

Lab 2 矩阵计算器

题目描述

你被要求设计一个计算器完成以下三项任务，本次Lab禁止调用Math包中的内容：

1. 给定两个矩阵 x, y ，计算 $x + y$ 的值；
2. 给定两个矩阵 x, y ，计算 x 乘以 y 的值；
3. 给定矩阵 x ，计算矩阵的转置 x^T ；

为了简单起见，本次矩阵都是方阵，即矩阵行列数相同。

输入格式

输入文件包含多组数据。

第一行表示询问次数 C ，表示本次测试的测试用例数目

之后为单个测试用例的内容：

测试用例第一行包含两个部分 T, S ，分别表示操作类型和矩阵尺寸。如果操作类型为1或2，则接下来的两行分别为 x 和 y 两个矩阵的一维展开数据。每一个数据包含 S^2 个数值，使用空格符进行分隔。

如果操作类型为3，则只存在一行数值。

输出格式

输出文件包括 C 行。

对于每个询问，输出一行答案。每行答案均为矩阵的一维展开形式，使用空格符进行分割。

样例 #1

样例输入 #1

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

样例输出 #1

```
10 10 10 10 10 10 10 10 10
30 24 18 84 69 54 138 114 90
```

样例解读 #1

第一行 2 表示本次询问有 C=2 个问题

接下来 1 3 表示第一个问题为矩阵加法，使用的数据为两个 3x3 的矩阵

下面两行分别表示两个矩阵 x 与 y

$$x = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, y = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$x + y = \begin{bmatrix} 9+1 & 8+2 & 7+3 \\ 6+4 & 5+5 & 4+6 \\ 3+7 & 2+8 & 1+9 \end{bmatrix} = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

第二个问题为两个矩阵的点乘

$$x = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, y = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$x \cdot y = \begin{bmatrix} 1 \cdot 9 + 2 \cdot 6 + 3 \cdot 3 & 1 \cdot 8 + 2 \cdot 5 + 3 \cdot 2 & 1 \cdot 7 + 2 \cdot 4 + 3 \cdot 1 \\ 4 \cdot 9 + 5 \cdot 6 + 6 \cdot 3 & 4 \cdot 8 + 5 \cdot 5 + 6 \cdot 2 & 4 \cdot 7 + 5 \cdot 4 + 6 \cdot 1 \\ 7 \cdot 9 + 8 \cdot 6 + 9 \cdot 3 & 7 \cdot 8 + 8 \cdot 5 + 9 \cdot 2 & 7 \cdot 7 + 8 \cdot 4 + 9 \cdot 1 \end{bmatrix} = \begin{bmatrix} 30 & 24 & 18 \\ 84 & 69 & 54 \\ 138 & 114 & 90 \end{bmatrix}$$

因此本样例输出为两行，分别为两个结果的一维展开，也就是从左往右，从上往下依次输出。

样例 #2

样例输入 #2

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

样例输出 #2

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

样例解读 #2

操作类型为 3，本次询问为矩阵转置操作，输入的矩阵为：

$$x = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

矩阵转置为：

$$x^T = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

即原有矩阵按对角线进行翻转，原有的行和列对换。

作业提交

本次作业提交一个java文件，为了方便oj评测，请同学们按照作业包中的目录与包名压缩后提交

本次作业主要为了让大家熟悉java不涉及oop的内容，**严禁抄袭**

参考阅读

源码与文档

jdk安装目录下可以找到很多有用的东西

- 可以找到 `src.zip`
- 通过README可以找到 `JDK Documentation`，特别是 `The Java Platform, Standard Edition API Specification`
 - 是典型的java doc

Java文件与类

- 如果有多个类，且没有public类，文件名可与任一类名相同
- 一个JAVA源文件最多只能有一个public类
- 如果一个Java源文件包含多个class，那么编译后会产生多个.class文件

从C到Java语法变化

类型

- 基本类型vs引用类型
- 基本类型变动
 - byte 8位带符号整数(-128 到 127)
 - char表示16位的单个Unicode字符
- 常见修饰符：public, protected, private, static, abstract, final。
- 常量：final修饰符修饰常量 `final datatype CONSTANT_NAME = value;`
- 字面值(literal)
 - 以两个单引号界定的单个Unicode字符。如:'男','女'
 - 可以用\uxxxx形式表示，xxxx为十六进制。如:'\u7537', '\u5973'
 - 转义字符表示：\n \t \b \r \f \ ' "

- 类型转换常用的库函数
 - parseInt()
 - parseXX()
 - String s = number + "";

命名

java标识符可以用\$开头，也可以在组成中包含之

输入输出

- System.out是标准输出流OutputStream的对象
 - println() 是该对象的一个方法，该方法向标准输出流（显示屏）**显示字符**
 - printf() 是类似C中printf()的方法来格式化输出
- System.in：标准输入流类InputStream的对象
- Scanner类（需要导入包java.util.Scanner）
 - Scanner scanner = new Scanner(System.in);
(构造函数Scanner的参数类型也可可为java.io.File)
方法主要为 nextXX()、next()

命名规范

一种可供参考的改善可读性的命名习惯

- 类名的每个单词的首字母大写：使用UpperCamelCase(驼峰式命名法)
- 变量和方法名使用小写，如果有多个单词，第一个单词首字母小写，其它单词首字母大写。
- 常量使用大写，单词间以下划线分隔。

流程控制

- &, |为无条件逻辑运算符，左右都要计算
- switch语句的判断条件expression的计算结果不能大于int

Java数组

- 数组变量是引用类型的变量，声明数组引用变量并不分配数组内存空间。必须通过new实例化数组来分配数组内存空间。

```
datatype[][] arrayRefVar; //提倡的写法：类型在前，[ ]在后
```

```
arrayRefVar = new datatype[arraySize][arraySize];
```

- 初始化

```
//方法1
double[ ] myList = {1.9, 2.9, 3, 3.5} //可以将int转化为double类型，这时不用指定维度size

//方法2

double[ ] myList;
myList = new double[ ] {1.9, 2, 3.4, 3.5} //可以将int转化为double类型，声明和创建不在一条
语句时，不能直接用{ }来初始化
```

- arrayRefVar.length可以获取数组大小