

OPERACIÓN GREYPENGUIN



Primera comunicación

-RIO: A ver chavales, aquí Río. El Profesor me ha pedido que os prepare una serie de pruebas para evaluar vuestra entrada al equipo de informáticos de la banda. Los indios no son malos, pero necesitamos a especialistas de Elite, así que cuento con vosotros.

-RIO: Las pruebas son del más alto nivel y las he dejado preparadas para que demostréis que sabéis salir de las situaciones más enrevesadas. No os confiéis y hacer lo que sabéis. Dar jarana de la buena.

-RIO: Así que chavales, ánimo y al toro.

Situación 1:

-RIO: A la hora de acceder a la caja fuerte de un ruso de estos ricachones, en donde están las claves de acceso a un bunker blindado en donde se guardan uno de los botines más jugosos del mundo (o eso aseguraba Berlín), nos encontramos que el patrón de ordenación de las claves de acceso sigue una secuencia de 10 números escrita en una tabla hash. Para extraer los datos, debemos ordenar los elementos de tipo double de esa tabla hash de mayor a menor.

-GreyPenguins: ¿Sisi entendido, pero como quieres que lo almacenemos o guardemos alguna petición?

-RIO: Tenéis que diseñar un objeto llamado deque, en el que podáis almacenar los elementos ubicados en las tablas hash. Tras esto, implementad un algoritmo de ordenación (podéis llamarlosort) que ordene la secuencia de dichos elementos, pudiendo así obtener las claves de desbloqueo de la caja fuerte del ruso.

-GreyPenguins: Al habla el escuadrón penguins hemos logrado la primera misión mediante el uso de una hash table, el lenguaje C++ no tiene su propia implementación de una Hash table por lo que nuestro equipo super especial avanzado de elite de GrayPenguins ha creado la clase Hash Table con los distintos atributos para representar la realidad, te hemos llenado el teclado de nieve, Río tu luego lo limpias no? Vale tío, otra cosa Río, creo que deberías asignarnos tareas más complicadas esto ha sido pan comido de verdad... ¿Así piensas reclutar a un equipo de elite?

-GreyPenguins: De todos modos, Río creo que se te ha pasado algo por alto, hay muchos seguratas, estos rusos no se andan con tonterías debe ser realmente importante lo que tengan ahí escondido.

Situación 2:

-RIO: Bien chavales, en esta situación tenéis que plantearos que hay que hacer también reconocimiento de la gente que hay dentro de la seguridad de la casa del ruso. Para poder categorizar a los equipos de seguridad, se necesita un sistema que permita recoger la información de los seguratas que tiene el ruso en su casa.

-RIO: El programa almacenará la información del nombre del segurata y un número correspondiente a la puntuación que sacaron en su evaluación como candidatos para securizar la casa del ruso.

-GreyPenguins: Entiendo por dónde van los tiros, pero de que tamaño quieres que hagamos el almacén no sabemos cuántos seguratas hay, espera.

-GreyPenguins: Uno, dos..., ¿tres? Ah no, ese es el de antes. Tres, cu-cuatro

-GreyPenguins: Bueno si, no sabemos cuántos hay son todos muy parecidos como para distinguirles, parecen kpopers

-RIO: Como de primeras no sabemos el número de seguratas que hay, el programa debe permitir introducir tantos seguratas como sea posible (usar para ello la clase vector).

-RIO: Una vez introducidos los datos de todos los seguratas, el sistema debe ordenarlos de mayor a menor puntuación para saber quiénes son los guardias con los que más cuidado debemos de tener, mostrando la información ordenada por pantalla.

-GreyPenguins: Y espera, como les damos una puntua...

(sonido de móvil colgado) Pi Pi Pi

-GreyPenguins: Agg da igual ya nos las apañaremos

-GreyPenguins: Hola buenas escuadrón penguins al teléfono, tarea más que resuelta lo único ha sido inventarse nombre para esos gorilas, no hay tantas especies de homínidos en el mundo como para cubrirles con mote. Para resolverla hemos usado la clase VectorPEL que nos enseñó el gran maestro pingüino Jorge García, que en pase descansa maestro, como lo extrañamos. El maestro nos enseñó a definir una clase Segurata, donde guardamos la información de cada ruso para poder almacenar sus instancias dentro del VectorPEL. Una vez tenemos sus nombres y su puntuación, hacemos un poco de magia con un algoritmo de ordenamiento, y devolvemos un Vector con todos los seguratas ordenados por su puntuación. Rio, y para que no te quejes después de que no te lo dejamos bonito ... si escribes un 2 en el menú principal, te desplegamos la información en un formato family friendly para que hasta el abuelo Cacho pueda leerlo, para que después no digas que los pingüinos no molan.

