Programming assignment 4

* You need to write a code to simulate open-row and close row-policy. The DRAM has only one bank. Initially, the row-buffer is closed. Time starts from zero.

We will provide an input file which has many lines. Initial few lines of this file specify the timing parameters of DRAM and bank-dimensions.

Here is the order:

ROWBUF_HIT_LATENCY 30

ROWBUF_MISS_LATENCY 60

ROWBUF_CONFLICT_LATENCY 90

ROWS_IN_BANK 64

COLUMNS_IN_ROW 48

The order of these lines will remain same in any input file we will use. In other words, we will not first write miss latency and then write hit latency. This will simplify your code.

As for the DRAM access, each line has three entries separated by commas:

RowID,ColumnID,TimeOfArrival

The time of arrival is specified in the same unit as row-buffer hit latency, etc. and thus, your code need not change any units.

In the input file, lines beginning with // have to be ignored.

We will not test your code with any incorrect/infeasible input.

You have to write a code which prints the completion time of these accesses with both open-row and close-row policy. That means, in a single run, compute both policies. The completion time refers to the time when a request has been served (which may not be the same time when row buffer is closed).

Your submission file should be named RollNumber_DRAM_RowPolicies.cpp. It takes only one input from commandline which is the input file.

$./binaryName DRAM_Parameters_Accesses.txt

For this input, the output should be named:
RollNumber_DRAM_Parameters_Accesses.txt

Some sample input and output files are attached.

Use the following command to get the proper spaces. This command will print the first line of the output file.

```
  outFile<<"Row,Column
"<<setw(13)<<"TimeOfArrival"<<setw(15)<<"OpenRow"<<setw(10)<<"CloseRow"<<
setw(10)<<endl;
```

If still, the spaces are not correct, don't worry.

* No partial marking

* Any output on "cout" will not be evaluated. Only that in the output file will be evaluated.