National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Advanced Operating Systems**	**Course Code:**	**CS505**
Program:	**MS(Computer Science)**	**Semester:**	**Spring 2019**
Due Date:	**8-3-2019**	**Total Marks:**	**5**
Section:	**MS**	**Weight**	**5**
Exam:	**Quiz 2**	**Page(s):**	**1**
Name:	**Solution**	**Roll #:**	

Write our name and roll # on the quiz paper.

Q1: In the following code, three processes produce output using the routine putc and synchronize using two semaphores L and R. Functions P() and V() are equivalent to wait() and signal() functions that you have studied in the class. The function putc() prints its argument on the screen.

```
semaphore L = 3, R = 0; /* initialization */
/* Process 1 */        /* process 2 */        /* process 3 */
L1:                    L2:                    L3
    P(L);                  P(R);                  P(R);
    putc('C');             putc('A');
    V(R);                  putc('B');             putc('D');
                           V(R);
    goto L1;               goto L2;               goto L3;
```

A) How many D's are printed when this set of processes runs? __3___
B) What is the smallest number of A's that might be printed when this set of processes runs? __0___
C) Is CABABDDCABCABD a possible output sequence when this set of processes runs? ___no___
D) Is CABACDBCABDD a possible output sequence when this set of processes runs? _yes____

Q2: Write a multi-threaded version of the following code. You should write your code with 4 threads. The main thread is responsible for printing the largest element (maximum) and the corresponding index in the array. Assume that numbers in the array cannot be duplicated.

```
#include <stdio.h>
#include <stdlib.h>
#include<time.h>

#define SIZE 100
#define MIN 1
#define MAX 10000

int array [SIZE];
```

```
int main() {

   srand(time(0));

   for(int i=0 ; i<SIZE ; i++)
      array[i] = MIN + rand() % MAX;

   int max = array[0];
   int maxIndex = 0;

   for(int i=0 ; i<SIZE ; i++)
      if(array[i] > max) {
         max = array[i];
         maxIndex = i;
      }

   printf("Max Element: array[%d] = %d \n",
                                    maxIndex, max);
   return 0;
}
```