CS 161 Lab 7
TA: Jose Rodriguez

Team Leader: Selik Samai
Team Members: James Hollister, Roberto Pasillas, Selik Samai

We load our trace file into a vector so that we only have to read from the file once instead of each iteration.

We use nested vectors to simulate the cache where the associativity determines the number of columns for the 2D matrix and the rows are determined by the (cache_size/ BLOCK_SIZE) / associativity.

Once we have calculated the matrix size, we then build cache and then read memory access from file where we determine if cache hits and miss.

We included a makefile that makes testing easier.

- *make* is used to compile the program. The object file is called cache.
- *./cache* [*your file*] runs the executable with your own specific trace file.
- *make run* will run the program with the included trace file
- *make clean* will remove the object file (cache).

```
~/Desktop/cs161/lab7
santo003@kilo-18 $ make run
g++     cache.cpp   -o cache
./cache trace

                LRU Replacement Policy
        1024    2048    4096    8192    16384
1       55.01   42.07   29.30   20.74   13.91
2       51.58   36.44   23.70   13.98   8.49
4       48.85   33.97   20.04   11.33   6.37
8       47.44   32.20   18.63   10.02   5.72

                FIFO Replacement Policy
        1024    2048    4096    8192    16384
1       55.01   42.07   29.30   20.74   13.91
2       53.31   38.32   25.47   15.37   9.49
4       51.86   37.07   23.11   13.67   7.86
8       51.14   35.98   22.40   12.79   7.44
~/Desktop/cs161/lab7
```

The left-most numbers indicate the set associativity while top most numbers indicate the cache size for each policy.

No known bugs.