

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA**

**Laboratorio 5**

**2018-1**

**Indicaciones generales:**

- Duración: 2h 50 min.
  - Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
  - Si la implementación es significativamente diferente a la estrategia indicada o no la incluye, la pregunta será corregida sobre el 50 % del puntaje asignado y sin derecho a reclamo.
  - Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 60 % del puntaje asignado a dicha pregunta.
  - Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
  - El orden será parte de la evaluación.
  - Su trabajo deberá ser subido a Campus Virtual en el espacio indicado por los Jefes de Práctica.
-

## Pregunta 1 (8 puntos)

Se sabe que toda función recursiva puede replicarse utilizando instrucciones iterativas y una pila. Cuando utilizamos recursividad hacemos uso de una pila de manera implícita (pila del sistema). Para implementar una versión iterativa de la función, debemos utilizar una pila explícita. Se tiene disponible en Paideia una implementación recursiva del algoritmo Quick Sort. Deberá reemplazar la función recursiva por una función iterativa utilizando una pila.

A continuación se presenta un ejemplo muy básico en pseudocódigo:

Dada la función recursiva:

```
countdown(n) {  
    if (n > 0) {  
        imprimir(n);  
        countdown(n-1);  
    }  
}
```

Se puede utilizar una función iterativa:

```
countdown(n) {  
    stack.push(n)  
    while( !stack.empty() ) {  
        parametro = stack.pop()  
        imprimir(parametro)  
        if (parametro > 0) {  
            stack.push(parametro - 1)  
        }  
    }  
}
```

## Pregunta 2 (12 puntos)

La planificación de los procesos en un sistema operativo es un tema de mucha importancia y existen muchos algoritmos que plantean estrategias de acuerdo a objetivos específicos. “Uno de los algoritmos más antiguos, simples, equitativos y de mayor uso es el de turno circular (round-robin). A cada proceso se le asigna un intervalo de tiempo, conocido como cuántum, durante el cual se le permite ejecutarse. Si el proceso se sigue ejecutando al final del cuántum, la CPU lo expropia (desaloja) para dársela a otro proceso. Si el proceso se bloquea o termina antes de que haya transcurrido el cuántum, está claro que la conmutación de la CPU se realiza cuando el proceso se bloquea. Es fácil implementar el algoritmo de turno circular. Todo lo que el planificador necesita es mantener una lista de procesos ejecutables, como se muestra en la figura 1 (a). Cuando el proceso utiliza su cuántum, se coloca al final de la lista, como se muestra en la 1 (b).” (Andrew S.Tanenbaum - Sistemas Operativos Modernos)



Utilice una cola para implementar una planificación de procesos utilizando round robin. Recibirá como datos de entrada el número de procesos N y el quatum Q. Luego deberá leer N procesos conformados por dos enteros el id del proceso y una rafaga. Deberá imprimir el id de proceso y el tiempo de finalización en el orden en que terminan de ejecutarse.

### Ejemplo

Entrada	Salida
5 2	id del proceso: 5 Tiempo de terminación: 10
1 3	id del proceso: 1 Tiempo de terminación: 11
2 6	id del proceso: 3 Tiempo de terminación: 15
3 4	id del proceso: 2 Tiempo de terminación: 19
4 5	id del proceso: 4 Tiempo de terminación: 20
5 2	

Profesores del curso: Ivan Sipiran  
Andres Melgar

Pando, 07 de Junio del 2018