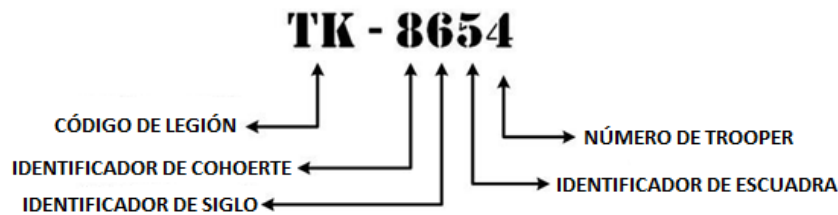


Guía de ejercicios prácticos

A continuación se plantean una serie de problemas, se deberá utilizar las funciones definidas en el TDA tabla *hash* para dar solución a estos.

1. Desarrollar un algoritmo que permita implementar una tabla *hash* para representar un diccionario que permita resolver las siguientes actividades:
 - a. agregar una palabra y su significado al diccionario;
 - b. determinar si una palabra existe y mostrar su significado;
 - c. borrar una palabra del diccionario;
 - d. la tabla debe tener 28 posiciones y manejar las colisiones con lista enlazadas;
 - e. mejorar el rendimiento de la tabla utilizando árboles binarios de búsqueda.
2. Desarrollar un algoritmo que implemente una tabla *hash* para una guía de teléfono, los datos que se conocen son número de teléfono, apellido, nombre y dirección de la persona. El campo clave debe ser el número de teléfono.
3. Implementar un tabla *hash* cerrada para guardar las cátedras de una carrera universitaria de acuerdo a su código, que permita resolver las siguientes actividades:
 - a. cargar cátedras de una carrera de las cuales se conoce nombre, modalidad (anual o cuatrimestral), cantidad de horas;
 - b. además se deben poder agregar los docentes vinculados con las cátedras;
 - c. debe ser una tabla cerrada;
 - d. debe poder solucionar las colisiones;
 - e. no podrán estar cargadas de manera correlativa de acuerdo a un número.
4. Desarrollar un algoritmo que implemente una tabla *hash* cerrada para cargar personajes de Star Wars de los que solo se conoce su nombre, contemplando las siguientes actividades:
 - a. la tabla inicialmente será de 20 posiciones;
 - b. deberá permitir el manejo de colisiones;
 - c. cuando el factor de carga de la tabla exceda el 75%, se deberá incrementar el tamaño de la tabla al doble y hacer un *rehashing* de las claves cargadas.

5. Desarrollar un algoritmo que implemente una tabla *hash* cerrada para administrar los contactos de personas de las cuales se conoce nombre, apellido y correo electrónico, contemplando las siguientes pautas:
 - a. El campo clave para generar las posiciones son el apellido y nombre.
 - b. Deberá contemplar una función de sondeo para resolver las colisiones.
6. Los *Stormtrooper* del imperio galáctico se identifican de la siguiente manera:



Darth Vader le encarga desarrollar los algoritmos para organizar los *Stormtrooper* cumpliendo con las siguientes demandas:

- a. Deberá generar 2000 *Stormtrooper* siguiendo el formato de la imagen anterior contemplando las siguientes legiones FL, TF, TK, CT, FN, FO y los dígitos generados de manera aleatoria;
 - b. deberá cargar los *Stormtrooper* generados en dos tablas *hash* encadenadas, en la primera se deberá agrupar de acuerdo a los tres últimos dígitos del código y en la segunda a partir de las iniciales de la legión;
 - c. ahora obtenga todos los *Stormtrooper* terminados en 781 para asignarlos a una misión de asalto y a los terminados en 537 para una misión de exploración;
 - d. ahora obtenga los *Stormtrooper* de la legión CT para que custodien a Darth Vader a una misión de exploración al planeta Hoth y los de la legión TF para una misión de exterminación a Endor.
7. Escribir un algoritmo que permita utilizar una tabla *hash* doble para guardar los datos de Pokémons, que contemple las siguientes actividades:
 - a. la primera tabla *hash* debe ser cerrada y la función *hash* debe ser sobre el tipo de Pokémon, con lo cual se obtendrá el acceso a la segunda tabla;
 - b. cada segunda tabla debe ser encadenada utilizando listas enlazadas y la función *hash* deberá utilizar el número del Pokémon como clave;
 - c. el tamaño de la primera tabla debe ser lo suficientemente grande como para que pueda almacenar todos los distintos tipos de Pokémon, debe manejar las colisiones con alguna función de sondeo;

