Test de entrada

Ingeniería de Software

1. Descripción

Un banco prominente recientemente sufrió un ataque cibernético donde hackers se apropiaron de información crucial almacenada en sus bases de datos. Los datos robados fueron encriptados por los hackers, lo que impide el acceso del banco a su propia información. La información encriptada incluye detalles de transacciones, datos personales de los clientes, y registros de operaciones financieras. El banco ha logrado recuperar archivos encriptados y ahora enfrenta el desafío de desencriptarlos para recuperar la información vital.

El banco necesita un equipo especializado en ciberseguridad que pueda desencriptar los datos. Esta tarea implica analizar los métodos de encriptación utilizados por los hackers, lo que requiere un profundo conocimiento en criptografía y seguridad informática. Además, una vez desencriptados los datos, el banco deberá analizarlos para identificar cualquier manipulación o alteración de los mismos.

Otro aspecto crítico es la necesidad de reforzar sus sistemas de seguridad para prevenir futuros ataques. Este incidente también plantea preocupaciones sobre la protección de datos personales y la responsabilidad del banco en garantizar la seguridad de la información de sus clientes.

En esta primera fase de desarrollo, se realizará solo el proceso de evaluar el sistema de desencriptación que se cree que usaron los hackers, los cuales se detallan a continuación:

2. Consideraciones importantes:

- El archivo contiene datos al azar de una palabra, una frase corta y un párrafo.

3. Insumos

Se adjunta la siguiente fuente de datos en formato txt para ser procesado.

.

Archivo	Descripción
key.txt	``` -------------------BEGIN KEY--------------------- SQUARE1 | SQUARE2 | SQUARE3 1 2 3 | 1 2 3 | 1 2 3 1 E P S | 1 M + Z | 1 F G O 2 D U C | 2 L K X | 2 R I J 3 V W Y | 3 N B T | 3 H A Q -------------------END KEY---------------------- ```

Archivo	Descripción
datos_encriptados.txt	Contiene información a ser desencriptada

VNS

Cada integrante tiene un key.txt y un *datos_encriptados*.txt personal, el cual se encuentra en

https://drive.google.com/drive/folders/1tPCv0SpyYYLH3yRFa-SjL7PXJZhkDmEa?usp=drive_link

4. Desafíos

El software debe leer como entrada la carpeta (`datasets`) con los archivos txt que se adjuntan y como salida debe generar un archivo llamado **resultado.txt**, que contiene la información desencriptada.

El método para desencriptar la Información, se ejemplificará y es de la sgte manera:

Este sería un ejemplo de un archivo a desencriptar (En este caso sólo tiene "GBYNLVLEIFD" = "Hello World" encriptado)

datos_encriptados.txt

```
GBYNLVLEIFD
```

Paso 1:

Se separará los caracteres en grupos de a 7 y lo que sobre va en el último grupo.

Por ejemplo: `GBYNLVLEIFD, debe quedar así GBYNLVL EIFD`

Paso 2:

Luego cada carácter se buscará en la matriz (key.txt)

Por ejemplo: La letra `G, está en el Square3, fila 1 columna 2`

```
-------------------BEGIN KEY---------------------
SQUARE1 | SQUARE2 | SQUARE3
  1 2 3 |   1 2 3 |   1 2 3
1 E P S | 1 M + Z | 1 F G O
2 D U C | 2 L K X | 2 R I J
3 V W Y | 3 N B T | 3 H A Q
-------------------END KEY-----------------------
```

por lo tanto, quedará así

```
G
3 # Esta línea corresponde al "Square" que pertenece
1 # Esta línea corresponde a la fila que pertenece
2 # Esta línea corresponde a la columna que pertenece
```

Y así revisar cada una de las letras, de tal manera que este sería el resultado

```
GBYNLVL EIFD
3212212 1331 # Esta línea corresponde al "Square" que pertenece
1333232 1212 # Esta línea corresponde a la fila que pertenece
2231111 1211 # Esta línea corresponde a la columna que pertenece
```

VNS

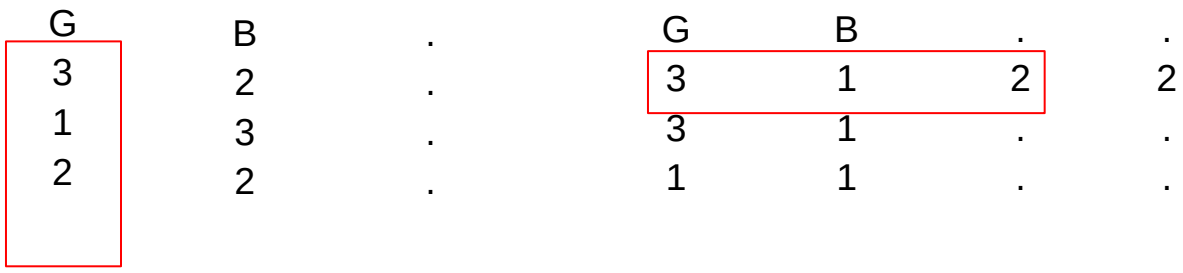
Paso 3:

Una vez estén agrupados, estos deberían volver a su posición real, esto se haría reordenando los arrays, intercambiando las posiciones de vertical a una sucesión horizontal.

