**Laboratorio N°1**

Alumno: Chañi Laura, Jose Carlos

**Laboratorio N°1 - Ing. Oscar Ramirez**

1. Ejercicio 1: Cifrado César
   1. Descifrado César

def cifradoCesarMayus(cadena):

resultado = ''

i = 0

while i < len(cadena):

ordenClaro = ord(cadena[i])

ordenCifrado = 0

if(ordenClaro >= 65 and ordenClaro <= 90):

ordenCifrado = (((ordenClaro-65)+3) % 26)+65

resultado = resultado+chr(ordenCifrado)

i = i+1

return resultado

def descifradoCesarMayus(cadena):

resultado=''

i=0

while i<len(cadena):

ordenNormal = ord(cadena[i])

ordenAlterado = 0

if(ordenNormal>=65 and ordenNormal<=90):

ordenAlterado=(((ordenNormal-65)-3)%26)+65

resultado = resultado+chr(ordenAlterado)

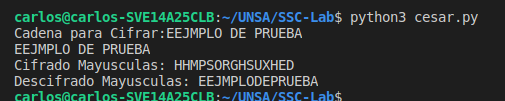
i=i+1

return resultado

cifrado= cifradoCesarMayus(cadena)

print('Cifrado Mayusculas: '+cifrado)

print('Descifrado Mayusculas: '+descifradoCesarMayus(cifrado))



* 1. Cesar - Mayusculas y Minusculas

def cifradoCesarMayusAndMinus(cadena):

resultado = ''

i = 0

while i < len(cadena):

ordenClaro = ord(cadena[i])

ordenCifrado = 0

if(ordenClaro >= 65 and ordenClaro <= 90):

ordenCifrado = (((ordenClaro-65)+3) % 26)+65

if(ordenClaro>=97 and ordenClaro<=122):

ordenCifrado = (((ordenClaro-97)+3) % 26)+97

resultado = resultado+chr(ordenCifrado)

i = i+1

return resultado

def descifradoCesarMayusAndMinus(cadena):

resultado=''

i=0

while i<len(cadena):

ordenNormal = ord(cadena[i])

ordenAlterado = 0

if(ordenNormal >= 65 and ordenNormal <= 90):

ordenAlterado = (((ordenNormal-65)-3) % 26)+65

if(ordenNormal>=97 and ordenNormal<=122):

ordenAlterado = (((ordenNormal-97)-3) % 26)+97

resultado = resultado+chr(ordenAlterado)

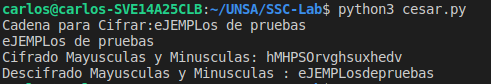
i=i+1

return resultado

cifrado= cifradoCesarMayusAndMinus(cadena)

print('Cifrado Mayusculas y Minusculas: '+cifrado)

print('Descifrado Mayusculas y Minusculas : '+descifradoCesarMayusAndMinus(cifrado))



* 1. Cesar Generalizado

def cifradoCesar(cadena, M):

resultado = ''

i = 0

while i < len(cadena):

ordenClaro = ord(cadena[i])

ordenCifrado = 0

if(ordenClaro >= 65 and ordenClaro <= 90):

ordenCifrado = (((ordenClaro-65)+M) % 26)+65

if(ordenClaro>=97 and ordenClaro<=122):

ordenCifrado = (((ordenClaro-97)+M) % 26)+97

resultado = resultado+chr(ordenCifrado)

i = i+1

return resultado

def descifradoCesar(cadena,M):

resultado=''

i=0

while i<len(cadena):

ordenNormal = ord(cadena[i])

ordenAlterado = 0

if(ordenNormal >= 65 and ordenNormal <= 90):

ordenAlterado = (((ordenNormal-65)-M) % 26)+65

if(ordenNormal>=97 and ordenNormal<=122):

ordenAlterado = (((ordenNormal-97)-M) % 26)+97

resultado = resultado+chr(ordenAlterado)

i=i+1

return resultado

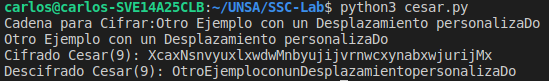
cadena = input("Cadena para Cifrar:")

print(cadena)

cifrado = cifradoCesar(cadena,9)

print('Cifrado Cesar(9): '+cifrado)

print('Descifrado Cesar(9): '+ descifradoCesar(cifrado,9))



