```
CPS-360         Assignment #10,         10 points

Goal: To explore 4-way associative cache memory.

Constraint:
        - all modules must have parameters (exceptions as below)
        - do not prompt the user for input

Statement:

Properly documented, modular C-program (and Makefile) on the lines
given below.

Goal is to model:
 - L1, 4-way associative cache with 4096 cache-lines
 - data-block is 64 bytes (16 32-bit words)
 - address size 32 bits
 - explore two policies for choosing a victim for replacement
   (used only when all entries in the set-in-question are in use):
   1. choose a victim at random using random number generator
      (random number: in range 0..3 for 4-way)
   2. replacement policy is first-in-first-out (FIFO) algorithm:
      . on a miss oldest entry in the set is the victim for replacement,
        (any time there is a replacement on a miss, set the cache entry
        age to 0 and increment the age of all other entries in the set)
      . on a hit do nothing!

Note: Be aware if index of a set is sindex, then next-set is 4 entries away,
use sindex<<2 to find the correct address of set in cache!!!

Globals:

Struct for a cache line:

        #define SETS 1024
        #define LINES 4
        struct cacheline{
                char valid;             /* for valid/invalid entry */
                unsigned char age;      /* for age of entry */
                int tag;
        };
        typedef struct cacheline cache;
        cache   l1[SETS * LINES];
        int   reference, miss;          /* keep track of cache misses */



Sample input:
0x00000000
0x11100030
0x10000020
0x11000010
0x00000000
0x11100030
0x00000000
0xfff00030
0x00000000
0x10000020
0x12100042
0x12a00050
0x12300059
0x1240005a
0x12100059
```

```
0x1210005f
0x11100030
0x10000020
0x11000010

Sample output:

Fifo:   19 references, 8 misses
Random: 19 references, 12 misses

The program will be invoked as:

    ./nwaycache algo  < inputfile
        (eg., ./nwaycache fifo < datfile )

where inputfile is redirected to stdin (use scanf() to read address from stdin).

=========
Some insight using FIFO (example: not to be confused with sample output):


->> 4-way associative: tag 16 bits, setindex 10 bits, offset-in-block 6-bits.

address         tag     set     line    hit      cache action?
0x00000000      0       0       0       no      0 goes in set0-line0 (age0 0)
0x11100030      4368    0       1       no      4368  goes in set0-line1 (age0 1, age1 0)
0x10000020      4096    0       2       no      4096 goes in set0-line2 (age0 2, age1 1, age2
0)
0x11000010      4352    0       3       no      4352 goes in set0-line3 (age0 3, age1 2, age2
1, age3 0)
0x00000000      0       0               yes
0x11100030      4368    0               yes
0x00000000      0       0               yes
0xfff00030      65520   0       0       no      replaces 0 (age0 0, age1 3, age2 2, age3 1)
0x00000000      0       0       1       no      replaces 4368 (age0 1, age1 0, age2 3, age3 2)
0x10000020      4096    0               yes
0x12100042      4624    1       0       no      4624 goes in set1-line0 (age0)
0x12a00050      4768    1       1       no      4768 goes in set1-line1 (age0 1, age1 0)
0x12300059      4656    1       2       no      4656 goes in set1-line2 (age0 2, age1 1, age2
0)
0x1240005a      4672    1       3       no      4672 goes in set1-line3 (age0 3, age1 2, age2
1, age3 0)
0x12100059      4624    1               yes
0x1210005f      4624    1               yes
0x11100030      4368    0       2       no      replaces 4096 (age0 2, age1 1, age2 0, age3 3)
0x10000020      4096    0       3       no      replaces 4352 (age0 3, age1 2, age2 1, age3 0)
0x11000010      5352    0       1       no      replaces 65520 (age0 1, age1 3, age2 2, age3 1
)

=======

Try small/meaningful modules like:

    int main(void)
    int checkargs()
    void usage()
    int isahit(?)
    int isamiss(?)
    void initcache(void)
    void printrslts(?)
    void procesaref(?)
    void getnextref(?)
    void randomalgo(?)
    void fifoalgo(?)
    etc.
```

Turn-in: Submit via BB tape archive file globalid-a10.tar containing files:
a10/Makefile and a10/a10.c

Top level algorithm is simple:

```
    set-up and initialize structures and variables

    while more references to go:
       get next reference
       process next reference
       ...
```

To read using scanf() till end-of-file:

```
   int nextref;

   while(scnaf("%x", &nextref) != EOF)
        {
                /* process nextref */
        }
```
