



# Engineering Applications General C++

Exercise

## > 2. Function

### 1. Explicit Euler Method

1. 编写一个函数，能够计算一个2x2矩阵和2x1向量的乘积；

```
double * matrix_product(double * mat_22, double * vec_2);  
(仿照两个向量求和的写法)
```

2. 编写一个函数，能够按行按列显示一个2x2矩阵；

```
void print_mat22(const double *mat);  
(使用setw(int)设置输出语句的宽度，需要#include <iomanip>; )
```

3. 编写一个函数，能够按列显示一个2x1的向量；

```
void print_vec2(const double *vec);
```

4. 编写一个函数，能够根据给定的步长、步数、系统矩阵、初始向量计算显式欧拉法的结果（结果是一个2x1的向量）；

```
double * explicit_euler(double * mat_22, double * vec_2,  
double h, int count);
```

## > 2. Function

### 1. Explicit Euler Method

5. 在主函数中，给定系统矩阵为： $\begin{bmatrix} 0.0 & 1.0 \\ -\frac{9.81}{0.2} & 0.0 \end{bmatrix}$ ，初始向量为 $\begin{pmatrix} \frac{\pi}{6} \\ 0.0 \end{pmatrix}$ ，步长为0.0005，计算10000次显式欧拉法之后的结果；

6. 尝试能不能在显式欧拉法的计算函数中，调用自己编写的向量和矩阵运算函数；

Tips: 显式欧拉法计算函数可以用递归完成；

可以对比Modelbildung und Simulation的Übung 6 (Pool)检查结果的正确性。

## > 2. Function

### 2. Invert Character Array

编写一个函数，能够将字符串倒序输出；

```
char * invert_char(char * str);
```

示例：invert\_char( "general C++" )应返回 "+ +C lareneg"

\* 可能用到的函数：int strlen(char \*)，该函数可以返回一个字符串的长度(不包括空字符 '\0' )

\* 函数实现部分采用双指针指向字符串头尾，可以简化处理过程。