ji-6. 配列

（Java プログラミング入門）

URL: https://www.kkaneko.jp/pro/ji/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

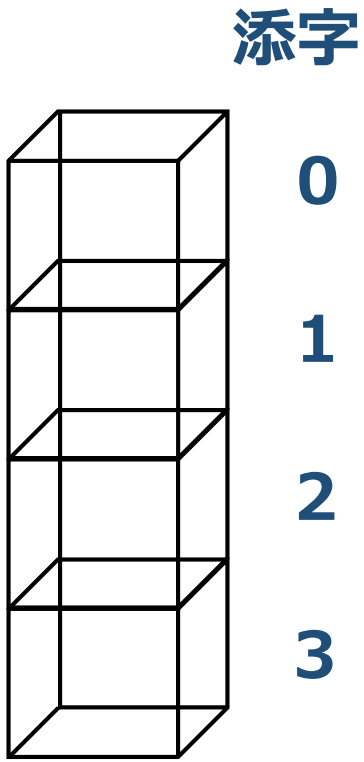
目標

- **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

  Python3, Java, C/C++, C#, JavaScript,
  R, アセンブリ言語，SQL など

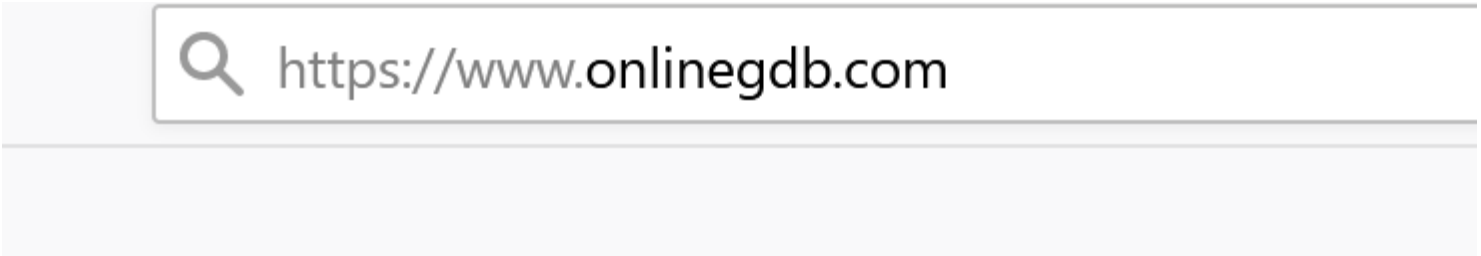
- オンラインなので、「秘密にしたいプログラム」を扱うには十分な注意が必要

Online GDB で Java を動かす手順

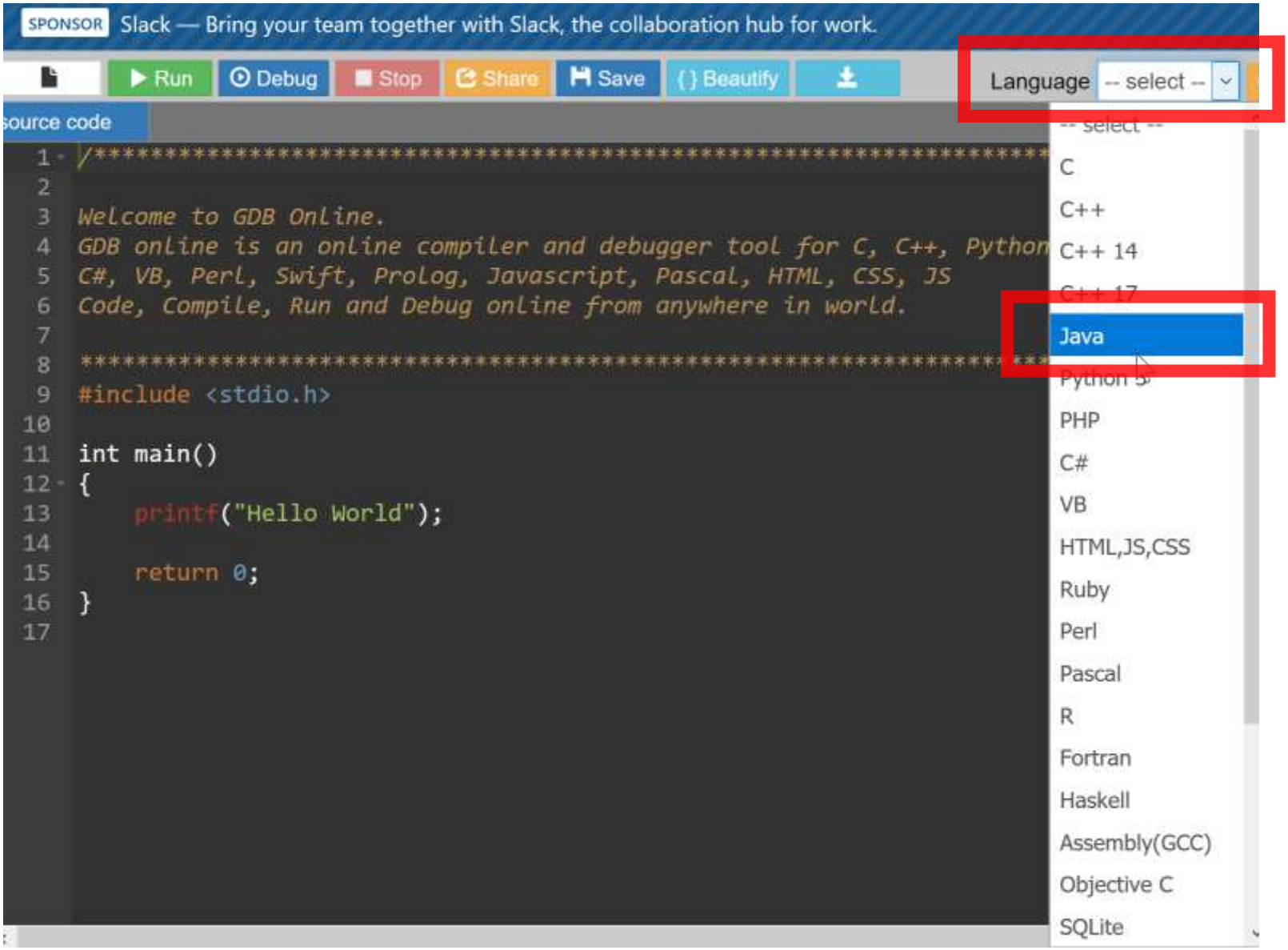
① ウェブブラウザを起動する

② 次の URL を開く

https://www.onlinegdb.com



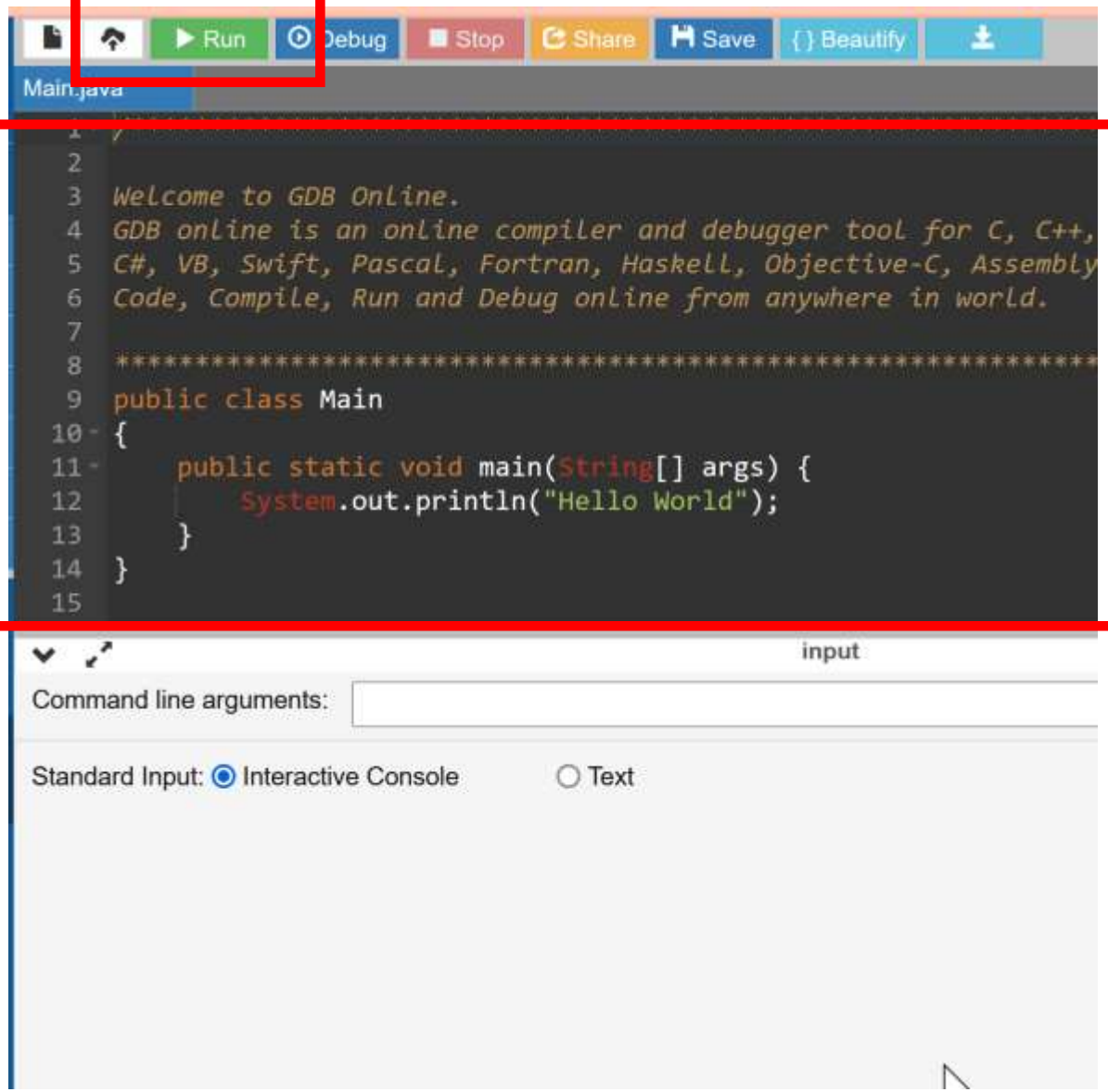
③「**Language**」のところで，「**Java**」を選ぶ



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

