牛客网-华为机试练习题 70

题目描述

- 如果A是个x行y列的矩阵，B是个y行z列的矩阵，把A和B相乘，其结果将是另一个x行z列的矩阵C。这个矩阵的每个元素是由下面的公式决定的：

原型：

voidmatrix_multiply(int *m1,int *m2,int *r, int x, int y, int z);

输入参数：

int *m1：x行y列的矩阵(array1[x][y])

int *m2：y行z列的矩阵(array2[y][z])

int x：矩阵m1的行数

int y：矩阵m1的列数/矩阵m2的行数

int z：矩阵m2的列数

输出参数：

int *r：矩阵m1, m2相乘的结果(array3[x][z])

返回值：

void

输入描述:

输入说明：
1、第一个矩阵的行数
2、第一个矩阵的列数和第二个矩阵的行数
3、第二个矩阵的列数
4、第一个矩阵的值
5、第二个矩阵的值

输出描述:

输出两个矩阵相乘的结果

示例1

输入

```
2
2
2
3 8
8 0
9 0
18 9
```

输出

```
171 72
72 0
```

解决代码:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (sc.hasNext()) {
            int r1, c1, r2, c2;
            r1 = Integer.parseInt(sc.nextLine());
            c1 = Integer.parseInt(sc.nextLine());
            r2 = c1;
            c2 = Integer.parseInt(sc.nextLine());

            int[][] x = new int[r1][c1];
            int[][] y = new int[r2][c2];

            for (int i = 0; i < r1; i++) {
                String[] str = sc.nextLine().split(" ");
                for (int j = 0; j < str.length; j++) {
                    x[i][j] = Integer.parseInt(str[j]);
                }
            }

            for (int i = 0; i < r2; i++) {
                String[] str = sc.nextLine().split(" ");
                for (int j = 0; j < str.length; j++) {
                    y[i][j] = Integer.parseInt(str[j]);
                }
            }
            print2DArray(matrixMultiplication(x, y));
        }
        sc.close();
    }

    private static int[][] matrixMultiplication(int[][] x, int[][] y) {
        int r1 = x.length, c1 = x[0].length, r2 = y.length, c2 = y[0].length;
        int[][] result = new int[r1][c2];
        for (int i = 0; i < r1; i++) {
            for (int j = 0; j < c2; j++) {
                result[i][j] = getResult(x, i, y, j);
            }
        }
        return result;
    }

    private static int getResult(int[][] x, int r, int[][] y, int c) {
        int r1 = x.length, c1 = x[0].length, r2 = y.length, c2 = y[0].length;
        int[] a = new int[c1];
        int[] b = new int[r2];

        for (int i = 0; i < c1; i++) {
            a[i] = x[r][i];
            b[i] = y[i][c];
        }
        int sum = 0;
```

```
    for (int i = 0; i < c1; i++) {
        sum = sum + a[i] * b[i];
    }
    return sum;
}

   public static void print2DArray(int[][] a) {
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length - 1; j++) {
                sb.append(a[i][j]).append(" ");
            }
            sb.append(a[i][a[i].length - 1]).append("\n");
        }
        System.out.println(sb.toString().substring(0, sb.length() - 1));
    }

}
```