

Maestría en Ciencias de la Computación y Matemáticas Industriales
Proceso de admisión 2020
Centro de Investigación en Matemáticas (CIMAT), A.C.
Examen de programación
(Tiempo: 2 horas)

Nombre: _____

Fecha: _____

Instrucciones:

- Escribir lo más claro posible para que al digitalizar las respuestas sean legibles.
- No se debería de pasar más de 30mn en cada problema.
- Si se solicita código, escribirlo lo más claramente posible. Especificar el lenguaje/pseudo-código utilizado.
- Si requiere funciones estándar como determinar el máximo o mínimo de dos valores ($\max(a, b)$, $\min(a, b)$), ordenar (sort), determinar el tamaño de una cadena de caracteres (strlen), obtener el valor absoluto (abs), raíz cuadrada (sqrt) o logaritmos (log) puede utilizarlas sin implementarlas. Será necesario que implemente cualquier función diferente a las mencionadas anteriormente.

Problema 1 [3 point]

Dado un conjunto de N segmentos cerrados $\{[a_1, b_1], [a_2, b_2], \dots, [a_N, b_N]\}$, se dice que el conjunto está *cubierto* por un conjunto de M valores $\{v_1, v_2, \dots, v_M\}$, si se cumple que para cada segmento hay al menos un valor que queda dentro del mismo (incluyendo sus extremos, por ser segmentos cerrados).

a) Escriba una función `bool cover(int a[N], int b[N], int v[M])` que devuelva `true` si los M valores cubren los N segmentos, o `false` en caso contrario.

Ejemplo: ante la ejecución con los valores $a = \{3, 2, 8\}$, $b = \{4, 5, 9\}$, y $v = \{3\}$, el método debe devolver `false`, mientras que ante la ejecución con $a = \{3, 2, 8\}$, $b = \{4, 5, 9\}$, y $v = \{3, 8\}$ el método debe devolver `true`.

b) Escriba una función `int minRequired(int a[N], int b[N])` que devuelva el mínimo número de valores requeridos para cubrir los N segmentos.

Ejemplo: ante la ejecución con los valores $a = \{3, 2, 8\}$, $b = \{4, 5, 9\}$, el método debe devolver el número 2.

Problema 2 [2.5 point]

En una página web se nos ha abierto un libro de n páginas, y actualmente estamos situados en la

página p_1 . Existen dos botones de *scroll* que nos dejan respectivamente adelantar y retrasar s páginas. Por supuesto, no podemos situarnos antes de la página 1 ni tras la página n , por lo que si, estando entre las últimas $n - x$ páginas, tratamos de adelantar x páginas, nos llevará a la última. De la misma manera, si estamos en alguna de las primeras x páginas y tratamos de atrasar x páginas, nos llevará a la primera. El libro es muy grande, tiene más de 10^9 páginas.

Dados la página inicial (p_1), una página a la que queremos ir (p_2), el número de páginas del libro (n) y el valor s , desarrolle la función `int numPress(int p1, int p2, int n, int s)` que determine el número mínimo de veces que debemos presionar alguno de los botones de *scroll* para alcanzar la página p_2 . Si es imposible llegar a la página p_2 , devuelva el número -1. Su código debería ser capaz de calcular la respuesta en menos de 1 segundo.

Ejemplo: ante la ejecución con los valores $p_1 = 9$, $p_2 = 8$, $n = 10$ y $s = 2$, el método debe devolver el valor 2, ya que podemos alcanzar la pagina en dos pasos. En el primer paso (+2), adelantamos hasta la página 10 (el final del libro) y en el segundo paso (-2), atrasamos hasta la página 8.

Ejemplo: ante la ejecución con los valores $p_1 = 9$, $p_2 = 10$, $n = 11$ y $s = 2$, el método debe devolver el valor -1.

Aclaración: si no se le ocurre un algoritmo para cuando el número de páginas es mayor a 10^9 , pero sí para casos con 10^5 páginas, conteste con dicho algoritmo.

Problema 3 [2.5 point]

Escribir una función que reciba dos parámetros enteros (n, k) y devuelva como resultado el k -ésimo dígito del número entero positivo n . Los dígitos son enumerados de derecha a izquierda iniciando desde el valor cero.

Ejemplo: Si n es el entero positivo 53678, la función `dígito(n, 1)` debería regresar el dígito 7.

Restricción: Evitar el uso de conversiones del tipo de dato entero (`int`).

Problema 4 [2 point]

Al conjunto de números pertenecientes a la siguiente secuencia: $\{1, 3, 6, 10, 15, 21, 28, 36, 45, \dots\}$, se les conocen como números triangulares. Escribir una función que reciba un número entero positivo y determine si dicho número pertenece al conjunto de números triangulares.