1. Ejercicio 2: Mono-Alfabético

def cifradoMonoAlbabetico(cadena, clave):

resultado=''

i=0

while i<len(cadena):

inicial=ord(cadena[i])

final=0

mov = ord(clave[i % len(clave)])

if(mov >= 65 and mov <= 90):

mov = (mov-64)%26

if(mov >= 97 and mov <= 122):

mov = (mov-96)%26

if(inicial >= 65 and inicial <= 90):

final = (((inicial-65)+mov) %26)+65

if(inicial >= 97 and inicial <= 122):

final = (((inicial-97)+mov) %26)+97

resultado = resultado+chr(final)

i=i+1

return resultado

def descifradoMonoAlfabetico(cadena, clave):

resultado=''

i=0

while i<len(cadena):

inicial=ord(cadena[i])

final=0

mov = ord(clave[i % len(clave)])

if(mov >= 65 and mov <= 90):

mov = (mov-64)%26

if(mov >= 97 and mov <= 122):

mov = (mov-96)%26

if(inicial >= 65 and inicial <= 90):

final = (((inicial-65)-mov) %26)+65

if(inicial >= 97 and inicial <= 122):

final = (((inicial-97)-mov) %26)+97

resultado = resultado+chr(final)

i=i+1

return resultado

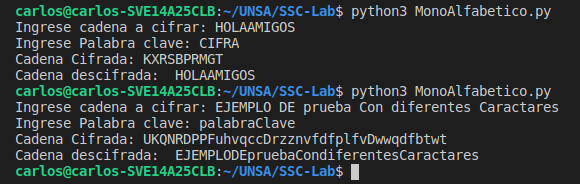
cadena = input('Ingrese cadena a cifrar: ')

clave = input('Ingrese Palabra clave: ')

cifrado = cifradoMonoAlbabetico(cadena, clave)

print('Cadena Cifrada: '+cifrado)

print('Cadena descifrada: ',descifradoMonoAlfabetico(cifrado,clave))



**Laboratorio N°1 - Ing. Lucy Delgado**

package Laboratorio01;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.PrintWriter;

public class Encriptacion {

public static class Letras{

public int repeticiones;

public char caracter;

public Letras(char a){

caracter=a;

repeticiones=0;

}

}

public static void main(String[] args) {

String cadena = "Puedo escribir los versos más tristes esta noche.\n" +

"Escribir, por ejemplo: “La noche está estrellada,\n" +

"y tiritan, azules, los astros, a lo lejos.”\n" +

"El viento de la noche gira en el cielo y canta.\n" +

"Puedo escribir los versos más tristes esta noche.\n" +

"Yo la quise, y a veces ella también me quiso.\n" +

"En las noches como ésta la tuve entre mis brazos.\n" +

"La besé tantas veces bajo el cielo infinito.\n" +

"Ella me quiso, a veces yo también la quería.\n" +

"Cómo no haber amado sus grandes ojos fijos.\n" +

"Puedo escribir los versos más tristes esta noche.\n" +

"Pensar que no la tengo. Sentir que la he perdido.\n" +

"Oir la noche inmensa, más inmensa sin ella.\n" +

"Y el verso cae al alma como al pasto el rocío.\n" +

"Qué importa que mi amor no pudiera guardarla.\n" +

"La noche está estrellada y ella no está conmigo.";

char[] caracteres = cadena.toCharArray();

String cadenaSusti ="";

for(int i=0;i<caracteres.length;i++) {

int iniOrder = (int) caracteres[i];

int aux =iniOrder;

if(iniOrder>=97 && iniOrder<=122) {

iniOrder = (iniOrder-97)+65;

