

Ejercicio de Práctica en Papel

Nombre y Apellido: Comisión.....

El siguiente ejercicio debe ser resuelto con lapicera/lápiz. Todas las hojas deben tener nombre y apellido. Tener en cuenta la legibilidad de la solución entregada, ya que de no comprenderse lo escrito no se podrá corregir. Deben entregarse todas las hojas que contengan código asociado a la solución. Recuerde que si hace uso de funciones auxiliares, debe incluirlas y/o codificarlas según corresponda.

Tiempo de Resolución: 90 minutos.

Puntaje Requerido: 20/40 puntos.

Consigna: en la competencia de programación más linda de Argentina, compiten equipos en 4 niveles, tres integrantes por equipo y una computadora. Los datos de cada participante se encuentran almacenados en un struct **Participante**, que guarda la siguiente información:

- Nombre completo, un string
- Nro. CUIL (long long)
- Nro. Equipo (int)
- Nivel Competencia: 1,2,3, ó 4.

Y los datos de todos los participantes están almacenados en un vector **Participantes**, con tamaño físico=300.

Además, se dispone de un vector **Equipos**, el cual almacena la siguiente información para cada equipo:

- Nro. Equipo (un entero que coincide con su posición en el vector Equipos).
- Nombre de fantasía, un string
- Institución, un string
- Integrantes: la estructura de datos que Ud. crea conveniente para almacenar los datos completos de los tres integrantes del equipo.

Finalmente, se tiene una matriz **MatrizTecnica** de 4 filas (una para cada uno de los niveles de la competencia) y 35 columnas (tamaño físico). En cada celda se tienen los siguientes datos:

- Nro. Equipo, envíos realizados, problemas aceptados, minuto del último envío aceptado. (el valor minuto está en el rango 1..240).
- a) Defina los tipos de datos necesarios para almacenar la información de *Participantes*, *Equipos* y *MatrizTecnica*, y luego codifique la función `main()` y declare en la misma, el vector *Participantes*, el cual inicialmente está vacío y tiene tamaño 0. También declare el vector *Equipos* y la *MatrizTecnica*, utilizando los tamaños lógicos que sean necesarios.
- b) Diseñe y codifique una función `CargarEquipo()`, con los parámetros apropiados. La función deberá:
- Leer los datos de cada integrante y ejecutar una llamada a `CargarIntegrante()` (no es necesario codificar esta función, pero sí definir el prototipo de la misma y todos sus parámetros para poder ejecutar la llamada).
 - Leer los datos restantes del equipo y efectivamente agregar el nuevo equipo a *Equipos*.

Asumiendo que los datos de todas las estructuras se encuentran debidamente cargados,

- c) codifique la función `Premios()`, que reciba la *MatrizTecnica*, un número *Level* de nivel de competencia, y un entero *position* en 1..3, y retorne la siguiente información del equipo que obtuvo la posición *position* en la competencia de nivel *Level*:
- Nro. de Equipo

Algoritmos y Estructuras de Datos

Parcial 2 AEDD - Comisión Cuatrimestral 4/11/2024

- Nombre de Fantasia
- Porcentaje aceptados/enviados

- d) Codifique la función *UTNSantaFeesTop()* que determine si se cumple la siguiente condición: en los 3 niveles de competencia más altos (el nivel de fila 0 corresponde a secundaria) la UTN Santa Fe obtuvo alguno de los tres primeros puestos. Los equipos se ordenan según resuelvan mayor cantidad de problemas, y en caso de empate resulta ganador el equipo que hizo el último envío en menor cantidad de tiempo.

La *MatrizTecno* se ha ordenado de tal forma que todas las filas contienen a los tres primeros equipos de ese nivel, en sus tres primeras posiciones (ordenados del 1° al 3°).

- e) Codifique la función *RankingGlobal()* que obtenga los 3 primeros puestos entre todos los equipos en las tres primeras posiciones de cada nivel, según el valor de Enviados/Aceptados. La función debe retornar los 3 números que identifican a estos equipos.

Importante: Para la resolución del problema el alumno puede codificar todas las estructuras de datos y funciones que considere necesarias. Los campos de las estructuras deben respetar lo enunciado en la consigna. En los casos en los que no se indica un prototipo, los parámetros formales de las funciones (cantidad y tipo) deben definirse según los objetivos propuestos. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.