2017.1 Human Media Multicore Computing
Midterm Exam (10am-11:30am, 4/21)

supervisor signature	

StudentID# : ( ) , Name : ( )

※ You may answer either in English or Korean language unless instructed to answer in English.

1. (26 points) Fill out the blanks (a)~(m) with **the most appropriate English words**. x_____ means the word starts with alphabet 'x'.

- Abbreviation 'SIMD' stands for
  (a. S_____) (b. I_____) (c. M_____) (d. D_____)

- (e.) is coordination of simultaneous events in order to obtain correct run-time order.

- Parallelism overheads mainly include
  - cost of (f.),
  - cost of (g.),
  - cost of (h.),
  - and extra (redundant) computation

- Amdahl's law implies that parallel programming is worthwhile when (k.).

- In a shared memory multiprocessor system with a separate (i.) memory for each processor, it is possible to have many copies of shared data: one copy in the main memory and one in the local (same as i.) of each processor that requested it. When one of the copies of data is changed, the other copies must reflect that change. (j.) is the discipline which ensures that the changes in the values of shared data are propagated throughout the system in a timely fashion.

- (k.) means the capability of a system to increase total throughput under an increased load when computational resources are (l.).

- Race condition is a type of flaw in an electronic or software system where (m.) the sequence or timing of other uncontrollable events.

2. (15 points) Determine whether each of following statements is True or False. If you are not sure, you may not give your answer, because if your answer is wrong, penalty (score subtraction: -2 points) is applied.

(a) Each thread of a program has its own heap area where memories are dynamically allocated. (True / False)

(b) One of the main advantages of cluster computing is to provide global memory access for achieving high parallel performance. (True / False)

(c) Dynamic load balancing is more suitable for heterogeneous multicore system than for homogeneous multicore system. (True / False)

(d) In UNIX, global variables are shared by processes in a multi-process program. (True / False)

(e) Fine-grained parallel system has high overhead associated with communication or synchronization compared to coarse-grained parallel system. (True / False)

3. (12 points) (a) What is producer-consumer problem in multi-threaded and concurrent programming? Explain with sufficient details.
( )

(b) What does the JAVA code "**`X.join();`**" do? Explain with sufficient details. ( )

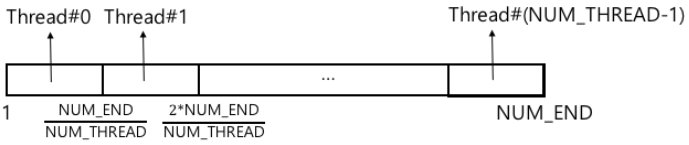
(c) What is Moore's law? Explain. ( )

4. (8 points) There are largely four steps for creating a parallel program: (i) decomposition, (ii) assignment, (iii) orchestration, and (iv) mapping.

(a) Describe what needs to be done in the decomposition step. Explain with sufficient details.
( )

(b) Describe what needs to be done in the assignment step. Explain with sufficient details.
( )

5.(39 points) Consider a multi-threaded JAVA program that computes the number of prime numbers between 1 and NUM_END using NUM_THREAD threads (Prob 4 of Lab 1). Assume that NUM_END can be divided by NUM_THREAD. Consider a static load balancing method where we divide entire number range (1 ~ NUM_END) into NUM_THREAD chunks : (1 ~ NUM_END/NUM_THREAD), (NUM_END/NUM_THREAD+1 ~ 2*NUM_END/NUM_THREAD), ..., ((NUM_THREAD-1)*NUM_END/NUM_THREAD+1 ~ NUM_END), and assign *i*-th chunk to *i*-th thread. Each thread calculates the number of prime numbers from its assigned chunk number range using **`isPrime()`** function. The main thread finally collects the results from each thread and sum them up to print the final result.



The main problem of above method is that it results in bad load balancing and poor parallel performance.

(a) Why does above method result in bad load balancing? Explain with sufficient details.

( )

(b) Explain your method that modifies above method to solve above problem. Your method should use static load balancing approach that results in good load balancing. Draw a picture that is similar to above picture for your explanation.

Explain your method:	Picture:

(c) Explain why your method results in good load balancing. ( )

(d) Fill out empty boxes with appropriate JAVA code that adopts a static load balancing approach with good load balancing.

----------< source code >----------

```
class PrimeThread extends Thread {
  int min_val, max_val, counter;

  PrimeThread(int x,int y) {
    min_val = x;
    max_val = y;
    counter=0;
  }

  public int getCounter() {
    return counter;
  }

  private boolean isPrime(int x){
    int i;
    if (x<=1) return false;
    for (i=2;i<x;i++) {
      if ((x%i == 0) && (i!=x)) return false;
    }
    return true;
  }

  
}
```

```
public class ex4 {
  private static final int NUM_THREAD=4;
  private static final int NUM_END=200000;

  public static void main(String[] args) {
    int i,sum=0;
    int Width;
    PrimeThread[] t = new PrimeThread[NUM_THREAD];

    
    System.out.println("number of prime numbers: "+sum);
  }
}
```