

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

**ALGORITMIA**

**Examen 1**

**(Segundo semestre 2015)**

**Indicaciones generales:**

- Duración: 2h 55 min.
- Materiales o equipos a utilizar: Apuntes de clase personales, escritos a mano o impresos.
- 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 o si la implementación es significativamente diferente a la estrategia indicada, el alumno no obtendrá el puntaje completo en dicha pregunta.
- 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 considerado dentro de la evaluación.
- Cada programa debe ser guardado en un archivo con el nombre *preg#codigo-de-alumno.c* y subido a PAIDEIA en el espacio indicado por los Jefes de Práctica.

Puntaje total: 20 puntos

---

**Cuestionario:**

**Pregunta 1 (4 puntos) Sorting**

Indique el nombre o número de equipo en el que realizó el primer trabajo grupal.

**Pregunta 2 (6 puntos) Tuercas y Tornillos**

Se tiene un conjunto de  $n$  tornillos de diferentes tamaños y sus correspondientes  $n$  tuercas. Se te permite probar una tuerca con un tornillo juntos para determinar si la tuerca es más grande que el tornillo, más pequeña que el tornillo, o encaja perfectamente. Sin embargo, no hay forma comparar dos tuercas o dos tornillos juntos. El problema consiste en emparejar cada tornillo con su respectiva tuerca. Diseñe un algoritmo para este problema con un caso promedio de eficiencia de  $O(n \log n)$ .

Expresé el algoritmo en pseudocódigo o lenguaje natural y mediante un ejemplo demuestre que cumple con la tarea planteada. Además, indique y justifique adecuadamente cuál es la eficiencia del algoritmo.

**Pregunta 3 (6 puntos) Las Torres**

Érase una vez, en un antiguo imperio, había dos torres de formas diferentes en dos diferentes ciudades. Las torres fueron construidas poniendo baldosas circulares una sobre otra. Cada una de las baldosas era del misma altura y tenía un radio. No es de extrañar que a pesar de que las dos torres eran de diferente forma, tenían muchos azulejos en común. Sin embargo, más de mil años después de su construcción, el emperador ordenó a sus arquitectos eliminar algunos de los azulejos de las dos torres de modo que tengan exactamente la misma forma y tamaño, y al mismo tiempo permanecer lo más alto posible. El orden de los azulejos en las nuevas torres debe seguir siendo el mismo que en las torres originales. El emperador

pensó que, de esta manera las dos torres podrían ser capaz de presentarse como el símbolo de la armonía y la igualdad entre las dos ciudades. Decidió que el nombre de ellas sería “las Torres Gemelas”. Ahora, cerca de dos mil años más tarde, le desafían con un problema aún más simple: dado las características de dos torres desiguales se le pide sólo averiguar el número de mosaicos en las grandes torres gemelas que se pueden construir a partir de ellas.

**Entrada:** La entrada consiste en diversos bloques de datos. Cada bloque de dato describe un par de torres. La primera línea del bloque de datos contiene dos enteros  $N_1$  y  $N_2$  ( $1 \leq N_1, N_2 \leq 100$ ) indicando el número de mosaicos respectivamente en las dos torres. La siguiente línea contiene  $N_1$  enteros positivos que indican el radio de los mosaicos (desde arriba hacia abajo) en la primera torre. Luego sigue otra línea que contiene  $N_2$  enteros positivos que indican el radio de los mosaicos (desde arriba hacia abajo) en la segunda torre. La entrada termina cuando se ingresa un cero a cada uno de los valores de  $N_1$  y  $N_2$ .

**Salida:** Por cada par de torres en la entrada, primero se muestra el número de la torre gemela seguida por el número de mosaicos (en una torre) en la torre gemela más alta que puede construirse a partir de ellos. Imprimir una línea en blanco después de la salida de cada conjunto de datos.

Ejemplo de Entrada:	Ejemplo de Salida:
7 6 20 15 10 15 25 20 15 15 25 10 20 15 20 8 9 10 20 20 10 20 10 20 10 20 10 20 10 10 20 10 10 20 0 0	Torre Gemela Nro. 1 Número de mosaicos: 4  Torre Gemela Nro. 2 Número de mosaicos: 6

Implemente un programa en C que resuelva el problema descrito.

## PARTE ELECTIVA

Para **UNA** de las preguntas que se presentan a continuación, elabore un programa en C que resuelva el problema descrito.

### Pregunta 4 (4 puntos) Criptoaritmética

Periódicos y revistas a menudo tienen ejercicios de criptoaritmética de la forma:

$$\begin{array}{r}
 \text{SEND} \\
 + \text{MORE} \\
 \hline
 \text{MONEY}
 \end{array}
 \xrightarrow{\text{SOLUCIÓN}}
 \begin{array}{r}
 9567 \\
 + 1085 \\
 \hline
 10652
 \end{array}$$

El objetivo es asignar a cada letra un dígito del 0 al 9 de tal forma que la aritmética funcione correctamente. Las reglas son que a todas las ocurrencias de un letra se les debe asignar el mismo dígito, y ningún dígito puede ser asignado a más de una letra.

### Pregunta 5 (4 puntos) El Taxi de Pancracio

Como cada jueves, Pancracio debe ir a la casa de su amigo Venancio para jugar Play Station. Para ello, Pancracio se desplaza desde su casa a la casa de su amigo Venancio en su taxi, un Austin Cambridge del 65. Él se ha comprometido en pasar por el restaurante del “Tío Bigote” para comprar hamburguesas para comer mientras juegan. Se trata de implementar la función **mikala** que le ayude a Pancracio a llegar a la casa de Venancio pasando por el restaurante del “Tío Bigote”. El mapa de calles debe representarse como una matriz de entrada a la función cuyas casillas contienen uno de los siguientes valores: 0 para “camino libre”, 1 para “camino bloqueado”, 2 para identificar la posición del restaurante del “Tío Bigote” y 3

para identificar la posición de la casa de Venancio. Hoy Pancraccio no está de muy buena suerte, su taxi se ha malogrado permitiéndole sólo manejar hacia el ESTE y hacia el SUR, suficiente para encontrarse con Venancio. La función **mikala** debe permitir saber si Pancraccio y Venancio podrán comer hamburguesas mientras juegan, de ser así deberá devolver la secuencia de casillas que componen el camino de regreso de la casa de Venancio al restaurante de “Tío Bigote”.

Profesores del curso: Fernando Alva  
Robert Ormeño

Pando, 17 de octubre de 2015