CSCI 2170 OLA 5 (Due midnight, Thursday April 1st, Deadline Sunday April 3rd)

The *Bake My Day* bakery wishes to forecast the cost of producing various bakery products during the year. Each bakery product is produced using a different mix of raw ingredients. Moreover, because many of these ingredients are seasonal, the price of these ingredients varies from month to month. The bakery keeps a recipe file that is a table with one line per product that indicates the amount of ingredient needed for each product. This table looks like:

Bakery Recipe Table

Bakery Product	Ingredient 1	Ingredient 2	….	Ingredient N
A	--	--	--	--
B	--	--	--	--
…	--	--	--	--
M	--	--	--	--

The manager of the bakery also keeps a computerized spreadsheet that forecasts the price of ingredients for the various products by month as follows:

Ingredient Prices by Month

Ingredient	January	February	March	…	October
1	--	--	--	--	--
2	--	--	--	--	--
3	--	--	--	--	--
…	--	--	--	--	--
N	--	--	--	--	--

What the *Bake My Day* manager wants is a table forecasting the monthly cost of bakery products that resembles the following:

Monthly Cost of Bakery Products Forecast

Bakery Product	January	February	….	October
A	--	--	--	--
B	--	--	--	--
…	--	--	--	--
M	--	--	--	--

Write a C++ program to calculate and print the *Monthly Cost of Bakery Products Forecast* table using data from the files described below. All table values are non-negative integers.

A starter skeleton program, heavily stubbed, is available as **$PUB/bakeryStart.cpp**. Copy the skeleton and use it as your starting point**: $ cp $PUB/bakeryStart.cpp .**

Develop your code incrementally. The input files for this assignment will be "**products.dat**", "**ingredients.dat**", and "**prices.dat**".

- The first line of the **products.dat** file has the number (*M*) of product names found in the file. The remaining *M* lines of the file contain the names of all the bakery products, one product name per line. There will be at most 24 bakery products; use the symbolic constant `MAX_PRODUCTS` to control this. The name of each product is at most 12 characters long; use the symbolic constant `MAX_NAME_LEN` to control this.
- The first line of the **ingredients.dat** file has the number (*N*) of ingredients per line of the file. The remaining *M* lines of the file contain the *N* ingredient amounts needed for each product. If a particular ingredient is not needed for a product, the amount is set to zero. There will be at most 30 different ingredients; use the symbolic constant `MAX_INGREDIENTS` to control this.
- The first line of the **prices.dat** file has the beginning and ending months of the cost projections. These are expressed as an integer pair, where January is encoded as 1, February as 2, and so on. Note that the cost projection period may span the end of a year; thus the beginning month value might be bigger number than the ending month. (See example below.) The remaining *N* lines of the file contain the prices for ingredients for each of the months. There will be at most 12 months in a forecast; use the symbolic constant `MAX_MONTHS` to control this.

The files are related in a straightforward fashion. Thus, for example, the second line of the **prices.dat** file contains data for the product identified by the second line of the **products.dat** file. Copy the data files into the "OLA5" project directory with the following commands:

```
$cp ~cen/data/products.dat .
$cp ~cen/data/ingredients.dat .
$cp ~cen/data/prices.dat .
```

EXAMPLE: Suppose the data files appear as follows:

products.dat

```
3
Donut
Bagel
Kaiser Roll
```

ingredients.dat

```
5
0  10  15   5   1
0   6   3  10  20
3   6   3  10   4
```

prices.dat

```
11   2
```

```
10  10   5   1
30  10  15  20
50  25  25  35
40  20  40  30
20  10   8  10
```

Then the output (to stdout) should look something like:

```
Monthly Cost of Bakery Products Forecast

Product          11     12      1      2
-----------+------+------+------+------+
Donut          1270    585    733    885
Bagel          1130    535    725    725
Kaiser Roll     840    405    612    568
```

Makefile

For this assignment, you are required to write a makefile to compile the program. Refer to example makefiles discussed in class to write one for this program**.**

Program Submission

Before submission, generate a log file "ola5log" that contains the listing of the program, the compilation of the program, as well as running of the program. The makefile needs to be used to compile the program and generate the executable program.

script ola5log
pr -n -t -e4 change.cpp
pr makefile ← display the makefile
make
RunBakery ← this is assuming the name of your executable file is "RunBakery"
exit

Then, submit the program with the following handin command:

```
$ handin ola5 bakery.cpp ola5log
```