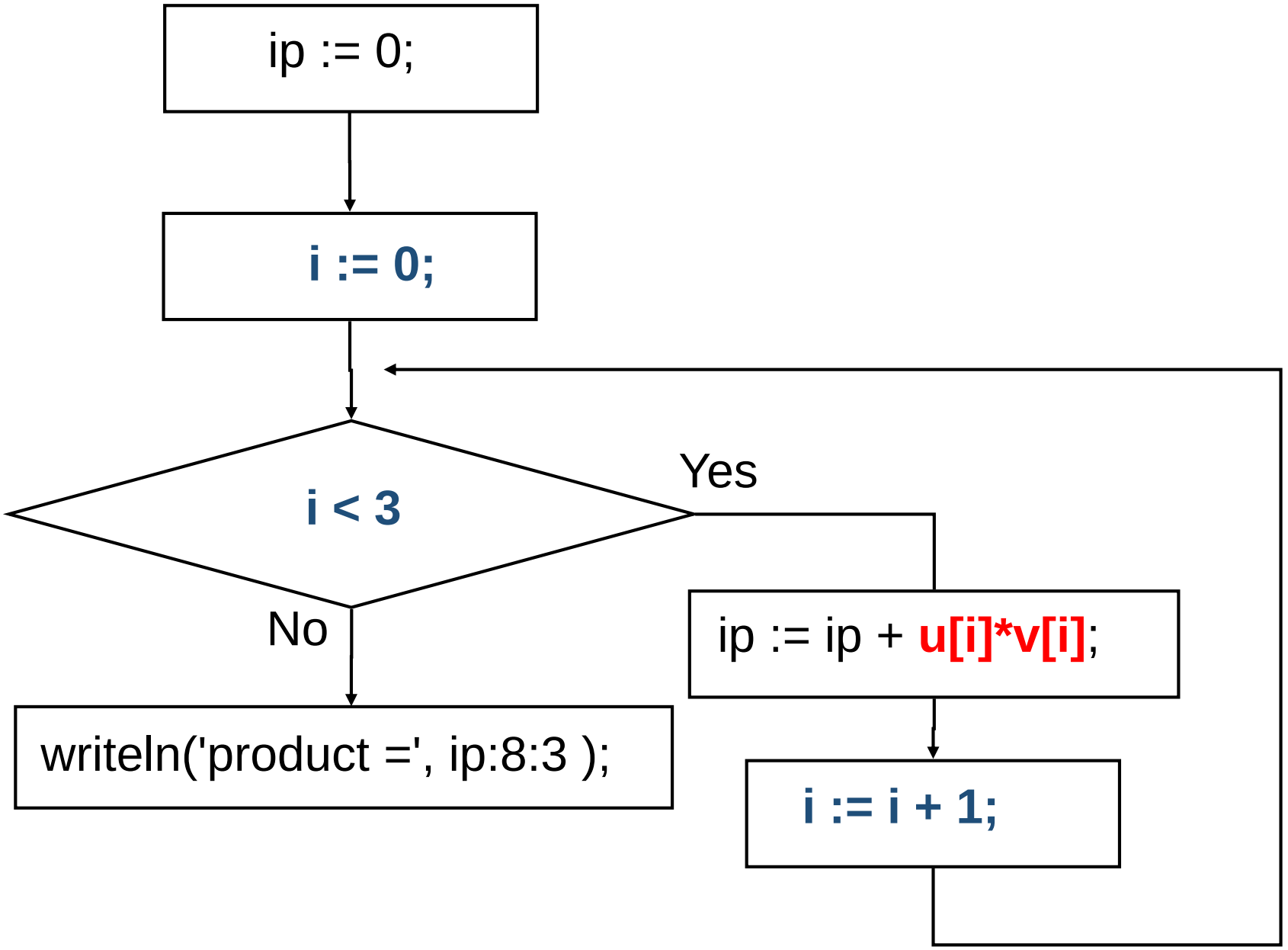

**配列からの**
**読み出し**

# 配列の宣言

```
var u, v : array [0..2] of real;
```

名前は　　添字は　　浮動小数
ｕとｖ　０から２まで　データ

# プログラム実行順



# ベクトルの内積

| | i の値 | 繰り返し条件式が成り立つか | ip の値 |
|---|---|---|---|
| 繰り返し1回目 | i = 0 | i < 3 が成り立つ | ip := ip + u[0] * v[0]; |