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        int y, m;
        int num[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        Scanner s = new Scanner(System.in);
        System.out.println("Please Enter y(year) =");
        y = s.nextInt();
        System.out.println("Please Enter m(month) =");
        m = s.nextInt();
        if ((m == 2) && (((y % 400) == 0) || (((y % 100) != 0) && ((y % 4) == 0)))) {
          System.out.printf("number of days %d/%d is 29¥n", y, m);
        } else {
          System.out.printf("number of days %d/%d is %d¥n", y, m, num[m]);
        }
    }
}
```

配列への書き込み

うるう年の判定

配列からの読み出し

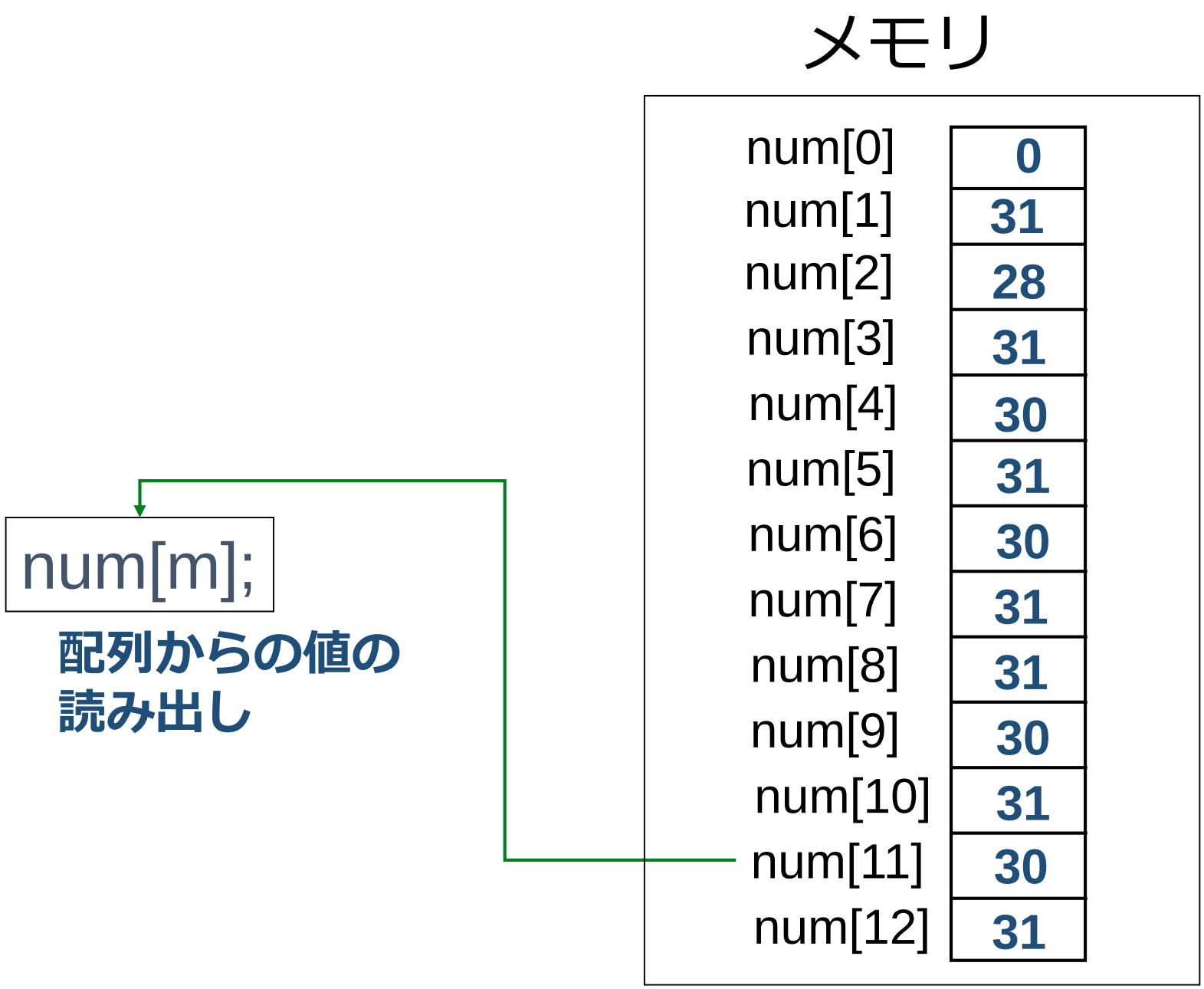
月の日数

実行結果の例

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

```
Please Enter y(year) =
2022
Please Enter m(month) =
2
number of days 2022/2 is 28
```

プログラムとデータ


メモリ
num[0]
num[1]
num[2]
num[3]
num[4]
num[5]
num[6]
num[7]
num[8]
num[9]
num[10]
num[11]
num[12]
0
31
28
31
30
31
30
31
31
30
31
30
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　　**int**
  - 浮動小数データ　**float** や **double**
- **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

```
int num[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

整数　名前は num

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

- 値の初期化

```
int num[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

- 名前と添字で読み書き

```
System.out.printf("number of days %d/%d is %d\n", y, m, 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

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

```
public class Main
{
    public static void main(String[] args) {
        double p;
        int i;
        double u[] = {1.9, 2.8, 3.7};
        double v[] = {4.6, 5.5, 6.4};
        p = 0;
        for(i = 0; i <= 2; i++) {
            p = p + (u[i] * v[i]);
        }
        System.out.printf("p(product) = %f¥n", p);
    }
}
```

