Application Requirements

In this lesson, we will look at the requirements of an application which makes computations on the data from a CSV file.

Imagine you're working with some sales data and one task is to calculate a sum of orders for some products. Your shopping system is elementary, and instead of a database, you have CSV files with the order data. There's one file per product.

For example, here are the book sales:

date	coupon code	price	discount	quantity
5-12-2018		10.0	0	2
5-12-2018		10.0	0	1
6-12-2018	Santa	10.0	0.25	1
7-12-2018		10.0	0	1

Each line shows a book sale on a specific date. For example on 5th Dec, there were three sales, 10$ each, and one person bought two books. On 6th Dec we had one sale with a coupon code.

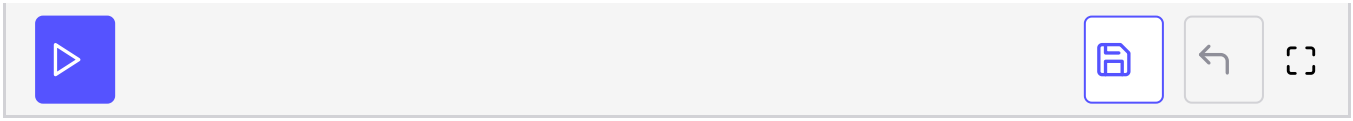
The data is encoded as a CSV file: `sales/book.csv`:

```
5-12-2018;;10.0;0;2;
5-12-2018;;10.0;0;1;
6-12-2018;Santa;10.0;0.25;1;
7-12-2018;;10.0;0;1;
```

The application should read the data and then calculate the sum, in the above case we have the following code:

```
#include <iostream>
using namespace std;

int main() {
  double sum;
  sum = 10*2+10*1+       // 5th Dec
      10*(1-0.25)*1 +  // 6th Dec with 25% coupon
      10*1;            // 7th Dec
  cout << sum;
}
```



For the above sales data, the final sum is 47.5\$.

Here are requirements of the application we want to build:

- The app loads all CSV files in a given folder - read from the first argument in the command line
- The files might contain thousands of records but will fit into memory. There's no need to provide extra support for huge files
- Optionally, the app reads the start and the end date from the second and the third command-line argument
- Each CSV line has the following structure: `date; coupon code; unit price; quantity; discount;`
- The application sums all orders between given dates and prints the sum to the standard output

---

We'll implement the serial version first, and then we'll try to make it parallel.