つまり ip の値は u[0]*v[0]

| | | | |
|---|---|---|---|
| 繰り返し2回目 | i = 1 | i < 3 が成り立つ | ip := ip + u[1] * v[1]; |

つまり ip の値は u[0]*v[0] + u[1]*v[1]

| | | | |
|---|---|---|---|
| 繰り返し3回目 | i = 2 | i < 3 が成り立つ | ip := ip + u[2] * v[2]; |

つまり ip の値は u[0]*v[0] + u[1]*v[1]
+u[2]*v[2]

| | | |
|---|---|---|
| 繰り返し4回目 | i = 3 | i < 3 が成り立たない |

# 例題３．棒グラフを描く

- 整数の配列から，その棒グラフを表示するプログラムを作る．
  ループの入れ子で，棒グラフの表示を行う

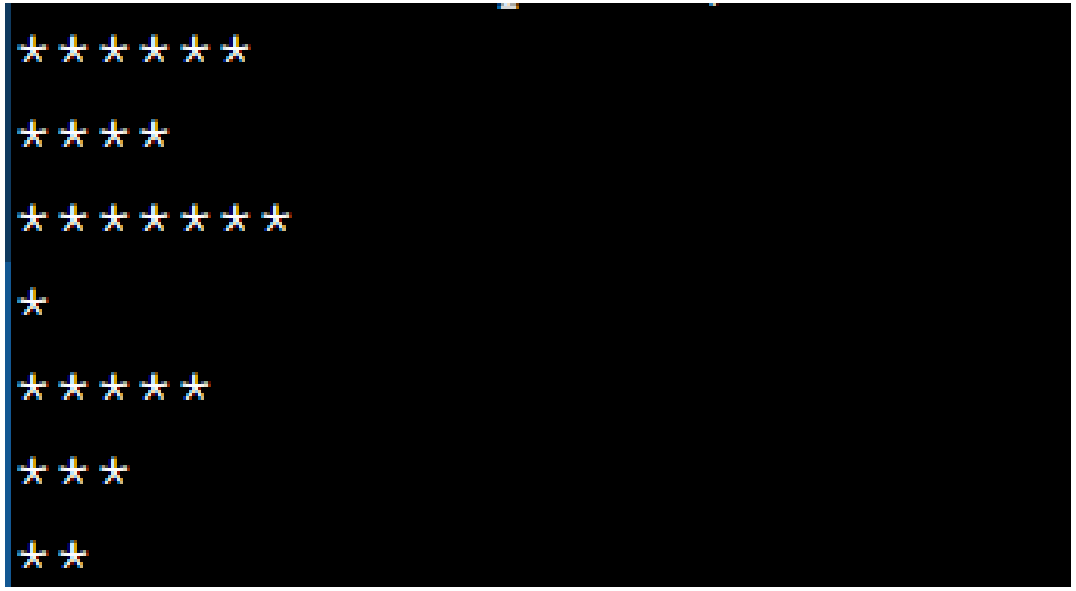
```
program sum;
var a : array [1..7] of integer;
var i, j : integer;
begin
  a[1] := 6; a[2] := 4; a[3] := 7; a[4] := 1;
  a[5] := 5; a[6] := 3; a[7] := 2;
  for i := 1 to 7 do begin
    for j := 1 to a[i] do begin
      write('*');
    end;
    writeln;
  end;
  readln
end.
```

