

# 第一部分：实现读入两个矩阵的功能

1. 延续前几周的作业，让你的程序支持读入两个矩阵，矩阵维度是 $m \times n$ 和 $n \times k$ ，矩阵可以自己生成（建议自己写个小程序，要求生成一个 $3 \times 3$ 的矩阵，一个 $100 \times 100$ 的矩阵，以及一个 $1000 \times 1000$ 的矩阵）

2. 实现读入的两个矩阵相乘的功能

2. 至少运行 $3 \times 3$ ， $100 \times 100$ ， $1000 \times 1000$ 三个例子，统计计算的总时间（用自己写的计时器），将源程序和计算结果提交

## 第二部分：采用BLAS实现矩阵相乘

1. 自己下载和编译BLAS，在Makefile中链接BLAS

2. 写自己程序和BLAS库的接口：先写函数声明再写引用，为程序接口写适当注释  
(英文注释)

3. 采用BLAS实现矩阵相乘，统计计算时间。并随着不同尺寸的大小，比较和自己实现的矩阵乘法的效率差别

```
Mat_Demo operator*(const Mat_Demo &m1, const Mat_Demo &m2)
{
    assert(m1.nc() == m2.nr());

    cout << " m1.nc=" << m1.nc() << endl;
    cout << " m2.nr=" << m2.nr() << endl;

    Mat_Demo m3(m1.nr(), m2.nc(), true);
    m3.name="m3";

    for(int i=0; i<m1.nr(); ++i)
    {
        for(int j=0; j<m2.nc(); ++j)
        {
            for(int k=0; k<m1.nc(); ++k)
            {
                m3(i,j) += m1(i,k) * m2(k,j);
            }
        }
    }

    return m3;
}
```

## 第三部分：实现读入一个实对称矩阵的功能

1. 让你的程序能读入一个实数对称矩阵
2. 让你的程序调用LAPACK来对角化这个实数矩阵
3. 读入一个 $3 \times 3$ 的矩阵，输出特征值和特征向量，验证LAPACK程序求解的正确性
4. 读入一个 $100 \times 100$ 的矩阵和一个 $1000 \times 1000$ 的矩阵，统计计算的总时间（用自己写的计时器），将源程序和计算结果提交