配列への書き込み

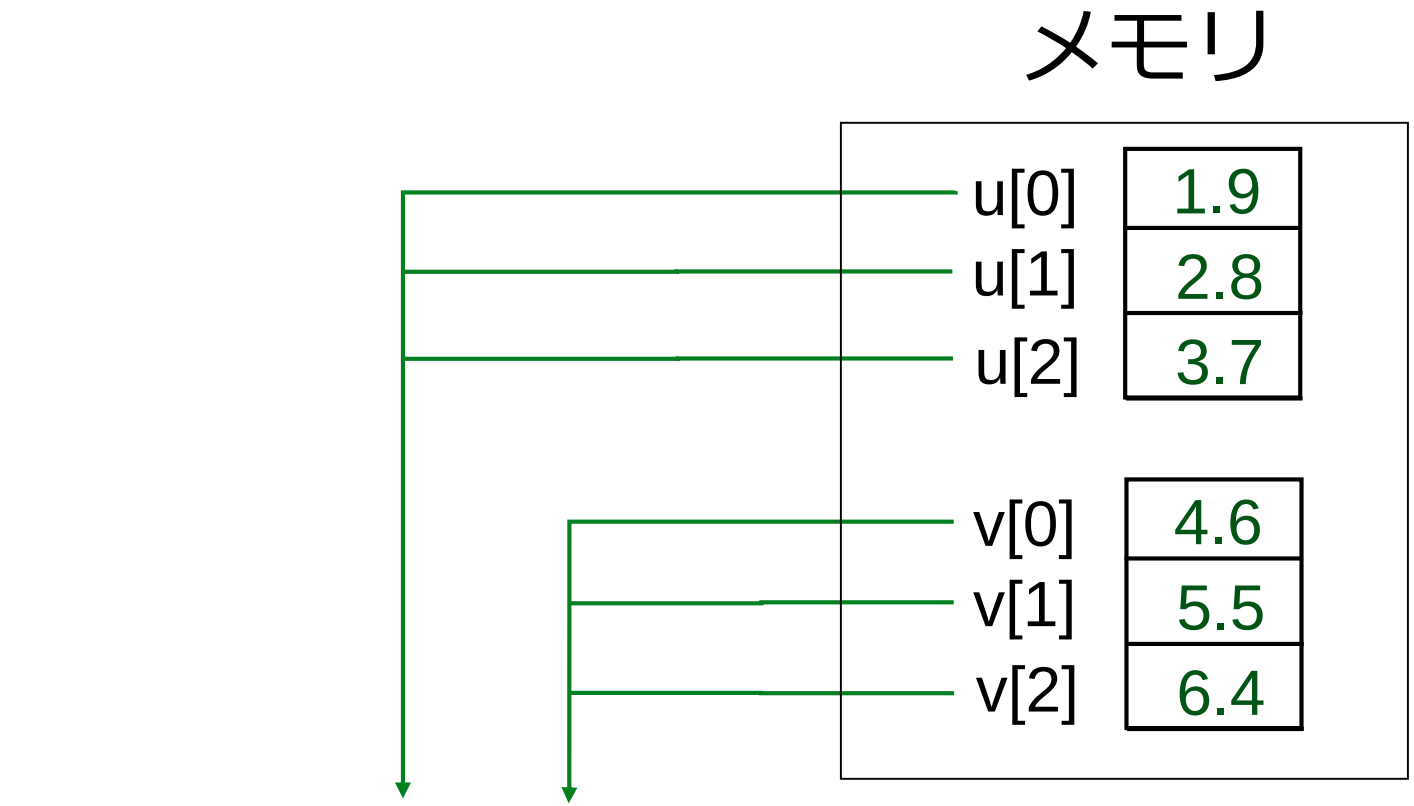
配列からの読み出し

ベクトルの内積

実行結果の例

```
p(product) = 47.820000
```

プログラムとデータ




```
p = p + (u[i] * v[i]);
```

配列からの読み出し

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	p = p + 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	p = p + 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	p = p + 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