配列への書き込み

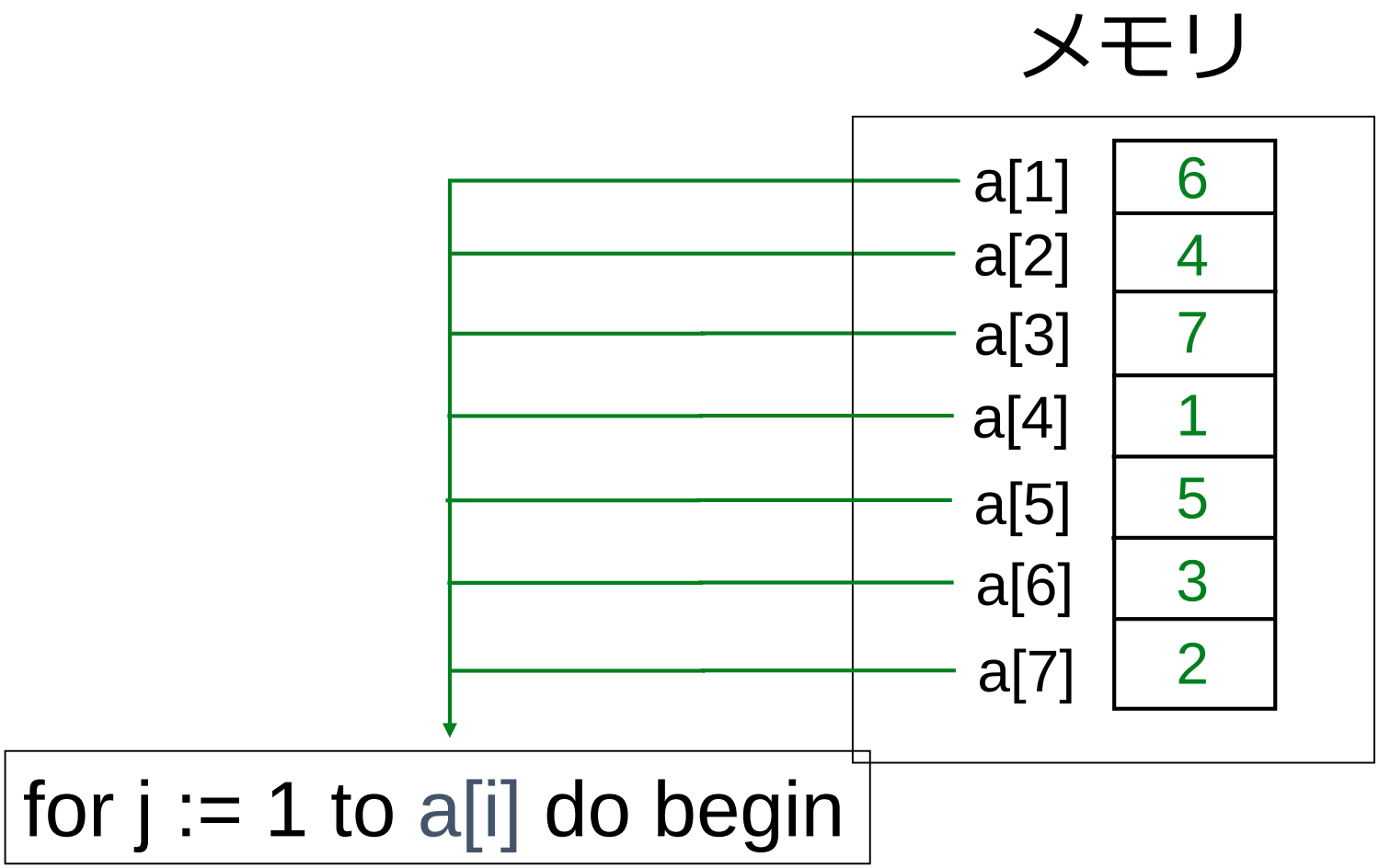
配列からの読み出し

# 棒グラフを描く

実行結果の例

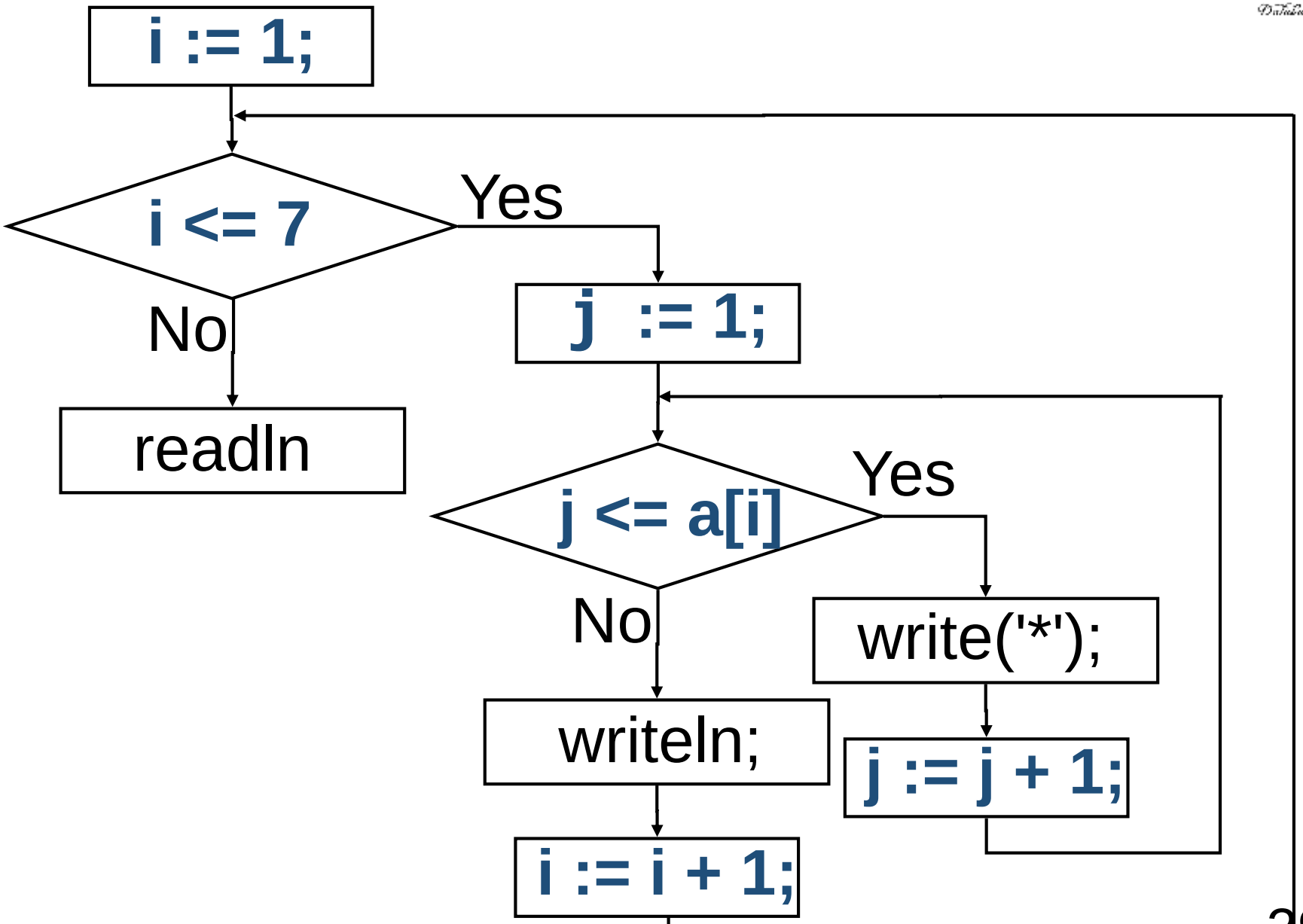


# プログラムとデータ




**配列からの**
**読み出し**

# プログラム実行順



# 例題４． Horner 法による多項式の計算

- **n次の多項式**

  f(x) = a0 + a1・x + a2・x2 + ・・・ +an・xn

  について，**次数 n** と，**係数 a0 から an** を読み込んで，**f(x) を計算**するプログラムを作る
  - まず n を読み込んで，その後に a0 から an を読み込む．最後に x を読み込む.
  - 次ページで説明する Horner法を使う
  - 読み込んだ係数は，いったん配列に格納する． n は高々２０とする.

# Horner法

$f(x) = a0 + a1 \cdot x + a2 \cdot x^2 + \cdots + an \cdot x^n$

$= a0 + ( a1 + ( a2 + \cdots + ( an-1 + an \cdot x ) x \cdots ) x ) x$

例えば, $5 + 6x + 3x^2$

$= 5 + ( 6 + 3x ) x$

計算手順

① an

② an-1 + an・x

③ an-2 + ( an-1 + an・x ) x

・・・ （$a_0$ まで続ける）

```
program sum;
var a : array [0..20] of real;
var i, n : integer;
var x, y : real;
begin
    write('Please Enter n: ');
    readln(n);
    for i := 0 to n do begin
        write('Please Enter a[', i, '] : ');
        readln(a[i]);
    end;
    write('Please Enter x: ');
    readln(x);
    y := a[n];
    i := n - 1;
    while i >= 0 do begin
        y := y * x + a[i];
        i := i - 1;
    end;
    writeln('y =', y:8:3 );
    readln
end.
```

