

1. 次のチェッカーフラグのパターンを boolean 型の 10 行 10 列の 2 次元配列に格納しなさい。その後、配列に格納されたパターンを画面に表示するコードを書きなさい。

(パターンの作り方)

i 行 j 列の配列要素に、

- i+j の値が偶数の場合 → true を代入する
- i+j の値が奇数の場合 → false を代入する

とすればよい。ここで、true は□を表し、false は■を表す。また、偶数と奇数の区別は、剰余演算子%を用いればよい。

(実行例)

```
>java Assignment11_1
```

```

□■□■□■□■□■
■□■□■□■□■□
□■□■□■□■□■
■□■□■□■□■□
□■□■□■□■□■
■□■□■□■□■□
□■□■□■□■□■
■□■□■□■□■□
□■□■□■□■□■
■□■□■□■□■□
□■□■□■□■□■
■□■□■□■□■□

```

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。

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

(参考) $m \times n$ 行列 $A = (a_{ij})$ に対して $n \times m$ 行列 (a_{ji}) を行列 A の転置行列 tA という

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。

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

4. あるクラスの学生が国語、数学、英語の試験を受けた。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と平均を求めよ。さらに、成績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ようにすること。

(配列の構成例) 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
double[] average = new double[num];  // 各学生の点数の平均
int[] ranking = new int[num];         // 順位
```

(出力例)

```
ID Jap. Math. Eng. Total Ave. Rank
j2006001 34 100 66 200 66.7 2
j2006002 31 21 47 99 33.0 3
j2006003 98 79 85 262 87.3 1
```

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、改行は一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	1文字	数値	1文字
0	[空白] ^ - \]	5	,
1	_	6	/
2	<	7	(
3	' ← (表示は 이스케이프 シーケンスで '¥' とします)	8	=
4)	9	-

6. 次の足し算のドリルと解答を2次元配列として初期化しなさい。この配列を読み込み、答えが正しければOを、誤っている場合にはX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)

```
O 問1 2+3=5
X 問2 6+8=11
O 問3 -5+2=-3
O 問4 7+7=14
X 問5 -2+-6=8
```