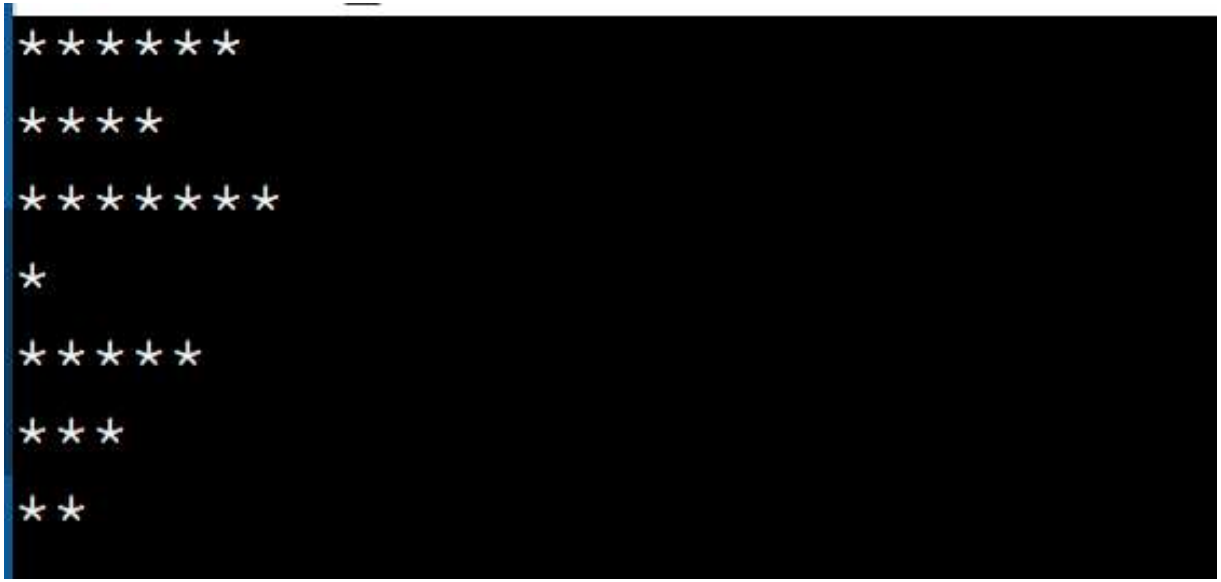
```
public class Main
{
    public static void main(String[] args) {
        int i, j;
        int a[] = {6, 4, 7, 1, 5, 3, 2};
        for(i = 0; i < 7; i++) {
            for(j = 1; j <= a[i]; j++) {
                System.out.printf("*");
            }
            System.out.printf("¥n");
        }
    }
}
```

