Assignment 1, Code

- As background to review the solution

```
lastTrans = 1;
...
int start()
{
if (lastTrans < 0) return -1
else
   {
      lastTrans = -(++lastTrans);
            return lastTrans;
   }
}
```

```
int read(int blockAddr, int tId, int *block)
{
  /* find the cache element e containing the block whose
     disk address is blockAddr */
  if (there is such a cache element e) {
      /* the disk block at blockAddr is in cache */
      Cache(e).tId = tId;
      block = &Cache(e).newBlock;
     /* &Cache(e).newBlock = address of Cache(e).newBlock */
      return 0;
  }
  else {
     /* pick a cache entry e, where Cache(e).tId = 0.
        If there is no such entry, then return -1 */
     diskRead(blockAddr, &Cache(e).oldBlock);
     Cache(e).newBlock = Cache(e).oldBlock;
     block = &Cache(e).newBlock;
     Cache(e).blockAddr = blockAddr
     Cache(e).tId = tId;
     return 0;
  }

}
```

```
/* A transaction should call write(&Cache(e).newBlock,
tId) after it updates Cache(e).newBlock. */

int write(int blockAddr, int tId)
{
/* find the cache entry e for block blockAddr */
if (there is no such entry) return 0
else
  {
        Cache(e).tId = tId;
             return 1;
 };
}
```

```
int commit(int tId)
{
  for (each cache entry e where Cache(e).tId == tId)
  {
    status = diskWrite(Cache(e).blockAddr,
                       &Cache(e).newBlock);
    Cache(e).tId = - tId;
    if (status == -1) {
          Abort(tId);
         return -1;
    }
    Cache(e).oldBlock = Cache(e).newBlock
   };
   for (each cache entry e where Cache(e).tId == -tId)
      Cache(e).tId = 0;
   lastTrans = -lastTrans;

   return 0;
}
```

```
int abort(int tId)
{
  for (all cache entries e, where Cache(e).tId == -tId)
  {
     repeat
     {
       status = diskWrite(Cache(e).blockAddr,
                &Cache(e).oldBlock)
     }until (status == 0);
     /* Of course, this will not terminate if diskWrite
        keeps failing, but ignore that issue */

     Cache(e).newBlock = Cache(e).oldBlock;
  };
  for (all cache entries e) {
          Cache(e).tId = 0
  };
  lastTrans = -lastTrans;
  return 0;
}
```