Por ejemplo
la “G”, que tenía valores verticales de 312, ahora quedaría de manera horizontal. Luego le seguirían los valores de la letra “B”, todo agregado de manera horizontal. En caso de llegar al límite de columnas, esta continuaría en la siguiente línea.
Finalmente, así debería quedar ordenado.

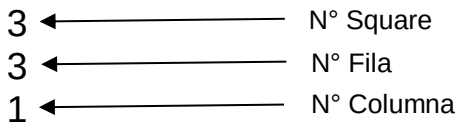
```
3122321332 1
3122113122 2
1111322311 1
```

Apoyo Visual la “G”



Paso 4:

Se debe procesar inversamente los valores a través de la “Key”, para obtener el mensaje guardado, de la siguiente manera:



La letra `que está en el Square3, fila 3 columna 1 es la H`

```
--------------------BEGIN KEY----------------------
SQUARE1 | SQUARE2 | SQUARE3
  1 2 3 |   1 2 3 |   1 2 3
1 E P S | 1 M + Z | 1 F G O
2 D U C | 2 L K X | 2 R I J
3 V W Y | 3 N B T | 3 H A Q
--------------------END KEY-----------------------
```

VNS

por lo tanto, quedará así:

```
HELLO+W ORLD
3122321 3321
3122113 1222
1111322 3111
```

Paso 5:

Finalmente se juntan todos los caracteres y se reemplazan los "+" por espacios, lo cual nos deja el mensaje original "Hello World"

NOTA IMPORTANTE: El ejemplo descrito anteriormente, se aplica a un archivo de pruebas dado en este desarrollo *datos_encriptados.txt*.
Sin embargo, es este el método de desencriptación que se aplicará para recuperar la información encriptada que incluye detalles de transacciones, datos personales de los clientes, y registros de operaciones financieras.

5. Ejecución:

- La ejecución del programa debe ser por línea de comandos, sólo a través del uso del ejecutable JAR, sin necesidad de tener que usar el IDE para ejecutar el programa.
- Se debe pasar por líneas de comandos como parámetros la carpeta donde está los archivos txt y el nombre del archivo donde se generarán los resultados.
- A modo de ejemplo se adjunta la forma en que debería ejecutarse la aplicación:

```
java -jar mi_programa.jar datasets output.txt
```

Lo anterior debería tener la siguiente estructura de directorio:

```
├── mi_programa.jar
├── resultados.txt
├── datasets
      ├── key.txt
      └── datos_encriptados.txt
```

- El archivo con las salidas debe ser reescrito cada vez que se ejecute el archivo jar.

6. Consideraciones de la entrega:

- Debe entregar en un archivo **zip**, con los siguientes elementos:
  - El código fuente con su proyecto.
  - El archivo jar ejecutable.
  - NO DEBE INCLUIR Los archivos txt del datasets.

¡IMPORTANTE! Tenga presente que el plagio parcial o total es una falta grave, por lo que es calificado con nota mínima e informado al director de carrera, arriesgando sanciones académicas.

VNS

RUBRICA

Criterio		Bueno	Suficiente	Insuficiente
Implementación del Código		5	2,5	0
Formato correcto del Proyecto		3	1,5	0
Estrategia de Desencriptación	Paso 1	3	1,5	0
	Paso 2	5	2,5	0
	Paso 3	5	2,5	0
	Paso 4	5	2,5	0
	Paso 5	3	1,5	0
Generación del jar		3	1,5	0
Correcta Ejecución del jar		5	2,5	0
Archivo de resultado correcto		5	2,5	0

Puntaje	Nota	Puntaje	Nota
0	1	21	3,5
0,5	1,1	21,5	3,6
1	1,1	22	3,6
1,5	1,2	22,5	3,7
2	1,2	23	3,7
2,5	1,3	23,5	3,8
3	1,4	24	3,9
3,5	1,4	24,5	3,9
4	1,5	25	4
4,5	1,5	25,5	4,1
5	1,6	26	4,1
5,5	1,7	26,5	4,2
6	1,7	27	4,3
6,5	1,8	27,5	4,4
7	1,8	28	4,5
7,5	1,9	28,5	4,6
8	2	29	4,7
8,5	2	29,5	4,8
9	2,1	30	4,9
9,5	2,1	30,5	4,9
10	2,2	31	5
10,5	2,3	31,5	5,1
11	2,3	32	5,2
11,5	2,4	32,5	5,3
12	2,4	33	5,4
12,5	2,5	33,5	5,5
13	2,5	34	5,6
13,5	2,6	34,5	5,7
14	2,7	35	5,8
14,5	2,7	35,5	5,8
15	2,8	36	5,9
15,5	2,8	36,5	6
16	2,9	37	6,1
16,5	3	37,5	6,2
17	3	38	6,3
17,5	3,1	38,5	6,4
18	3,1	39	6,5
18,5	3,2	39,5	6,6
19	3,3	40	6,6
19,5	3,3	40,5	6,7
20	3,4	41	6,8
20,5	3,4	41,5	6,9
		42	7

VNS