element Array[ ][ ][ ]

v P D

(1)

Sum of all days in each Product.

int prA, prB, prC;

for sums all products, all days for one vendor, increment changes vendor.

for. do the sum, then the increment changes the product

for

Sum one products sales for ev. y da;

Find a way to assign each Product variable to it's ~~total~~ Sum

Output to file

Program workflow

- Start main.cpp
- open entrada.txt
- open [redacted] salida.txt
- Read entrada.txt
- Assign information to respective variables in array & private member variables. Validate data before entering.
- Run menu
- Take choice input
- Switch statement
    1. Total sales per product over all. (all days summed)
    2. Name of product with highest sales in a given day.
    3. List all vendors with given product's sales in a given day.

Program Workflow (cont)

- Validate choice main.cpp
- Run chosen function
- Once done, loop menu and switch
- If user chooses to end program, exit loop.
- close entrada.txt.
- close salida.txt
- End of program.

Note: Make main as simple as possible.

Reading file

- Declare int vendors, products, days
- in >> vendors, in >> products, in >> days;
- Skip a line

- Read product lines one at a time. Each time will have a series of for loops to assign product nos to all slots

```
000   100   200
001   101   201
002   102   202
 ↑ ↑ ↑
 v p d
```

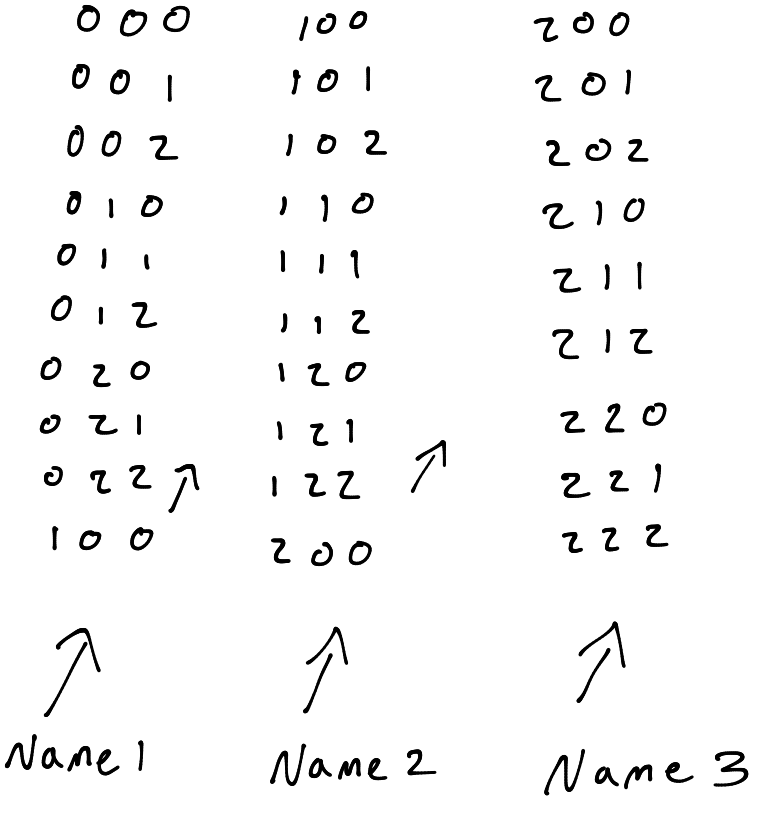
in >> PrName

ProductName = PrName;

- Skip a line

Reading file (cont)

- Read vendor names in a similar fashion as product names but with an additional loop.



Reading file (cont 2)

- Read Sale data in a similar fashion but with more loops. In the txt file, each block represents a vendor. Each line represents a product. Each column of the block is a day. Each numbers in a block is the number of sales of a product in a day.