



深蓝学院
shenlanxueyuan.com

项目三：编译期长整数加法



主讲人 李潇楠



作业要求

●基础：编译期长整数加法，非负整数十进制

1. 输出

2. 数组反转

3. 加法

●扩展：编译期长整数进制转换

- 定义一个结构体模板Cont，后续的所有操作都是基于它来完成。它的type是整型变长参数。对于长整数147258可以表示为Cont<1,4,7,2,5,8>
- 接下来是对Cont进行循环输出，每次从变长参数中分离出第一个元素，直至“递归”的终点。使用fold expression更简洁。

输出

```
29  /* Print */
30
31  template<typename T>
32  struct PrintHelper;
33
34
35  template<template<unsigned...> class TCont, unsigned... vals>
36  struct PrintHelper<TCont<vals...> >{
37      static void Exec(){
38          ((std::cout<<vals), ...);
39      }
40  };
41
42  template <typename T>
43  void print(){
44      PrintHelper<T>::Exec();
45  }
```

```
2  //fold expression -> unfold
3  template <typename T>
4  void print();
5
6  template<template<unsigned...> class TCont>
7  struct PrintHelper<TCont> >{
8
9      static void Exec(){
10      }
11  };
12
13  template<template<unsigned...> class TCont,unsigned val,unsigned... vals>
14  struct PrintHelper<TCont<val,vals...> >{
15
16      static void Exec(){
17          std::cout<<val;
18          print<TCont<vals...> >();
19      }
20  };
```

数组反转

- 这里需要创建一个空数组放在最前面<>

也就是输入<>,<remainder...> 输出 <processed...>

- 而中间的过程类似这样

<processed...>,val,<remainder...> -> <val,processed...>,<remainder...>

直到remainder为空，这样数组就反转过来了

- 由于输入的数组a和b可能不一样长，在特化时需要把情况分清楚。比如， $a = \text{Cont}\langle 1, 2, 3 \rangle$, $b = \langle 1, 2 \rangle$ ，处理到第三位时b已经无法分离出参数了。
- 那么可以分四种情况：a或b在当前位是否还有数字。
- 而如果计算到最高位还要进位，需要在结果的最高位放一个1，反之则不用。这样看应该是五种情况，分别去特化即可。

- 如果能实现基础部分，这个扩展不会太困难，参照第一次作业短除法的实现，灵活运用循环的元编程方式。
- 需要注意的是，每次做完短除法，被除数的前导0应该全部删除。

Q&A

感谢各位聆听 !
Thanks for Listening