}

iniOrder = sustituir(iniOrder);

if(iniOrder>=65 && iniOrder<=90) {

cadenaSusti=cadenaSusti+ (char)iniOrder;

}

}

//System.out.println(cadenaSusti);

escribirArchivo(cadenaSusti,"/home/carlos/poema20\_pre.txt");

String lectura = leerArchivo("/home/carlos/poema20\_pre.txt");

frecuencias(lectura);

UNICODE8(lectura);

addWord(lectura);

Kasiski(lectura);

}

public static int sustituir(int tilde) {

//a

if(tilde ==225 || tilde==193)

return 65;

//e

else if(tilde == 233 || tilde == 201)

return 69;

//i

else if (tilde == 237 || tilde == 205 || tilde == 72 || tilde == 74)

return 73;

//o

else if(tilde == 243 || tilde==211 )

return 79;

//u

else if(tilde ==250 || tilde==218 || tilde ==85 || tilde==87)

return 86;

//ñ

else if(tilde == 209 || tilde == 241 )

return 78;

//k

else if(tilde == 75)

return 76;

//y

else if(tilde ==89)

return 90;

else

return tilde;

}

public static void escribirArchivo(String cadena, String path) {

FileWriter fichero = null;

PrintWriter pw = null;

try

{

fichero = new FileWriter(path);

pw = new PrintWriter(fichero);

pw.println(cadena);

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

// Nuevamente aprovechamos el finally para

// asegurarnos que se cierra el fichero.

if (null != fichero)

fichero.close();

} catch (Exception e2) {

e2.printStackTrace();

}

}

}

public static String leerArchivo(String path) {

File archivo = null;

FileReader fr = null;

BufferedReader br = null;

try {

// Apertura del fichero y creacion de BufferedReader para poder

// hacer una lectura comoda (disponer del metodo readLine()).

archivo = new File (path);

fr = new FileReader (archivo);

br = new BufferedReader(fr);

// Lectura del fichero

String linea;

String lineaFinal="";

while((linea=br.readLine())!=null)

lineaFinal=lineaFinal+linea;

return lineaFinal;

}

catch(Exception e){

e.printStackTrace();

return null;

}finally{

// En el finally cerramos el fichero, para asegurarnos

// que se cierra tanto si todo va bien como si salta

// una excepcion.

try{

if( null != fr ){

fr.close();

}

}catch (Exception e2){

e2.printStackTrace();

}

}

}

public static Letras[] ordenar(Letras[] arreglo) {

for(int i = 0; i < arreglo.length - 1; i++)

{

for(int j = 0; j < arreglo.length - 1; j++)

{

if (arreglo[j].repeticiones < arreglo[j + 1].repeticiones)

{

Letras tmp = arreglo[j+1];

arreglo[j+1] = arreglo[j];

arreglo[j] = tmp;

}

}

}

return arreglo;

}

public static void frecuencias(String cadena) {

Letras[] abc = new Letras[26];

for(int i=0;i<abc.length;i++) {

abc[i]= new Letras((char)(i+65));

}

char[] caracteres = cadena.toCharArray();

for(int i=0;i<caracteres.length;i++) {

char unico = caracteres[i];

int pos = ((int)unico)-65;

abc[pos].repeticiones=abc[pos].repeticiones+1;

}

abc=ordenar(abc);

for(int i=0;i<abc.length;i++) {

System.out.println(abc[i].caracter+ " --> "+ abc[i].repeticiones);

}

}

public static void UNICODE8(String cadena) {

String resultado = "";

char[] caracteres = cadena.toCharArray();

for(int i =0;i<cadena.length();i++) {

resultado=resultado+ (int)caracteres[i];

}

System.out.println(resultado);

}

public static void addWord(String cadena) {

String resultado ="";

char[] caracteres = cadena.toCharArray();

int a=0;

for(int i=0;i<cadena.length();i++) {

resultado = resultado+ caracteres[i];

a=a+1;

if(a==20) {

resultado=resultado+ "AQUI";

a=0;

