

2. 行列Aを用いて配列 array を初期化しなさい。次に、同じ大きさの4行4列の配列 t_array を作成しなさい。配列 array の転置行列 tA を配列 t_array に求めるコードを書きなさい。転置行列とは、i 行 j 列の値と j 行 i 列の値を入れかえた行列です。

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

(実行例)

転置行列を求めます

行列：

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

転置行列：

1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

3. 行列A,Bを用いてそれぞれ配列 array_A と array_B を初期化しなさい。その積を3行3列の配列 array_AB に求めるコードを書きなさい。

$$A = \begin{pmatrix} 1 & 2 & 1 & -3 \\ -2 & 1 & -2 & 0 \\ 1 & -1 & 1 & -1 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 2 \\ 2 & -3 & -1 \\ -1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

(実行例)

行列の積を求めます

1 2 1 -3

-2 1 -2 0

1 -1 1 -1

X

1 1 2

2 -3 -1

-1 2 1

2 1 2

=

-2 -6 -5

2 -9 -7

-4 5 2

4. あるクラスの学生が3科目の試験を受けた。学籍番号と試験結果を配列に入力して、各学生について合計を求めよ。さらに、合計に応じて順位をつけて表の形で出力するプログラムを作成しなさい。但し、学生数は最初に入力すること。

(配列の構成) 学生数が num の場合 :

```
String[] student_ID = new String[num]; // 各学生の学籍番号
int[][] score_table = new int[num][3]; // 各学生の3科目の点数
int[] total = new int[num]; // 各学生の点数の合計
int[] ranking = new int[num]; // 順位
```

(実行例)

成績処理を行います

学生数を入力してください

3 ←(入力して Enter キーを押す)

1 人目 :

学籍番号?>A001 ←(入力して Enter キーを押す)

科目 1 の点数?>47 ←(入力して Enter キーを押す)

科目 2 の点数?>9 ←(入力して Enter キーを押す)

科目 3 の点数?>63 ←(入力して Enter キーを押す)

2 人目 :

学籍番号?>A002 ←(入力して Enter キーを押す)

科目 1 の点数?>54 ←(入力して Enter キーを押す)

科目 2 の点数?>90 ←(入力して Enter キーを押す)

科目 3 の点数?>89 ←(入力して Enter キーを押す)

3 人目 :

学籍番号?>A003 ←(入力して Enter キーを押す)

科目 1 の点数?>12 ←(入力して Enter キーを押す)

科目 2 の点数?>34 ←(入力して Enter キーを押す)

科目 3 の点数?>87 ←(入力して Enter キーを押す)

番号	科目 1	科目 2	科目 3	合計	順位
A001	47	9	63	119	3
A002	54	90	89	233	1
A003	12	34	87	133	2

5. 4行13列の数値パターンで int 型の2次元配列を初期化しなさい。その後、配列に格納された0から9の各数値を以下の1文字で置き換えて画面に表示しなさい。このとき、改行は1行毎に入れて下さい。

(数値パターン)

```
0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0
2, 3, 4, 1, 5, 6, 0, 2, 3, 4, 0, 5, 6
7, 1, 8, 8, 6, 0, 0, 7, 1, 8, 8, 6, 0
0, 8, 3, 9, 0, 0, 0, 0, 8, 3, 9, 0, 0
```

(数値と文字の対応)

数値	文字
0	[空白/スペース]
1	-
2	<
3	' ← (表示は 이스케이프シーケンスで ¥' とします)
4)
5	,
6	/
7	(
8	=
9	-

6. 次の足し算ドリルと解答結果を用いて5行3列の2次元配列を初期化しなさい。この配列を読み込み、答えが正しければ○を、誤っている場合には×を実行例のように出力しなさい。

(足し算ドリル)

```
2 + 3 = 5
6 + 8 = 11
-5 + 2 = -3
7 + 7 = 14
-2 + (-6) = 8
```

(2次元配列表現)

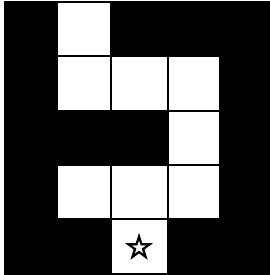
```
2 3 5
6 8 11
-5 2 -3
7 7 14
-2 -6 8
```

(実行例)

```
○ 問1 2+3=5
× 問2 6+8=11
○ 問3 -5+2=-3
○ 問4 7+7=14
× 問5 -2+-6=8
```

7. 次の迷路を 5 行 5 列の int 型の 2 次元配列 map で表現しなさい。各セルは下に示すように整数と対応付けることとします。

(迷路)



☆はゴールです

(整数との対応)

□ → 0
■ → 1
☆ → 2

(迷路の 2 次元配列表現)

```
int[][] map={  
    {1, 0, 1, 1, 1},  
    {1, 0, 0, 0, 1},  
    {1, 1, 1, 0, 1},  
    {1, 0, 0, 0, 1},  
    {1, 1, 2, 1, 1}};
```

次にユーザの現在位置を 1 行 2 列の int 型の 1 次元配列 user_pos で表現します。最初ユーザは 0 行 1 列の位置にいるとし、以下のように初期化しておきます。

```
int[] user_pos={0,1};
```

〔1〕上のマップ配列 map とユーザ現在位置 user_pos を画面に表示しなさい。ここで、ユーザ現在位置は、'・' (点) として表現することとします。

(実行例 1)

```
■・■■■  
■   ■  
■■■ ■  
■   ■  
■■☆■■
```

〔2〕 キーボード入力により、ユーザを移動できるようにします。入力された文字により下の
ようにユーザを移動させます。壁がある場合は移動せずに同じ場所に留まります。もし、ゴ
ールに辿り着いたらプログラムを終了します。

(キーと移動方向の対応)

w → 上へ1マス移動
s → 下へ1マス移動
a → 左へ1マス移動
d → 右へ1マス移動

(実行例 2)

```
■・■■■  
■      ■  
■■■  ■  
■      ■  
■■☆■■
```

s

←(入力して Enter キーを押す)

```
■ ■■■  
■・  ■  
■■■  ■  
■      ■  
■■☆■■
```

d

←(入力して Enter キーを押す)

```
■ ■■■  
■・  ■  
■■■  ■  
■      ■  
■■☆■■  
:
```

a

←(入力して Enter キーを押す)

```
■ ■■■  
■      ■  
■■■  ■  
■・  ■  
■■☆■■
```

s

←(入力して Enter キーを押す)

```
■ ■■■  
■      ■  
■■■  ■  
■      ■  
■■・■■
```

ゴールです!!