

CSE 140 HW #5

a) Examine the code given below to compute the average of an array:

```
total = 0;
for(j=0; j < k; j++) {
    sub_total = 0;
    /* Nested loops to avoid overflow */
    for(i=0; i < N; i++)
        sub_total += A[j*N + i];
    total += sub_total/N;
}
average = total/k;
```

When designing a cache to run this application, given a constant cache capacity and associativity, will you want a larger or smaller block size? Why?

b) examine the MODIFIED code given below:

```
total = 0;
for(i=0; i < N; i++) {
    sub_total = 0;
    /* Nested loops to avoid overflow */
    for(j=0; j < k; j++)
        sub_total += A[j*N + i];
    total += sub_total/k;
}
average = total/N;
```

Generally, how will the size of the array and the cache capacity impact the choice of block size for good performance? Why?

c) Translate the following line of code into MIPS. Assume i is \$s0, j is \$s1, base address of A is \$a0, N is \$a1 and sub_total is \$s2.

```
sub_total += A[j*N + i];
```

d) Now consider that we are executing one of these programs for the very first time. Assume we have a memory system with TLB, L1 I-cache, L1 Dcache, L2 cache and 2-level Page Table Virtual Memory system. List all the steps that will/may happen as we load instructions or data from memory. You will also need to list the steps taken when target instructions or data are not in cache or page tables. i.e. the steps to handle misses.

Load instructions:

1. Read instruction: check TLB if instruction is in memory by looking up its virtual page number.
2. If virtual page number is not present in TLB, read miss. Check L1 page table to see if virtual page number is present.

Load data:

1.

e) List and number all the ADDITIONAL steps that will happen when we are executing your translated code for $\text{sub_total} += A[j*N + i]$;

f) Suppose the whole program fit into the L1 caches and we are executing it many times. What are the steps from that will be skipped from (d) (just write down the #s).