

UNIVERSIDAD CENTRAL DEL ECUADOR



CRIPTOGRAFÍA Y SEGURIDAD DE LA INFORMACIÓN

NOMBRE: GRUPO 01

FECHA: 11-06-2024

**TEMA: Resultados de ejecución de
cada uno de los algoritmos**

PREGUNTA 1

```
PS C:\Users\Chriz> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:+ShowCodeDet
t.ls-java-project\bin' 'ejercicio1'
Ingrese la palabra: universidad
Número total de permutaciones: 39916800
Las primeras 10 permutaciones ordenadas alfabéticamente son:
addeiinrsuv
addeiinrsuv
addeiinrsuv
addeiinrsuv
addeiinrsvu
addeiinrsvu
addeiinrsvu
addeiinrsvu
addeiinrsvu
addeiinrusv
addeiinrusv
PS C:\Users\Chriz> █
```

1. Importa librerías necesarias: Usa ArrayList, Collections, List y Scanner de la biblioteca estándar de Java.
Estos imports traen clases de la biblioteca estándar de Java:
 - ArrayList, List: Para almacenar las permutaciones.
 - Collections: Para ordenar la lista de permutaciones.
 - Scanner: Para leer la entrada del usuario.
2. Solicita al usuario una palabra: Mediante Scanner, el programa pide al usuario que ingrese una palabra.
3. Genera permutaciones de la palabra: Usa un método llamado generarPermutaciones para obtener todas las permutaciones posibles de los caracteres de la palabra ingresada y las almacena en una lista.
Este método es recursivo y genera todas las permutaciones posibles de la palabra:
 - Si palabra está vacía ($n == 0$), se agrega el prefijo actual a permutaciones.
 - Si no, el método elige cada carácter de palabra y llama a sí mismo sin ese carácter, construyendo el prefijo paso a paso.
4. Imprime las permutaciones: Al finalizar, imprime cada una de las permutaciones generadas.
El programa imprime las primeras 10 permutaciones (o menos si hay menos de 10) de la lista.

PREGUNTA 2

```
Ingrese el valor de n: 4
Ingrese el mensaje: HOLAMUNDO
Matriz de cifrado:
H O L A
M U N D
O * * *
* * * *

Mensaje original: HOLAMUNDO
Mensaje cifrado: HOLAMUNDO*****
PS D:\UCE\8vo 9no\Cripto\ejercicio2\ejercicio2>
```

1. **Construcción de la matriz de cifrado:** En primer lugar, se construye una matriz de dimensiones $n \times n$, donde se colocan los caracteres del mensaje original. Si el mensaje es más corto que el tamaño de la matriz, se completan los espacios vacíos con asteriscos (*) o cualquier otro símbolo que se elija para este fin.
2. **Aplicación de la permutación en las filas:** Posteriormente, se realiza una permutación de las filas de la matriz. En este caso, el proceso consiste en mover todas las filas hacia abajo una posición. Esto provoca que la última fila pase a ocupar la primera posición, mientras que el resto de las filas se desplazan hacia abajo.
3. **Lectura del mensaje cifrado:** Finalmente, el mensaje cifrado se obtiene leyendo los caracteres de la matriz reorganizada fila por fila, de izquierda a derecha y de arriba hacia abajo, generando así el mensaje cifrado.

PREGUNTA 3

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\Roberto\Downloads\DeberCripto> & 'C:\Program Files\Ja
b1b571b28e671ebd5ca2\redhat.java\jdt_ws\DeberCripto_248ea85a\bin'
Ingrese el valor de n: 4
Ingrese el mensaje: roberto
Matriz de cifrado:
r o b e
r t o *
* * * *
* * * *

