TRIAL 1- BASIC

This is fairly standard shopping data. The client want to process this kind of file on a regular basis looking back over the last year but the file is slow.

You have inherited this file (BigStore) from a departing colleague - it works but it's not too fast. IT takes about 2 mins 55 seconds on my computer to compute the slow way.

The data is stored int eh file scanner_data.csv - there is an example below

ID	Date	Customer_ID	Transaction_ID	SKU_Category	SKU	Quantity	Sales_Amount
1	02/01/2016	2547	1	X52	0EM7L	1	3.13
2	02/01/2016	822	2	2ML	68BRQ	1	5.46
3	02/01/2016	3686	3	0H2	CZUZX	1	6.35
4	02/01/2016	3719	4	0H2	549KK	1	5.59
5	02/01/2016	9200	5	0H2	K8EHH	1	6.88
6	02/01/2016	5010	6	JPI	GVBRC	1	10.77
7	02/01/2016	1666	7	XG4	AHAE7	1	3.65
8	02/01/2016	1666	7	FEW	AHZNS	1	8.21

The customer_ID links to a file which for privacy reasons you're never gonna be given access to. Products are identified by their category (SKU_Catagory) and number (SKU). For example X52 might be veg and 0EM7L might be the code for cabbage.

Normally we do this yourself but the way this table is organised. Is that if a customer buys more than one product quite common in a supermarket then the transaction ID and the customer ID will be the same. If a customer comes back and buys more products, they will be given a different transaction ID.

Clearly, there is quite a lot that can be derived from this information. For example, we can find everything that the customer has ever bought and work out who are the biggest spenders. The client might need to know this because they might want to offer regular customers discount coupons. Alternatively, they might want to offer one off customers special deals to make them buy more than one product and return to the store. We don't know and we are not business computing students.

There is also a date column which means we can see how much was produced in a day.

The current file goes a little slowly - you job is to make it go faster.

Q: It takes 5 mins to test each time - what can I do about this?

It is OK to use a cut down version of the file - say the first 300 lines. This is up to you to do something for testing purposes.

Q: How to know when I have made it fast enough?

The real world answer is when you can sit before a huge file and get it processed before you.
I would like to make software to help you (I might release this after the assessment is out) my experience has show that this can cause confusion - and this is bad enough.

When I run bigStore_student.py using on my machine I get

```
 2968603 function calls in 176.697 seconds
```

When I run the optimised version I get

```
 3425799 function calls in 10.748 seconds
```

So a speed increase of 176.697 / 10.748 which gives 16.4 or 1640% but this is on the small data set.

Notice I'm not timing loading and saving as this is very much effected by the speed of the computer and side of hard drive. We should really do this on PCs in the cluster.

Q: How to know if the result are correct ?

There are 3 files Best_customers_correct.csv, daily_sales_correct.cvs, prod_report_correct.csv these are tables in mark down format.

Your outputs should match these, its up to you to test. When we receive you files we will be testing

Q: Can I alter product, transaction , customer class?

Yes you can but I made all my changes in BigStore.py, you don't have to. Provided the functions speed up we don't care how.

Look on blackboard for the league table

You can do better than this (even with out a faster machine this was on my laptop) **but you don't need to**.

To help you I will put up a league table. This will have speeds for no improvement , Par (that is as good as my score) and beyond par which I am interested to see.

HideSmall.csv.mac.gz

No improvement	Pro result	High score
1	16	???