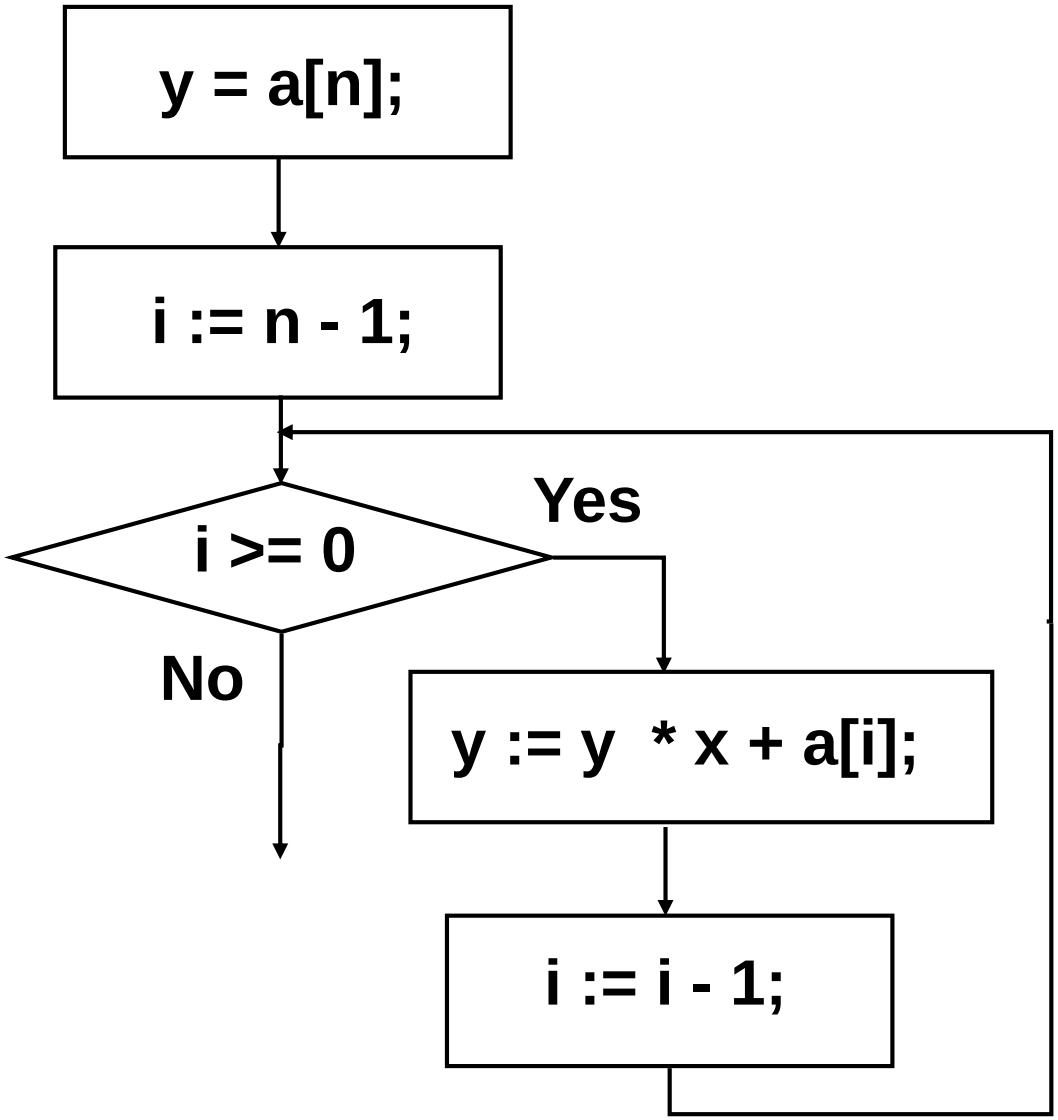
**配列への書き込み**

**配列からの読み出し**

# 実行結果の例

```
Please Enter n: 2
Please Enter a[0] : 5
Please Enter a[1] : 6
Please Enter a[2] : 3
Please Enter x: 3
y =   50.000
```

