Check which problems you are submitting:

☐ #1
☐ #2
☐ #3

How many pages total?______
Please do not write on the back of any pages.

_________________________
(print name)

_________________________
(signature)

_________________________
(NetId)
1. (20pts Total) Reader / Writer

Consider the Readers/Writers problem where any number of readers can examine a file, but only one writer at a time can update the file. A writer is only allowed access when there are no active readers. Consider the following code as a potential solution. The common variables are the two semaphores wrt and mutex. Both are initially set to 1, and the integer variable readcount is initially set to 0.

```c
1: semaphore wrt = 1, mutex = 1;
2: readcount = 0;
3:
4: writer()
5: {
6:     wait(wrt);
7:     //writing is done
8:     signal(wrt);
9: }
10:
11: reader()
12: {
13:     wait(mutex);
14:     readcount++;
15:     if (readcount == 1) wait(wrt);
16:     signal(mutex);
17:     //Do the reading
18:     wait(mutex);
19:     readcount--;
20:     if (readcount == 0) signal(wrt);
21:     signal(mutex);
22: }
```

a) (4pts) Define the term "race condition" in the context of the reader/writer problem.

b) (2pts) Is this a correct solution to the reader writer problem? (yes or no)

For c) and d) below state if the change will:

i) Have no significant effect
ii) Is needed for a correct solution
iii) Makes for an incorrect solution

For full credit you must explain your answer.

c) (7pts) What is the effect of swapping lines 15 and 16?
d) (7pts) What is the effect of omitting lines 18 and 21?
2. (20pts Total) Memory Management

Given memory partitions of 500K, 300K, 600K (in this order), how would each of the algorithms below place the following processes: 212K, 417K, 112K, 300K, 150K (in this order). Please show your work. Memory can be partitioned

a) First-fit (4pts)

b) Best-fit (4pts)

c) Worst-fit (4pts)

d) (2pts) Which algorithm makes the most efficient use of memory in this case and why?

e) (2pts) What is internal fragmentation?

f) (2pts) What is external fragmentation?

g) (2pts) What is one disadvantage of both Best and Worst fit?

3. (20 pts Total) Critical Section

Consider the proposed solution of the critical section problem listed below. Common variables flag1, and flag 2 are initially false.

```
//Process 1
while (true)
{
    while (flag2); //empty body
    flag1 = true;
    Critical section;
    flag1 = false;
    Noncritical section;
}

//Process 2
while (true)
{
    flag2 = true;
    Critical section;
    while (flag1); //empty body
    flag2 = false
    Noncritical section;
}
```

a) (5pts) Define the terms mutual exclusion, deadlock, and bounded waiting.
b) (5pts) Does the code above guarantee mutual exclusion? If no, give an execution sequence where mutual exclusion is violated. If yes, give an explanation why all three requirements hold.

c) (5pts) Could deadlock occur? If no, explain why it cannot occur. If yes, give an execution sequence that leads to deadlock.

d) (5pts) Could bounded waiting occur? If no, explain why it cannot occur. If yes, give an execution sequence that allows bounded waiting.
Question 1) (20 points) For each function below with input argument n, determine the asymptotic number of “basic operations” that will be executed. Justify your answer for each case. **Note:** For the recursive functions, you should first write the corresponding recurrence relation. Then solve the recurrence relation to come up with the asymptotic bound.

θ(1) θ(log n) θ(n) θ(nlog n) θ(2^n) θ(nlog n^2) θ(n^2) θ(n^3) θ(n!) Other? Please specify.

a) void func(int n) {
    if(n>1)
    {
        func(n-1);
        Perform n basic operations;
    } //endif
}

b) void func(int n) {
    if (n > 3)
    {
        func(n/4);
        func(n/4);
        func(n/4);
        func(n/4);
        Perform n basic operations;
    } //endif
}

c) void func(int n) {
    if (n > 1)
    {
        func(n/2);
        func(n/2);
        Perform 1 basic operation;
    } //endif
}

d)
void func(int n) {
    int i=n;
    while (i>0)
    {
        Perform 1 basic operation;
        i=i/4;
    } //endwhile
}

**Question 2)**

a) **(8 points)** Explain how heap data structures are different from binary search trees (BSTs). Provide at least two main differences and explain each.

b) **(12 points)** Apply the Heap Sort algorithm to sort the following list in ascending *(non-decreasing)* order. In addition to drawing the tree step by step, you should draw the array after each step. Show all your work.

```
int list[]={5, 6, 9, 8, 2, 1}
```

**Question 3) (20 points)** Consider two singly sorted linked lists, L1 and L2, each of which is sorted in ascending (non-decreasing) order. Assume L1 has n entries and L2 has m entries, where n, m>=0. Each entry has two components: a key component of type int and the usual next link component.

a) **(15 points)** Write a C++ Merge function to merge two given lists L1 and L2 in-place. That means your code should result in a singly merged list sorted in ascending order without creating a new list. Your Merge function should return a pointer to the head node of the merged list.

Here is an example of how merging would work. Assume the first linked list L1 has 4->35->95 and the other linked list L2 has 1->7->20->35, your code will produce 1->4->7->20->35->35->95 without using extra space.

b) **(5 points)** Analyze the time complexity of your code in part (a) in the worst-case. Justify your analysis.
Choose any 2 of the 3 problems.

1). Consider \( \Sigma_1 = \{ a, b, c \} \):
   a. (5 pts) State the Pumping Lemma for regular languages.
   b. (5 pts) Is the following language regular or not?
      \[ L_1 = \{ a^m b^n c^p : m \geq n \geq p \geq 0 \} \]
   c. (10 pts) Prove your answer to question b. You may use Pumping Lemma if needed.

2). Consider the context-free language over \( \Sigma_2 = \{ x, y \} \):
   \[ L_2 = \{ x^n y^n : n \geq 0 \} \]
   a. (10 pts) Give a context-free grammar for this language \( L_2 \).
   b. (10 pts) Draw the state diagram of a pushdown automaton to recognize this language.
      You may use the following notation to label your machine’s transitions:

      ![State Diagram](image)

      (read input symbol x, stack top is y, push symbol z)

3). The SUBSET-SUM Problem takes as input a set \( S \) of integers and an integer \( T \), the question is whether there exists a non-empty subset \( R \) that sums to \( T \).
   a. (5 pts) Define polynomial-time reducibility \( A \leq_P B \).
   b. (5 pts) In general, how do you prove that a given problem \( X \) is NP-Complete? Please list the steps.
   c. (10 pts) Prove that \( 3\text{-CNF-SAT} \leq_P \text{SUBSET-SUM} \). (3-CNF-SAT problem: Given a formula in 3-CNF, is there an assignment of the variables such that the formula evaluates to true? For example, \((x \lor \neg y \lor \neg z) \land (\neg x \lor y \lor z) \land (\neg x \lor y \lor \neg z)\) is a 3-CNF formula.)