Mensaje original: roberto
Mensaje cifrado: rr**ot**bo**e***
PS C:\Users\Roberto\Downloads\DeberCripto>
```

El ejercicio imprime en consola lo siguiente: Realiza el cifrado de un mensaje utilizando una permutación de columnas con una matriz de tamaño $(n \times n)$ siguiendo estos pasos:

1. Entrada de datos:

- Se ingresa el valor de " n " que define tanto las filas como las columnas de la matriz.
- Se ingresa el mensaje que se desea cifrar, eliminando los espacios.

2. Validación:

- Se verifica que la longitud del mensaje sea menor o igual que $(n \times n)$, es decir, que el mensaje pueda ajustarse dentro de la matriz. Si no es así, se solicita un nuevo valor para " n ".

3. Construcción de la matriz de cifrado:

- Se crea una matriz de tamaño $(n \times n)$
- Se llena la matriz con los caracteres del mensaje, fila por fila. Si sobran espacios, estos se completan con el carácter ' * '.

4. Impresión de la matriz de cifrado:

- La matriz generada se imprime para mostrar cómo quedó distribuido el mensaje dentro de la matriz.

5. Cifrado del mensaje:

- Para cifrar el mensaje, se lee la matriz **columna por columna** y se concatena el resultado en una nueva cadena que representa el mensaje cifrado.

6. Salida:

- Se imprime el mensaje original y el mensaje cifrado.

Explicación del resultado en la consola:

- Se ingresó el valor de $n = 4$ y el mensaje "**roberto**".
- El mensaje "**roberto**" tiene 7 caracteres, que es menor que es $4 \times 4 = 16$ por lo que es válido.
- La matriz de cifrado se llena de la siguiente manera:

```
Matriz de cifrado:
r o b e
r t o *
* * * *
* * * *
```

- Ahora, se lee la matriz **columna por columna** para generar el mensaje cifrado:
- Columna 1: r, r, *, *

- Columna 2: o, t, *, *
- Columna 3: b, o, *, *
- Columna 4: e, *, *, *

El mensaje cifrado resultante es: **rr*ot*bo**e*****

PREGUNTA 4

```

Ingrese la cadena a cifrar: HOLA MUNDO
Ingrese el valor de n (desplazamiento): 5
Alfabeto original: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alfabeto cifrado : FGHIJKLMNOPQRSTUVWXYZABCDE
Cadena original  : HOLA MUNDO
Cadena cifrada   : MTQF RZSIT
PS C:\Users\jtony>

```

1. Entrada de datos

- Definimos primero una cadena que será el alfabeto original a usar.
- Ingresamos en valor de n y el mensaje a cifrar.

2. Ajuste de desplazamiento

- Aquí, el desplazamiento n se reduce a un valor dentro del rango del tamaño del alfabeto, usando el operador módulo (%). Esto asegura que cualquier valor de n más grande que el tamaño del alfabeto se "envuelva" dentro del rango válido (0 a 25).

3. Creación del alfabeto cifrado

- Generamos el alfabeto cifrado desplazando las letras del alfabeto original en n posiciones.

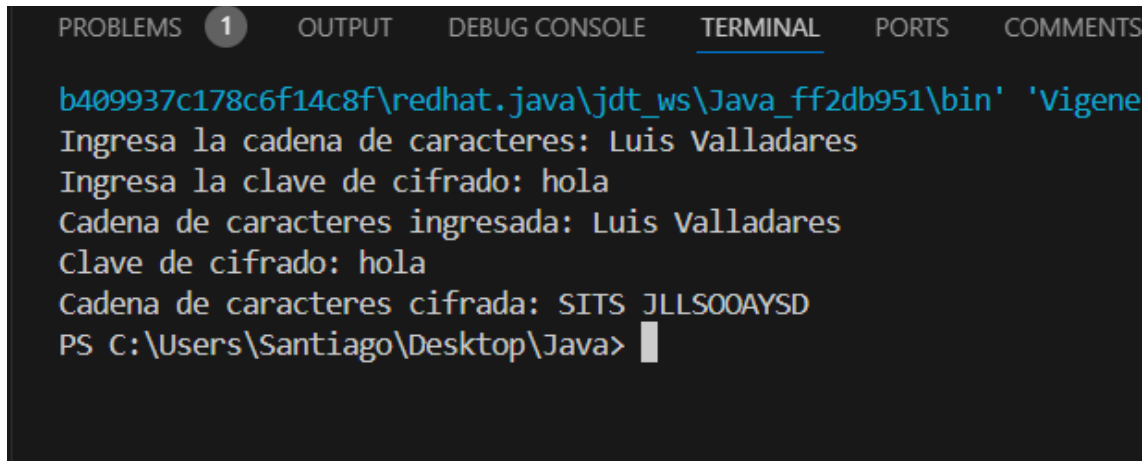
4. Construir la cadena cifrada

- Para cifrar el programa recorre cada carácter c en la cadena original:
- El index guarda la posición de c en el alfabeto original.
- Si c se encuentra en el alfabeto (index != -1), el programa agrega el carácter correspondiente del alfabeto Cifrado en la misma posición.
- Si c no está en el alfabeto (por ejemplo, un espacio o símbolo), se agrega directamente sin cambios.

5. Impresión de resultados

- Por últimos imprimimos los resultados de la cadena original y la cifrada junto con los alfabetos original y cifrado.

PREGUNTA 5

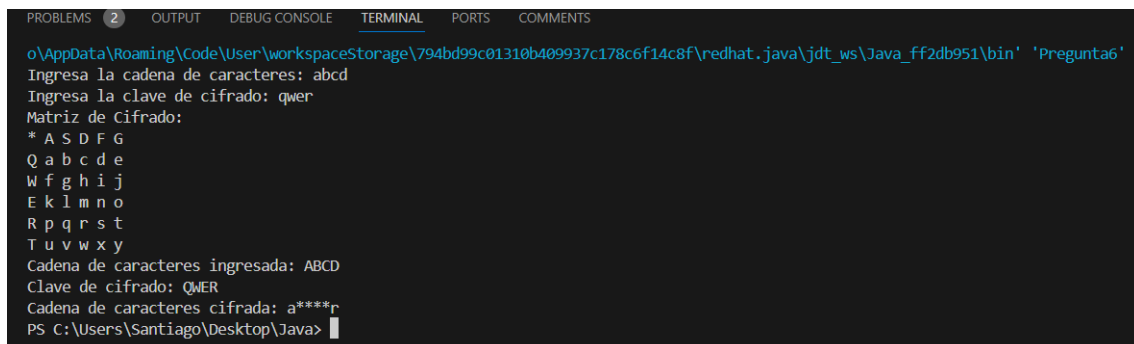