}

}

int i =0;

while(i<4) {

if(resultado.length()%4!=0) {

resultado=resultado+"X";

}else {

break;

}

i++;

}

System.out.println(resultado);

System.out.println(resultado.length());

}

public static void Kasiski(String cadena) {

char[] caracteres = cadena.toCharArray();

String salida="inicio";

int tri=1;

char trigLetra='-';

int distancia=0;

for(int i=1; i<caracteres.length;i++) {

if(caracteres[i-1]==caracteres[i]) {

tri=tri+1;

}

else {

tri=1;

}

distancia=distancia+1;

if(tri==3) {

trigLetra=caracteres[i];

salida=salida+" "+ distancia+" "+trigLetra;

distancia =0;

}

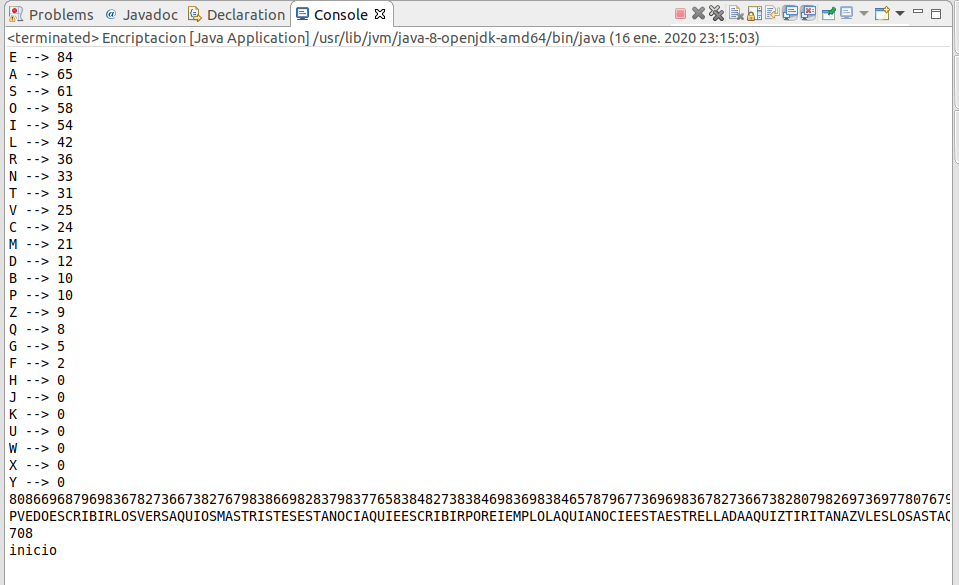
}

System.out.println(salida);

}

}

**Ejecución**



**Conclusiones:**

* El preprocesado es muy necesario antes de empezar a encriptar cualquier tipo de dato.
* Existen caracteres que si no se eliminan pueden dar pistas del la clave oculta.
* Se puede analizar los datos cifrados de acuerdo a la cantidad de caracteres que se repiten, por lo que tener las frecuencias ayudan al descifrado de claves.
* Si se desea cifrar caracteres, una opción para trabajar de manera numérica es transformar estos caracteres a algún código numérico (ASCII o UNICODE).
* Otra forma de analizar los datos cifrados es hallando posibles casos particulares como lo son los trigramas (Metodo Kasiski).
* Los cifrados que se basan en cambiar caracteres de acuerdo a una clave o función, son más sencillos de descubrir que algunos con funciones más complejas.
* Java permite trabajar con la codificación ASCII o UNICODE, siendo estas una ayuda a la hora de encriptar datos.
* El costo computacional de los métodos de cifrado vistos son bajos, sin embargo no ofrecen una alta seguridad y pueden ser descifrados.