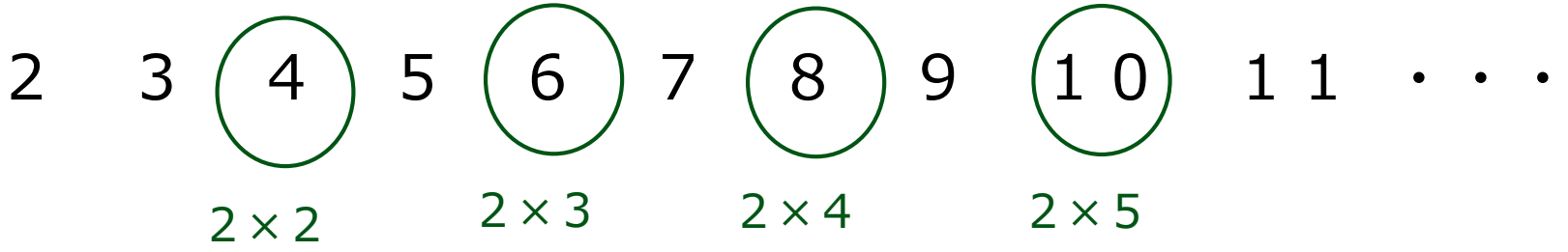
# Horner 法



# 例題５．エラトステネスのふるい

- 「**エラトステネスのふるい**」の原理に基づいて**100以下の素数**を求め，結果を表示するプログラムを作成する
  - １００以下の素数を求めるために，**２から１００までの配列**を使う
  - 配列には，**添字が素数なら１，そうでなければ０**をセットする

# エラトステネスのふるい (1/4)




まず，２の倍数を消す

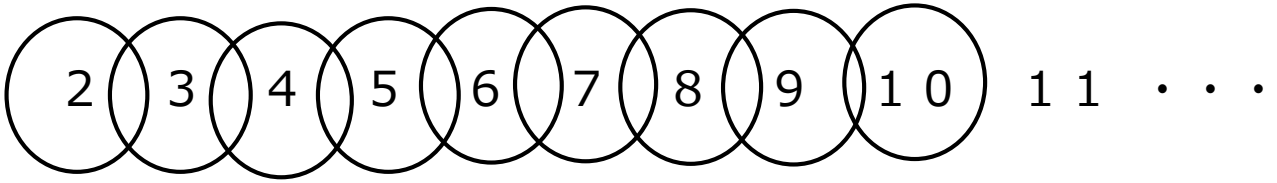
# エラトステネスのふるい (2/4)


次に，３の倍数を消す

# エラトステネスのふるい (3/4)


次に，５の倍数を消す
（「４の倍数」は考えない.
それは，「４」がすでに消えているから）

# エラトステネスのふるい (4/4)




以上のように，２，３，５，７の倍数を消す.

１０（これは１００の平方根）を超えたら，この操作を止める
（１００以下で，１１，１３・・・の倍数はすでに消えている）

```
program sum;
var p : array [2..100] of integer;
var n, i : integer;
begin
  for i := 2 to 100 do begin
    p[i] :=1;
  end;
  n := 2;
  while n <= sqrt( 100 ) do begin
    i := 2;
    while ( n * i ) <= 100 do begin
      p[ n * i ] := 0;
      i := i + 1;
    end;
    n := n + 1;
    while ( p[n] = 0 ) and ( n <= sqrt(100) ) do begin
      n := n + 1;
    end;
  end;
  for i := 2 to 100 do begin
    if p[i] = 1 then begin
      write( i, ',' );
    end;
  end;
  readln
end.
```

a[2] から a[100] までを「１」にセット

まず「２」の倍数から

n が 100 を超えたら終わり

n の倍数を「消す」

( n が 100 以上になったら終わり)

n の「次」の素数を探す

求めた素数の表示

# 実行結果の例

# 演習１．1000 までの素数

- **エラトステネスのふるい**を使って**1000までの数の素数**を求めるプログラムを作りなさい

  例題５のプログラムを書き換えなさい