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 diskBlockAddr, int tId, int *memBlock)
{
  /* find the cache element e containing the block whose
     disk address is diskBlockAddr */
  if (there is such a cache element e) {
      /* the disk block at diskBlockAddr is in cache */
      Cache(e).tId = tId;
      memBlock = &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 */
     status = diskRead(diskBlockAddr, &Cache(e).oldBlock);
     if(status != 0) {return -1};
     Cache(e).newBlock = Cache(e).oldBlock;
     memBlock = &Cache(e).newBlock;
     Cache(e).diskBlockAddr = diskBlockAddr;
     Cache(e).tId = tId;
     return 0
  }
}
```

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

int write(int diskBlockAddr, int tId)
{
   /* find cache entry e for block diskBlockAddr */
   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).diskBlockAddr,
                       &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).diskBlockAddr,
                          &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
}
```