**Cuestionario**

1. **Describa:**
   1. **Protección y seguridad de los datos:** Son los mecanismos y políticas de seguridad que un individuo u organización buscan para poder evitar que los datos no se ventilen al exterior o que estos datos sean accedidos por entidades ajenos a la organización o individuos que no cuenten con permisos necesarios para esta acción.
   2. **Criptografía:** Arte y técnica de escribir con procedimientos o claves secretas o de un modo enigmático, de tal forma que lo escrito solamente sea inteligible para quien sepa descifrarlo.
   3. **Seguridad y fortificación de redes:** Mecanismos que implementan servicios de seguridad y que estos a su vez implementan políticas de seguridad, todo con el fin de crear una defensa sólida contra alteraciones o ataques de terceros a un sistema específico.
   4. **Seguridad en aplicaciones informáticas, programas y bases de datos:** Servicios que evitan que información que el usuario, sin suficientes permisos, sean ventilados. También se incluyen en este grupo los procesos de software que brindan confiabilidad y seguridad a las aplicaciones de software.
   5. **Gestión de seguridad en equipos y sistemas informáticos:** Están incluidos todos los procedimientos y medidas de control para garantizar la seguridad y el correcto funcionamiento de estos equipos y sistemas.
   6. **Informática forense:** Es la aplicación de técnicas científicas y analíticas especializadas a infraestructuras tecnológicas que permiten identificar, preservar, analizar y presentar datos válidos dentro de un proceso legal.
   7. **Ciberdelito, ciberseguridad:** Es la actividad delictiva llevada a cabo mediante equipos informáticos o a través de Internet. El ciberdelito puede hacer uso de diferentes métodos y herramientas, como el phishing, los virus, spyware, ransomware o la ingeniería social, normalmente con el objetivo de robar información personal o de realizar actividades fraudulentas.
2. **Describa**
   1. **Gestión de la seguridad de la información:** Están involucrados aspectos de seguridad y políticas para preservar la información. Además como un concepto más claro son todas las políticas para administrar y gestionar la información de una organización.
   2. **Asesoría y auditoría de la seguridad:** Soluciones para cada necesidad mediante un procesos de consultoría donde se pueda analizar toda la información relativa a los riesgos, amenazas y vulnerabilidades que puedan existir, con el fin de realizar un diagnóstico, y proponer un tratamiento y solución adecuado, que proporcione un conjunto de medidas globales y satisfactorias para el bien que se pretende asegurar/proteger.
   3. **Análisis y gestión de riesgos:** Comprender el riesgo de la forma más detallada posible. Es el primer paso en el estudio de la evolución del riesgo. Lo usaremos para tomar decisiones en cuanto a si debemos tratar los riesgos y los métodos que utilizaremos.
   4. **Continuidad de negocio:** Aspecto muy importante en la seguridad de software, ya que este consiste en la fortaleza de un sistema para mantenerse funcionando a pesar de los posibles ataques o fallas internas que pueda tener.
   5. **Buen gobierno:** Referido al gobierno electrónico, es la correcta implementación de sistemas e integración de estos para lograr los objetivos del gobierno electrónico, entre los cuales encontramos: Acceso a servicios electrónicos, Claridad y Transparencia, etc.
   6. **Comercio electrónico:** Consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como redes sociales y otras páginas web.
3. **Describa alguna otra operación o función de preprocesamiento que se implemente sobre el texto claro en los criptosistemas.**

Existen también procesos para encriptar información que además garantizan poder conocer quien lee la información y quien lo envía. Un ejemplo de esto son los encriptados de clave pública y clave privada, que se puede aplicar a texto plano o también puede extenderse a encriptar archivos o paquetes de información.

**Bibliografía**

* Informatica Forense: <https://es.wikipedia.org/wiki/C%C3%B3mputo_forense>
* Ciberdelito: <https://www.avast.com/es-es/c-cybercrime>
* Comercio Electronico: <https://es.wikipedia.org/wiki/Comercio_electr%C3%B3nico>
* Analisis y Gestion de Riesgos: <https://www.escuelaeuropeaexcelencia.com/2016/07/gestion-de-riesgos-identificacion-analisis/>