LABORATORIO FINAL

TEMA: Cadenas de caracteres, Arreglos de Datos (uni y bidimensionales), Subprogramas (Funciones y Subrutinas) y Recursividad.

Materia: Algoritmia y programación II
Profesora: Rocio Ramos Rodríguez
Monitores: Ángel Cotes y Jeffrey Saavedra
Curso: IST2089

Primera entrega: Diccionario
Fecha primera entrega: 18 de octubre de 2016 hasta la 12:30 p.m.
Segunda entrega: Diccionario - Método de encriptación
Fecha segunda entrega: 27 de octubre de 2016 hasta la 12:30 p.m.
Tercera entrega: Todo
Fecha tercera entrega: 4 de noviembre de 2016 hasta la 12:30 p.m.
Fecha sustentación: semana del 8 al 11 de noviembre de 2016 en SDU.

Enunciado:

Se le ha solicitado implementar un buscador-traductor-encriptador, combinando los procesos explicados a continuación, de tal forma que:

1. Cuando el usuario digite una cadena de caracteres (máximo cuatro palabra), busque una a una en él diccionario.

1. Si la cadena se encuentra, debe ser encriptada utilizando el método Vigenére, en caso contrario informarle al usuario que no es válida la cadena para ser utilizada como clave.

2. Cada pareja debe diseñar su propia tabla para el método de encriptación Vigenére y debe investigar otro método de encriptación que debe utilizar antes del método propuesto. Utilizar su ingenio y creatividad, ver: http://es.wikihow.com/codificar-y-decodificar-utilizando-la-cifra-de-Vigen%C3%A8re

3. Una vez encriptada la clave, esta debe ser traducida a lenguaje binario y ser mostrada al usuario.

4. Debe existir la opción de realizar el des-encriptado, desde la cadena en lenguaje binario.

MÉTODO DE ENCRIPTACIÓN Vigenére

El cifrado de Vigenére, va tomando diferentes valores en función de la clave elegida. El cifrado de Vigenère utiliza una clave externa para realizar las sustituciones, con lo que este mismo algoritmo puede dar diferentes criptogramas para el mismo texto claro en función de la clave a utilizar.

Ejemplo:

Texto claro: s e g u r i d a d
Clave de cifrado: a b c
Para llevar a cabo el cifrado se divide el texto claro en grupos de tantas letras como tenga la clave, y a continuación se hace corresponder con las letras de la clave de cifrado:
Texto claro: s e g u r i d a d
Clave: a b c a b c a b c
A cada letra del texto claro le corresponde la que está dada por la posición que ocupa en el alfabeto la letra clave que le corresponde. Así, cuando la clave sea la letra «a», se avanzará una posición, si la clave es «b» serán dos, y si fuera «e» serán 5.
En el ejemplo, en primer lugar se transforma la letra «s» del texto claro según su clave «a», es decir, una letra en el alfabeto, el resultado será «t». En el segundo caso, la letra «e» según la clave «b» dará una «g», porque se avanza dos posiciones.
Texto claro: s e g u r i d a d
Clave: a b e a b e a b e
Criptograma: t g l v t n e c i
Resultado final: t g l v t n e c i

Ahora a comprobar cómo, cambiando la clave de cifrado y con el mismo texto claro, se obtiene un criptograma totalmente diferente:
Clave: bebe
Texto claro: s e g u r i d a d – – –
Clave: b e b e b e b e b e b e
Criptograma: u j i z t n f f f – – –
Resultado final: u j i z t n f f f
Para poder realizar el descifrado la única condición es conocer la clave que se ha utilizado en el proceso, y hacer los pasos a la inversa. Partiendo del criptograma, tendremos que dividir en grupos según la clave y, en esta ocasión, restar posiciones en vez de sumar.
Este método es algo más seguro que los vistos con anterioridad, debido principalmente a que el criptograma varía según una clave externa, no conocida en principio por un hipotético atacante. Sin embargo se ha demostrado que no resulta difícil romper este cifrado utilizando técnicas de criptoanálisis basadas en la incidencia de coincidencias en el criptograma.

DICCIONARIO

Debe poder traducir de inglés/español y de español/inglés, teniendo en cuenta lo siguiente:

1. Crear **dos arrays unidimensionales** de String para a almacenar máximo 100 palabras.
2. Crear un método (subrutina) para añadir una entrada al diccionario
3. Crear String traduce_to_Ingles(String e) y String traduce_to_Español(String in) devuelve la palabra traducida.
4. Utilizar una constante para indicar el límite por defecto de la capacidad del diccionario.
5. Si el usuario intenta añadir una palabra que ya está en el diccionario o si ya está lleno el array, se debe de informar al usuario de que la palabra no ha podido ser añadida.
6. En el inicio del programa, el usuario debe poder elegir entre usar el límite por defecto de capacidad del diccionario o bien establecer mediante teclado dicho límite.
7. El usuario debe poder introducir varias palabras separadas por comas y el programa debe traducir todas ellas.
8. **Opcional**: En caso de que el usuario indique que quiere traducir una palabra XXX y dicha palabra no tenga traducción, el programa deberá mostrarle un mensaje del tipo "*Usted quiso decir XXY*" donde XXY es una palabra de la misma longitud que XXX y con una única letras diferente, en caso de exista una que reúna tales requisitos.

CÓDIGO BINARIO:

El código binario es el sistema de representación de textos, o procesadores de instrucciones de ordenador utilizando el sistema binario (sistema numérico de dos dígitos, o *bit*: el "0" y el "1"). En informática y telecomunicaciones, el código binario se utiliza con variados métodos de codificación de datos, tales como cadenas de caracteres, o cadenas de bits. Estos métodos pueden ser de ancho fijo o ancho variable.

En un código binario de ancho fijo, cada letra, dígito, u otros símbolos, están representados por una cadena de bits de la misma longitud, como un número binario que, por lo general, aparece en las tablas en notación octal, decimal o hexadecimal.

Código ASCII:

El código ASCII (acrónimo inglés de American Standard Code for Information Interchange) es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.

Para la traducción a código Binario se debe consultar la tabla de conversiones de los caracteres y símbolos.

A tener en cuenta:

1. Utilizar las herramientas del ambiente gráfico de Java para solucionar este laboratorio.

2. Utilizar arreglos de datos, tanto unidimensionales como bidimensionales.

3. Implementar funciones y subrutinas tanto recursivas como no recursivas.

4. Para este laboratorio es necesario usar las operaciones y funciones de cadenas de caracteres.

5. La interfaz debe ser muy amigable y visualmente agradable al usuario.

6. La sustentación o modificación equivale al 60% del laboratorio y el programa enviado al link correspondiente del catálogo al 40%

7. El programa debe quedar guardado con el nombre lab_final_nombre1_nombre2 en la respectiva carpeta del servidor y enviar el comprimido al link correspondiente en el catálogo.

8. El intento de copia será calificado con 0,0 y anotación a la hoja de vida.