



UNI-IIFIIS

Algoritmia y estructura de datos: Primera Práctica Calificada

Pregunta 1 (de 8:00 a 8:38 am)

5 puntos

Diseñe un algoritmo que permita ingresar “n” ($n \leq 10$) números enteros positivos de entre 2 y 4 cifras (inclusive). Se garantiza que algunos de los números ingresados serán primos y otros no. Se pide:

- Presentar el menor de los números primos ingresados y la cantidad de veces que aparece.
- Presentar el número “P”, que se forma concatenando la suma de cifras de todos los números no primos ingresados, en el orden en que se introdujeron.

Por ejemplo, para $n=10$

Entrada: 200, 21, 17, 400, 23, 17, 608, 35, 202, 1011

Salida: Menor primo: 17, número de apariciones: 2,

$P = 23414843$

Note que: $P = 23414843$, es formado por las sumas: $2+0+0=2$, $2+1=3$, $4+0+0=4$, $6+0+8=14$, $3+5=8$, $2+0+2=4$, $1+0+1+1=3$

Nota: Resolver sin emplear arreglos, ni cadenas, ni funciones.

Envío de solución: hasta las 8:40 am

Nombre del archivo solución: apellido-nombre-pc1-p1

Enviar solución al buzón de tarea de UNI Virtual y al correo naudante@uni.edu.pe





UNI-IIFIS

Algoritmia y estructura de datos: **Primera Práctica Calificada**

Pregunta 2 (de 8:40 a 9:16 am)

5 puntos

Diseñe un programa que permita leer dos números enteros positivos cuyo número de cifras es mayor a 3. Luego el programa debe crear el mayor número que se puede formar con los dígitos en común. Ejemplo:

Entrada:

Primer número: 456702

Segundo número: 408937

Dígitos en común: 0, 4 y 7

Salida:

El mayor número formado por los dígitos en común es: 740

Nota: Resolver sin emplear arreglos, ni cadenas, ni funciones.

Envío de solución: hasta las 9:18 am

Nombre del archivo solución: apellido-nombre-pc1-p2

Enviar solución al buzón de tarea de UNI Virtual y al correo naudante@uni.edu.pe





UNI-IIFIS

Algoritmia y estructura de datos: **Primera Práctica Calificada**

Pregunta 3 (de 9:18 a 9:53 am)

5 puntos

Suponga que las posiciones de los casilleros de un tablero de ajedrez se identifican por un numero de 2 cifras, ejemplo 34 representa el casillero de la fila 3 y la columna 4. En base a lo anterior desarrolle un algoritmo en C++ que reciba las posiciones de dos caballos y determine si se atacan mutuamente. Los caballos se atacan si están en los casilleros extremos de una L formada por 4 casilleros.

Ejemplo 1: 42 y 54 se atacan

	c						
			c				

Ejemplo 2: 42 y 64 no se atacan

	c						
			c				

Nota: Resolver sin emplear arreglos, ni cadenas, ni funciones.

Envío de solución: hasta las 9:55 am

Nombre del archivo solución: **apellido-nombre-pc1-p3**

Enviar solución al buzón de tarea de UNI Virtual y al correo naudante@uni.edu.pe