Situación 3:

-RIO: Bien, pasamos a la situación 3. Esta vez queremos poder manejar la información almacenada en el vector de seguratas de la situación anterior. Planteando que el TAD InfoSegurata tiene la siguiente estructura, hay que realizar lo siguiente:

```
struct InfoSegurata
{
    string nombre_segurata;
    int    evaluación_candidato;
};
```

1. Se siguen almacenando los datos dentro de un objeto vector<InfoSegurata> y debe estar ordenado por la evaluación del candidato.
2. Es necesario realizar un programa, que pueda calcular y obtener las evaluaciones máximas y mínimas de cada segurata. Así podremos saber cuál es el segurata más y menos peligroso.
3. También se debe obtener el promedio de las seguratas para saber la dificultad de este y futuros golpes.
4. Toda esta información debe mostrarse en pantalla.

-GreyPenguins: Oído cocina, también quieres que te digamos ¿cuántas comidas hacen al día o si duermen con antifaz?... Bueno no pasa nada nos ponemos a trabajar en mejorar el almacén de datos

-GreyPenguins: Club Pengu... ¡ESCUADRON PENGUIN! quería decir. Rio ya está el “sistema de búsqueda avanzado para reconocimiento y evaluación de rusos polares mazados” o así es como lo hemos llamado. No ha sido muy difícil la tarea la verdad lo hemos solventado usando un poco de código de la situación 3, hemos agregado algunos atributos a la clase Segurata para que sea más fácil reconocerla puntuación mínima y máxima de cada segurata. Luego, con uno de esos bucles to chulos que le gustan a los programados, hacemos un promedio de todos los puntajes de los seguratas, PARA ASI DAR EL TOP TIER DE LOS SEGURATAS que lo imprimimos por consola línea a línea, junto con sus nombres y resto de atributos. Vale Rio, pon algo más difícil que vamos sobrados.

Situación 4:

-RIO: Muy bien Mundos Gaturro, sois realmente buenos en lo vuestro la verdad... A ver si esta nueva tarea la resolvéis con tanta velocidad y eficacia, el golpe es mañana

-RIO: Bien chavales, esta va a ser la última situación. Siguiendo los desarrollos realizados hasta ahora. Deberéis escribir una función que separe en 2 los datos almacenados en el vector. En el primero estarán únicamente aquellos más peligrosos (una evaluación mayor a 8), y en el otro aquellos que, dentro de ser chungos, no van a dar muchos problemas.

-RIO: Pero, en este caso tenéis que hacerlo de dos formas diferentes:

1. La primera forma es usando 2 vectores adicionales, uno de seguratas peligrosos y otro de seguratas menos peligrosos, conservando el original durante la separación y eliminándolo al terminar de separarlos.
2. La otra forma, es generar una función de seguratas peligrosos, que compruebe del vector original cuales son peligrosos y sacándolos del vector original e introduciéndolos en el nuevo. Dejando en el vector original los menos peligrosos y en el nuevo los más peligrosos.
3. Para ambas formas, realizar las funciones sin usar métodos iterativos. Únicamente recursivos.

-GreyPenguins: Mmm entendemos, esto es para que Oslo no se distraiga con los mindundis de 8 puntos y solo vaya a por los de su calaña... Ya estamos trabajando en ello, mañana se efectuará el golpe sin el más mínimo problema

-GreyPenguins: Perfecto Rio, todo vuestro el software ya está en su totalidad, espero que vaya bien en el golpe, ya me contarás. Ah por cierto esta última tarea era molestar por molestar con esas especificaciones tan sibaritas... de todos modos nada que no pueda afrontar GreyPenguins, mediante el uso de las técnicas legendarias explicadas por el pingüino Jorge García en el bootcamp de programación sobre nieve, aprendimos sobre la sentencia move y las referencias & , utilizamos esto para crear un nuevo vector donde según la distancia mínima vamos agregando seguratas al vector teniendo en cuenta su puntuación. Vale y con eso terminamos todo Rio, no te olvides de darnos un extra del botín que Hacienda está muy arriba nuestro últimamente y nos quieren sacar to tío. May the force be with you Rio.

Fin