

## Project 编译期长整数加法

这门课之前设计了一个长整数加法的项目。该项目是在运行期接收用户的输入并完成的计算。现在，我们希望将整个计算过程放到编译期完成，即实现一个编译期的长整数加法。

考虑到大家接触元编程的时间相对较短，因此我们在这里对长整数加法的实现进行了简化：这里只考虑两个 10 进制非负整数的相加问题。比如，对于如下的定义：

```
template <unsigned int...> class Cont;  
using res = Add<Cont<1, 9, 9>, Cont<1>>;
```

res 的结果应为 Cont<2, 0, 0>——即表示 199 加 1 等于 100。

注意，我们假定容器中接收的都是 unsigned 类型的数据，同时每个数据的取值只能是 0~9。

为了便于结果验证，我们需要首先编写一个函数模板 print，给定一个编译期数组，将其中的值打印出来。在此基础上，就可以实现长整数加法了。进一步，为了简化大家的实现过程，我们可以将整个过程简化成如下几步：

1. 实现一个数组翻转的元函数，将输入的数组翻转，从而使得低位位于数组的开头，高位位于数组的结尾。
2. 实现加法逻辑，计算后的结果依旧是低位位于数组的开头，高位位于数组的结尾。
3. 调用之前实现的数组翻转元函数，将结果数组翻转，确保高位位于数组的开头，低位位于数组的结尾。
4. 调用之前实现的 print 函数，打印输出结果。

扩展：如果你有兴趣，可以考虑实现另一个元函数，给定一个数组与两个数 m/n，返回一个新的数组。新的数组与元数组表示相同的数值，但元数组是 m 进制的（每个数字的取值是 0~m-1），而新数组是 n 进制的（每个数字的取值是 0~n-1）。