配列への書き込み

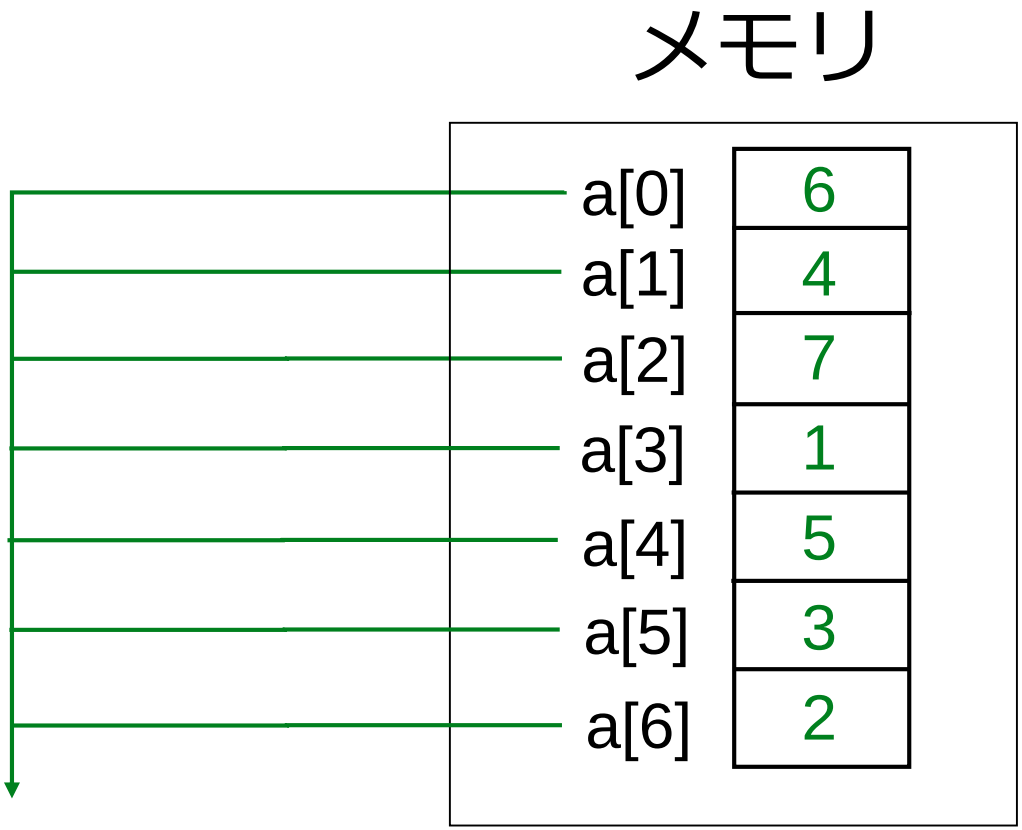
配列からの読み出し

棒グラフを描く

実行結果の例



プログラムとデータ




```
for(j = 1; j <= a[i]; j++) {
```

配列からの
読み出し

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, $5 + 6x + 3x^2$

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

<u>計算手順</u>

① an

② an-1 + an・x

③ an-2 + (an-1 + an・x) x

・・・ （a_0 まで続ける）

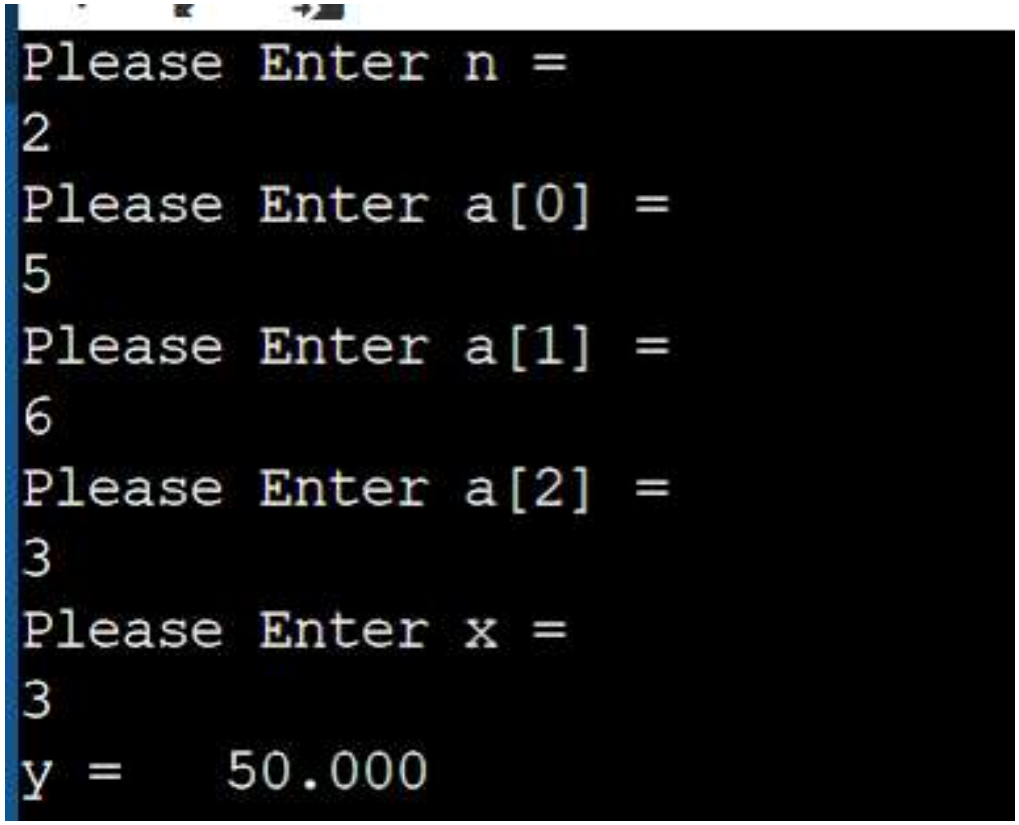
```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        int i, n;
        double x, y;
        double a[] = new double[20];
        Scanner s = new Scanner(System.in);
        System.out.println("Please Enter n =");
        n = s.nextInt();
        for(i = 0; i <= n; i++) {
            System.out.printf("Please Enter a[%d] =¥n", i);
            a[i] = s.nextFloat();
        }
        System.out.println("Please Enter x =");
        x = s.nextFloat();
        y = a[n];
        i = n - 1;
        while(i >= 0) {
            y = y * x + a[i];
            i = i - 1;
        }
        System.out.printf("y = %8.3f", y);
    }
}
```

