

# 代码优化实验指导

## 实验说明

本实验包含三个代码文件：main.cpp、opt.cpp和opt.h，opt.h中声明了一系列程序所需变量和函数，其含义如下：

- Table：一个用结构体表示的二维表格，num表示该表格行数，quantity、price、date、discount、tax指针分别指向该表的一列内容首部。也就是说，该表行数为num，由quantity、price、date、discount、tax五列组成。结构体定义如下：

```
struct table
{
    size_t num;
    int* quantity;
    int* price;
    int* date;
    double* discount;
    double* tax;
};
```

- calculate函数：该函数接受一个参数int n，筛选出Table中满足 $date \leq n$ 的所有行（记其在表中的行序号为 $a_1、a_2、a_3.....a_m$ ），并对这些行进行下列运算，结果存储到对应的变量中：
  - $SumQuantity$ :  $\sum_{k=1}^m quantity[a_k]$
  - $SumBasePrice$ :  $\sum_{k=1}^m price[a_k]$
  - $SumDiscPrice$ :  $\sum_{k=1}^m price[a_k] \times (1 - discount[a_k])$
  - $SumCharge$ :  $\sum_{k=1}^m price[a_k] \times (1 - discount[a_k]) \times (1 + tax[a_k])$
  - $AvgQuantity$ :  $(\sum_{k=1}^m quantity[a_k]) \div num$
  - $AvgPrice$ :  $(\sum_{k=1}^m price[a_k]) \div num$
  - $AvgDiscount$ :  $(\sum_{k=1}^m discount[a_k]) \div num$
  - $total$ :  $m$
- calculate1函数和calculate2函数的作用一致，区别在于calculate1函数**没有使用SIMD指令**，calculate2函数**需要使用SIMD指令**。

## 实验要求

你需要优化已给出的calculate1函数并实现calculate2函数，使其在输出正确结果的情况下拥有尽可能短的运行时间。**你只能更改calculate函数的内容，增加或删除如getdata一样的辅助函数，不允许修改其他内容。**

## 实验测试

- 在本地测试中，你可以使用如下指令来生成程序并运行：

```
g++ opt.cpp main.cpp -o main
./main
```

- 在最终的评测中，我们会修改main.cpp的内容，包括但不限于修改make函数、修改给calculate函数传递的参数、添加校验你计算结果是否正确的代码，并与你提交的opt.cpp、opt.h进行编译运行。
- date的值域是[1, 100]且date列的值均匀分布，因此 $n/100$ 代表选择率，我们测试时将测试三组：低选择率5%~10%，中选择率：40%~60%，高选择率：90%~95%，测试时从三个范围中任选一个值作为输入控制。

## 特别提示

- 我们提供的程序中并没有判断你计算结果是否正确的代码，因此请自行验证你的计算结果是否正确，若结果不正确可能会损失较多分数。