Problem 1:
Calculating L2 and L3 misses for following configurations:

Trace File	**A. Inclusive**		**B. Non-Inclusive Non-Exclusive**		**C. Exclusive**	
	L2 Misses	**L3 Misses**	**L2 Misses**	**L3 Misses**	**L2 Misses**	**L3 Misses**
Bzip	5398166	1446388	5397576	1445846	5397576	889221
GCC	3036461	1373402	3029809	1366248	3029809	1242824
Gromacs	336851	170531	336724	170459	336724	159302
h264ref	969678	342146	965624	333583	965624	143681
hmmer	1743421	391226	1735322	376344	1735322	300046
Sphinix	8820349	8207362	8815130	8205144	8815130	7220776

- ➔ **L2 misses in Inclusive will be more than NINE and Exclusive**.
  Reason: In **inclusive** we invalidate the data from L2, if it is evicted from L3. It might be the case that evicted block is **hot block** and it results in the more misses. Whereas in **NINE** and **Exclusive** L3 miss eviction doesn't invalidate block in L2.

- ➔ **Exclusive policy has less L3 Misses than NINE and Inclusive** as it has more combined capacity of L2 and L3 cache due to exclusive entries in L2 and L3.

- ➔ **L2 Misses in NINE and Exclusive are same** because the effect of **replacement policy** on L2 cache is similar in both configuration. Because in both cases for every hit or miss in L2 ,we do same operations on L2 cache and after every access we have same blocks in both cache.

Source Code file: Prob1.c

Problem 2:
L3 Misses in Inclusive

Trace File	Cold Misses	Cold+Capacity	Capacity Misses	Conflict Misses
Bzip	119753			
GCC	773053			
Gromacs	107962	143254	35292	27277
h264ref	63703	111605	47902	230541
hmmer	75884	153447	77563	237779
Sphinix	122069			

NOTE: Not able to run program for some files, because its slow for large files.
Source Code: Prob2.c

Cold Misses are calculated in Prob1.c file.
Cold misses doesn't depend on **associativity, capacity of cache** and will be same for all caches unless **block size** is changed.

Prob2.c file gives the **Cold+Capacity** Misses occured in fully assossiative L3 with MIN algorithm and L2 with LRU.

Calculating Conflict Misses: Given a cache,We can't calculate **conflict misses** directly.For this ,We Run our address trace to **fully associative cache using MIN algo** and treating this as ideal configuration**(A)** where **conflict misses** are zero and uses this to calculate conflict misses for given **cache(B)** as follow:

Total misses in cache A=capacity+cold
Total misses in cache B=capacity+cold+conflict
Given that capacity and cold misses are same for both A and B.
Conflict misses in B=Total misses in cache **B-**Total misses in cache **A**