配列への書き込み

配列からの読み出し

配列からの読み出し

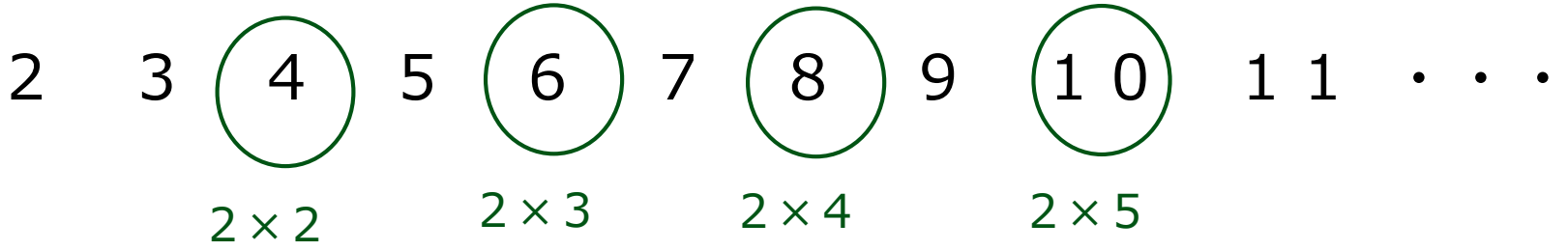
実行結果の例


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

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**素数**を求め，結果を表示するプログラムを作成する
  - **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 (1/3)



まず，２の倍数を消す

エラトステネスのふるい (2/3)

次に，３の倍数を消す

エラトステネスのふるい (3/3)

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）

```
import java.lang.Math;
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        int i, n, max;
        int p[] = new int[100000];
        Scanner s = new Scanner(System.in);
        System.out.println("Please Enter max =");
        max = s.nextInt();
        for(i = 1; i <= max; i++) {
            p[i] = 1;
        }
        n = 2;
        while(n <= Math.sqrt(max)) {
            i = 2;
            while((n * i) <= max) {
                p[n * i] = 0;
                i = i + 1;
            }
            n = n + 1;
            while((p[n] == 0) && (n <= Math.sqrt(max))) {
                n = n + 1;
            }
        }
        for(i = 2; i <= max; i++) {
            if(p[i] == 1) {
                System.out.printf("%d, ", i);
            }
        }
    }
}
```

p[1] から a[100] までを「１」にセット

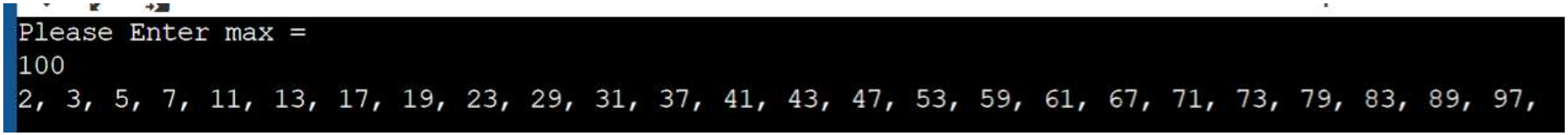
まず「２」の倍数は，素数ではない

n の倍数は素数ではない

n の「次」の素数を探す

求めた素数の表示

実行結果の例


Please Enter max =
100
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,