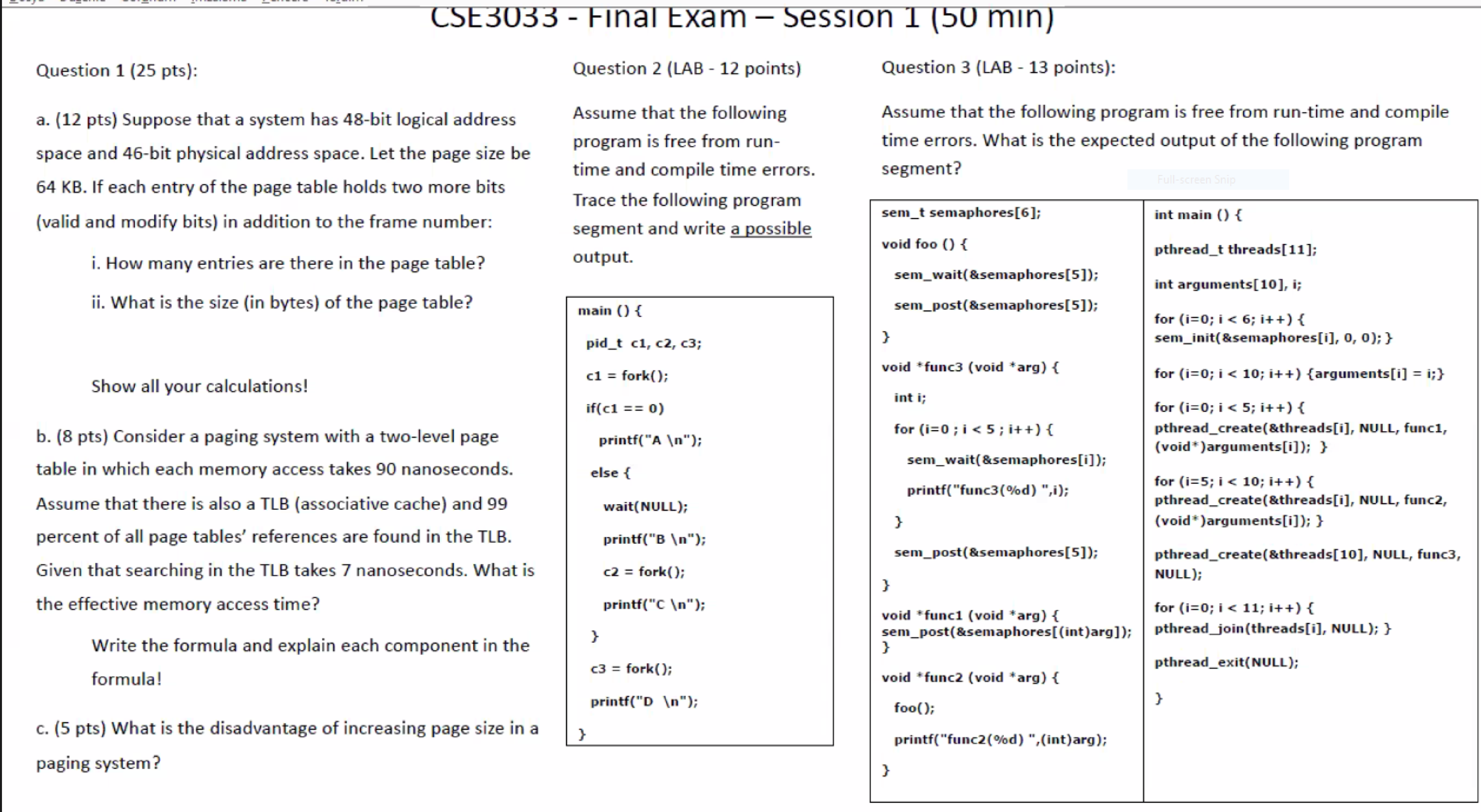

CSE3033 - Final Exam – Session 1 (50 min)

Question 1 (25 pts):

a. (12 pts) Suppose that a system has 48-bit logical address space and 46-bit physical address space. Let the page size be 64 KB. If each entry of the page table holds two more bits (valid and modify bits) in addition to the frame number:

i. How many entries are there in the page table?

ii. What is the size (in bytes) of the page table?

