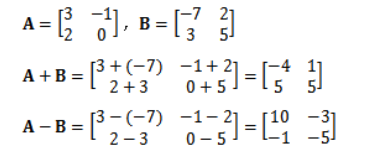
**任务描述**

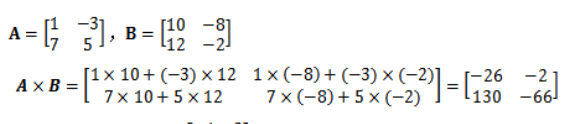
本关任务：矩阵是表示数据的常用的形式，比如图片可表示成矩阵的形式。方阵是行数等于列数的矩阵，行数和列数可用n表示。矩阵的数据data可使用二维列表表示，本关任务实现两个方阵的运算（加、减和乘）和方阵的转置运算。

**相关知识**

方阵的加减法：将方阵对应位置上的元素做加法或者减法。



方阵的乘法：拿第一个矩阵的每一行行与第二个矩阵的每一列相乘。



**（注：只有拥有相同行列数的方阵才可以进行加、减和乘运算，方阵形式不同不能进行运算）**

矩阵的转置：

**编程要求**

根据提示，在右侧编辑器 Begin-End 区间补充代码，实现方阵类FMatrix

**提示**

1. 构造方法可以传入方阵行列数n与数据data。data用二维列表表示，必须与n对应。
2. 实现类中的特殊方法：\_\_add\_\_、\_\_sub\_\_、\_\_mul\_\_和transpose转置方法,运算结果返回计算得到的FMatrix对象（若两个方阵形式不同返回字符串Wrong Computed!）。
3. 可用zip函数实现转置与矩阵乘

**测试用例说明**

测试输入：A=[[0,1,2],[3,4,5],[6,7,8]] B=[[1,0,0],[0,1,0],[0,0,1]]

预期输出：

A转置：[[0,3,6],[1,4,7],[2,5,8]] B转置：[[1,0,0],[0,1,0],[0,0,1]]

A+B:[[1,1,2],[3,5,5],[6,7,9]] A-B:[[-1,1,2],[3,3,5],[6,7,7]]

A\*B: [[0,1,2],[3,4,5],[6,7,8]]

测试输入：A=[[0,1,2],[3,4,5],[6,7,8]] B=[[1,0],[0,1]]

预期输出：

A转置：[[0,3,6],[1,4,7],[2,5,8]] B转置：[[1,0],[0,1]]

A+B: Wrong Computed! A-B: Wrong Computed!

A\*B: Wrong Computed!

|  |  |  |
| --- | --- | --- |
| 序号 | 矩阵A | 矩阵B |
| 1 | -52 -15 -53 60 -97  34 96 78 -98 -13  75 18 -66 -43 -69  76 -65 54 -56 13  5 -18 71 11 -65 | -22 -59 -70 18 -14  -85 -61 -37 13 -50  -44 -55 -97 -40 65  34 16 100 35 -94  27 80 -4 31 -2 |
| 2 | 28 -83 25 -63 -3  -97 -65 -68 9 -21  83 65 -95 -66 -74  41 32 -79 -2 -27  -14 81 -45 11 -7 | 31 -94 18 80 -9  46 -17 54 59 55  -1 -60 -12 -21 75  9 32 93 46 100  -15 5 21 -89 61 |
| 3 | 11 -43 5 -11  -92 -87 -18 -39  31 15 78 97  100 81 -28 -80 | 42 -87 90 -21  -64 -92 -10 43  -49 93 69 -44  91 -96 11 20 |
| 4 | 63 -2 -89  87 31 73  76 -66 90 | 99 23 -24  73 57 58  -28 -77 78 |
| 5 | -28 35 44  12 -52 -32  90 -100 -90 | -17 -58  -49 76 |

测试用例预期输出：

|  |
| --- |
| -52 34 75 76 5  -15 96 18 -65 -18  -53 78 -66 54 71  60 -98 -43 -56 11  -97 -13 -69 13 -65  -22 -85 -44 34 27  -59 -61 -55 16 80  -70 -37 -97 100 -4  18 13 -40 35 31  -14 -50 65 -94 -2  -74 -74 -123 78 -111  -51 35 41 -85 -63  31 -37 -163 -83 -4  110 -49 154 -21 -81  32 62 67 42 -67  -30 44 17 42 -83  119 157 115 -111 37  119 73 31 -3 -134  42 -81 -46 -91 107  -22 -98 75 -20 -63  4172 98 15724 82 -7413  -16023 -14760 -23246 -5093 9032  -3601 -8101 -3538 580 -2060  -76 -3345 -13805 -3194 10934  -3085 -8126 -5211 -4614 4541  28 -97 83 41 -14  -83 -65 65 32 81  25 -68 -95 -79 -45  -63 9 -66 -2 11  -3 -21 -74 -27 -7  31 46 -1 9 -15  -94 -17 -60 32 5  18 54 -12 93 21  80 59 -21 46 -89  -9 55 75 100 61  59 -177 43 17 -12  -51 -82 -14 68 34  82 5 -107 -87 1  50 64 14 44 73  -29 86 -24 -78 54  -3 11 7 -143 6  -143 -48 -122 -50 -76  84 125 -83 -45 -149  32 0 -172 -48 -127  1 76 -66 100 -68  -3497 -4752 -10200 -5813 -9425  -5533 14486 -4044 -7884 -8183  6174 -5689 -1548 16020 -15411  3209 143 2661 9138 -6381  3541 2956 5538 5733 1879  11 -92 31 100  -43 -87 15 81  5 -18 78 -28  -11 -39 97 -80  42 -64 -49 91  -87 -92 93 -96  90 -10 69 11  -21 43 -44 20  53 -130 95 -32  -156 -179 -28 4  -18 108 147 53  191 -15 -17 -60  -31 44 -85 10  -28 5 -8 -82  80 -78 9 141  9 177 -39 -100  1968 4520 1644 -2520  -963 18078 -9081 -1797  5347 -6135 9089 -1498  -6892 -11076 5378 1015  63 87 76  -2 31 -66  -89 73 90  99 73 -28  23 57 -77  -24 58 78  162 21 -113  160 88 131  48 -143 168  -36 -25 -65  14 -26 15  104 11 12  8583 8188 -8570  8832 -1853 5404  186 -8944 1368  -28 12 90  35 -52 -100  44 -32 -90  -17 -49  -58 76  Wrong Computed!  Wrong Computed!  Wrong Computed! |