```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
b409937c178c6f14c8f\redhat.java\jdt_ws\Java_ff2db951\bin' 'Vigene
Ingresa la cadena de caracteres: Luis Valladares
Ingresa la clave de cifrado: hola
Cadena de caracteres ingresada: Luis Valladares
Clave de cifrado: hola
Cadena de caracteres cifrada: SITS JLLS00AYS
PS C:\Users\Santiago\Desktop\Java>
```

Se uso la ecuación del cifrado de Vigenere ya que en todas las fuentes hacían una recomendación de usar esta ecuación para realizar un software que realice el cifrado con esta técnica.

Entonces con la fórmula: $C_i = (P_i + K_i) \% 26$ se tomo uno a uno la cadena de caracteres que se debía cifrar para así obtener el resultado del texto cifrado

PREGUNTA 6



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
o\AppData\Roaming\Code\User\workspaceStorage\794bd99c01310b409937c178c6f14c8f\redhat.java\jdt_ws\Java_ff2db951\bin' 'Pregunta6'
Ingresa la cadena de caracteres: abcd
Ingresa la clave de cifrado: qwer
Matriz de Cifrado:
* A S D F G
Q a b c d e
W f g h i j
E k l m n o
R p q r s t
T u v w x y
Cadena de caracteres ingresada: ABCD
Clave de cifrado: QWER
Cadena de caracteres cifrada: a****r
PS C:\Users\Santiago\Desktop\Java>
```

Se puede observar que utilizamos el mismo método de cifrado de sustitución polialfabético que en la pregunta anterior, pero en esta pregunta el enfoque fue utilizar una matriz de cifrado ya definida. La matriz contiene letras específicas en los ejes de las filas y columnas, que limitan los caracteres que pueden ser cifrados.

La cadena de caracteres a ser cifrada está limitada a las letras que aparecen en la primera fila de la matriz (eje X), y la clave debe estar en el eje Y de la misma matriz. Para cada carácter del texto y de la clave, el programa busca la posición en la matriz correspondiente y utiliza el valor en esa celda para cifrar. Si algún carácter de la cadena o de la clave no se encuentra en los ejes de la matriz, el programa rellena el resultado con "***", indicando que el carácter no es válido para el cifrado.