Show all your calculations!

b. (8 pts) Consider a paging system with a two-level page table in which each memory access takes 90 nanoseconds. Assume that there is also a TLB (associative cache) and 99 percent of all page tables' references are found in the TLB. Given that searching in the TLB takes 7 nanoseconds. What is the effective memory access time?

Write the formula and explain each component in the formula!

c. (5 pts) What is the disadvantage of increasing page size in a paging system?

Question 2 (LAB - 12 points)

Assume that the following program is free from run-time and compile time errors. Trace the following program segment and write a possible output.

```
main () {
 pid_t  c1, c2, c3;
 c1 = fork();
 if(c1 == 0)
   printf("A \n");
  else {
    wait(NULL);
    printf("B \n");
    c2 = fork();
    printf("C \n");
  }
  c3 = fork();
  printf("D  \n");
}
```

Question 3 (LAB - 13 points):

Assume that the following program is free from run-time and compile time errors. What is the expected output of the following program segment?

```
sem_t semaphores[6];

void foo () {
  sem_wait(&semaphores[5]);
  sem_post(&semaphores[5]);
}

void *func3 (void *arg) {
  int i;
  for (i=0 ; i < 5 ; i++) {
    sem_wait(&semaphores[i]);
    printf("func3(%d) ",i);
  }
  sem_post(&semaphores[5]);
}

void *func1 (void *arg) {
sem_post(&semaphores[(int)arg]);
}

void *func2 (void *arg) {
  foo();
  printf("func2(%d) ",(int)arg);
}
```

```
int main () {
pthread_t threads[11];
int arguments[10], i;
for (i=0; i < 6; i++) {
sem_init(&semaphores[i], 0, 0); }
for (i=0; i < 10; i++) {arguments[i] = i;}
for (i=0; i < 5; i++) {
pthread_create(&threads[i], NULL, func1,
(void*)arguments[i]);  }
for (i=5; i < 10; i++) {
pthread_create(&threads[i], NULL, func2,
(void*)arguments[i]); }
pthread_create(&threads[10], NULL, func3,
NULL);
for (i=0; i < 11; i++) {
pthread_join(threads[i], NULL); }
pthread_exit(NULL);
}
```

CSE3033 - Final Exam – Session 2 (50 min)

Question 4 (20 pts):

a. (10 pts) Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 4, 5,
3, 1, 5, 3, 4, 1, 6, 3, 6

How many page faults would occur for Least Recently Used replacement algorithm, assuming three frames? Mark each reference which causes a page fault. (Assume that all frames are initially empty).

b. (10 pts) What is the minimum number of page faults (Optimal) for the reference string given above by considering three frames? Show all your work.

Question 5 (15 pts): Consider the following snapshot of a computer system (PX: Process X, A-D: Resources):

Number of Resources in the System			
A	B	C	D
7	5	8	7

Maximum Demand				
	A	B	C	D
P0	2	3	1	4
P1	0	2	5	2
P2	4	1	0	5
P3	1	3	3	0
P4	3	0	3	3

Current Allocation				
	A	B	C	D
P0	1	0	1	1
P1	0	1	2	1
P2	3	0	0	3
P3	1	2	0	0
P4	1	0	3	0

a. (5 pts) Write down the “need” matrix and “available” vector.

b. (5 pts) Is the system in a safe state? Explain.

c. (5 pts) Suppose that P0 requests one instance of resource type A. Is it safe to grant this request according to Banker’s

Question 6 (15 pts):

Assume that we only have binary semaphores which have wait and signal operations. Implement a counting semaphore (S) using binary semaphores.

Hint: Consider the following variable and binary semaphores in your solution.

semaphore S1, S2; // binary semaphores

int C; // initial value of semaphore S.

a. Write the initialization code for S1 and S2.

b. Write the code for wait method: wait(S)

c. Write the code for signal method: signal(S)