- d. el tamaño de cada una de las segundas tablas debe ser 15;
 - e. el algoritmo debe permitir cargar tipos de Pokémon en la primera tabla y crear su respectiva segunda tabla, —en el caso de que no exista—;
 - f. si el Pokémon es de más de un tipo deberá cargarlo en cada uno de las tabla que indiquen estos tipos;
 - g. deberá permitir cargar Pokémons de los cuales se dispone de su número, nombre, tipo, nivel.
8. La alianza rebelde necesita comunicarse de manera segura pero el imperio galáctico interviene todas la comunicaciones, por lo que la princesa Leia nos encarga el desarrollo de un algoritmo de encriptación para las comunicaciones rebeldes, que contemple los siguientes requerimientos:
- a. cada carácter deberá ser encriptado a ocho caracteres;
 - b. se deberá generar dos tablas *hash* para encriptar y desencriptar, para los caracteres desde el “ ” hasta el “}” —es decir desde el 32 al 125 de la tabla ASCII.
9. Desarrollar un algoritmo que permita cifrar y descifrar un mensaje carácter a carácter, contemplando las siguientes pautas:
- a. Se debe utilizar una tabla *hash* para guardar los valores de codificación y decodificación respectivamente que se vayan utilizando.
 - b. Se deberá cifrar de la siguiente manera: primero, convertir al valor numérico correspondiente de la tabla ASCII cada carácter y luego, cada número de dicho valor se deberá remplazar por su valor correspondiente según los siguientes valores: 1 – “abd”, 2 – “def”, 3 – “ghi”, 4 – “ jkl”, 5 – “mnñ”, 6 – “opq”, 7 – “rst”, 8 – “uvw”, 9 – “xyz”, 0 – “#?&”, y se debe agregar al final el carácter %. Por ejemplo D = 68 debería quedar de la siguiente manera “opquvw%”.
10. Nick Fury director de la agencia S.H.I.E.L.D. intenta detener a la organización Hydra y a su líder Red Skull, los agentes de la agencia pueden interceptar los mensajes de Hydra pero están cifrados, por tanto no pueden hacer nada con estos; afortunadamente el Capitán América en una misión encubierta logró determinar las pautas del método de codificación. Ahora Fury nos solicita desarrollar el algoritmo que permita decodificar los mensajes, contemplando las siguientes pautas:
- a. Las codificación se realiza de la siguiente manera:

- i. primero se convierte el carácter a su valor en la tabla ASCII y se lo multiplica por 37 para transformarlo en un número de cuatro dígitos;
- ii. segundo se calcula un complemento en base al valor del carácter:

$$\text{complemento}(\text{character}) = \begin{cases} 79 + \text{character} - 32, & \text{character} \leq 78 \\ 32 + \text{character} - 79, & \text{character} > 78 \end{cases}$$
- iii. luego a cada digito obtenido en el punto uno se lo eleva al cuadrado y se le suma un complemento obtenido en el punto anterior y se transforma a carácter;
- iv. por último se juntan los cuatros caracteres y se le agrega al final el carácter correspondiente al complemento.

Por ejemplo el carácter *R* se codifica de la siguiente manera:

$R = 82, 82 * 37 = 3034$, complemento $= 32 + 82 - 79 = 35 = \text{"\#"}$

$3^2 + 35 = 44 = \text{","}$, $0^2 + 35 = 35 = \text{"\#"}$, $3^2 + 35 = 44 = \text{","}$, $4^2 + 35 = 51 = \text{"3"}$

El resultado final son estos cinco caracteres "\#,3\#" ;

- b. deberá utilizar una tabla hash cerrada para almacenar cada una de las cadenas de caracteres –de cinco caracteres– asociados a cada clave, una buena alternativa para la función *hash* podría ser la función de Bernstein;
- c. no se debe decodificar todas las cadenas de caracteres, esto debe hacerse a medida que se necesitan y no están en la tabla;
- d. ayuda al Capitán América descifrando los siguientes tres mensajes para poder conocer cuáles serán los próximos movimientos de Hydra (los mensajes están almacenados en archivos de texto, que deberá leerlos previamente desde cada archivo, en el siguiente link: https://github.com/belwalter/mensajes_codificados).